🐝 MMBee: Live Streaming Gift-Sending Recommendations via Multi-Modal Fusion and Behaviour Expansion

Jiaxin Deng[*†]
Institute of Automation
Beijing, China
dengjiaxin2022@ia.ac.cn

Shiyao Wang[*]
KuaiShou Inc.
Beijing, China
wangshiyao08@kuaishou.com

Yuchen Wang
KuaiShou Inc.
Beijing, China
wangyuchen11@kuaishou.com

Jiansong Qi
KuaiShou Inc.
Beijing, China
qijiansong@kuaishou.com

Liqin Zhao
KuaiShou Inc.
Beijing, China
zhaoliqin@kuaishou.com

Guorui Zhou[‡]
KuaiShou Inc.
Beijing, China
zhouguorui@kuaishou.com

Gaofeng Meng
Institute of Automation
Beijing, China
gfmeng@nlpr.ia.ac.cn

Abstract

Live streaming services are becoming increasingly popular due to real-time interactions and entertainment. Viewers can chat and send comments or virtual gifts to express their preferences for the streamers. Accurately modeling the gifting interaction not only enhances users' experience but also increases streamers' revenue. Previous studies on live streaming gifting prediction treat this task as a conventional recommendation problem, and model users' preferences using categorical data and observed historical behaviors. However, it is challenging to precisely describe the ***real-time content changes*** in live streaming using limited categorical information. Moreover, due to the ***sparsity of gifting behaviors***, capturing the preferences and intentions of users is quite difficult. In this work, we propose **MMBee** based on real-time Multi-Modal Fusion and Behaviour Expansion to address these issues. Specifically, we first present a Multi-modal Fusion Module with Learnable Query (MFQ) to perceive the dynamic content of streaming segments and process complex multi-modal interactions, including images, text comments and speech. To alleviate the sparsity issue of gifting behaviors, we present a novel Graph-guided Interest Expansion (GIE) approach that learns both user and streamer representations on large-scale gifting graphs with multi-modal attributes. It consists of two main parts: graph node representations pre-training and metapath-based behavior expansion, all of which help model jump out of the specific historical gifting behaviors for exploration and largely enrich the behavior representations. Comprehensive experiment results show that MMBee achieves significant performance improvements on both public datasets and Kuaishou real-world streaming datasets and the effectiveness has been further validated through online A/B experiments. MMBee has been deployed and is serving hundreds of millions of users at Kuaishou.

CCS Concepts

• **Information systems** → **Computational advertising**; **Multimedia information systems**.

Keywords

Graph, Multi-modal Learning, Live Streaming Recommendation

ACM Reference Format:
Jiaxin Deng, Shiyao Wang, Yuchen Wang, Jiansong Qi, Liqin Zhao, Guorui Zhou, and Gaofeng Meng. 2024. 🐝 MMBee: Live Streaming Gift-Sending Recommendations via Multi-Modal Fusion and Behaviour Expansion. In *Proceedings of Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'24).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

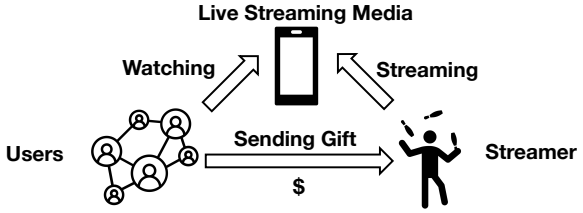


Figure 1: Example of the live streaming gifting scenario with the interactions among users and streamers.

Due to the rapid development of mobile device hardware and the Internet, live streaming has become a prevalent social service

[*]Equal contribution.
[†]Jiaxin Deng is also affiliated with School of Artificial Intelligence, University of Chinese Academy of Sciences.
[‡]Corresponding author.

KDD'24, August 25-29 2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

for people's daily lives. As one of the most popular live streaming platforms in China, *Kuaishou* has reached 386.6 million daily active users and the revenue generated by the live streaming business reached RMB 9.7 billion as of the third quarter of 2023, which heavily relies on Kuaishou's continuous optimization of the live streaming ecosystem and improvement of the recommendation system. As shown in Figure 1, on live streaming platforms, content creators can share their produced video content with users in real time, and users can interact with streamers and peers through live comments or discussions. They can even send virtual gifts to their favorite streamers, which is one of the main sources of revenue for the live-streaming business. Therefore, the task of live streaming gifting prediction is vital not only for enhancing user experience and streamer revenue but also for increasing the business effectiveness of the platform.

Recent years have witnessed several relevant methods for recommendation [12, 21, 25, 35, 36] and gifting prediction [11, 32] in live streaming. For example, MARS [11] introduces a two-stage recommendation approach applied in the Multi-Stream Party scenario, aiming to maximize reward earnings while optimizing user personal experience at the same time. However, this approach ignores the close connection between users' gifting behavior and the rapidly changing live content in the living room. To address this issue, MTA [32] designs a novel orthogonal module that fully utilizes the multi-modal features in live streaming. However, MTA treats the gift prediction as a time series prediction problem which does not consider users' personalization. Although typical behavior-based methods like SIM [20] can achieve personalized recommendations for gifting prediction, they may face the challenge of behavior sparsity in the context of live streaming. According to [6], DNN-based methods typically require a minimum of 5-10 historical behavior sequences to learn meaningful representations for modeling user interests. However, the average length of user's gifting behavior is as low as 0.3 anchors in our scenario. Therefore, gifting prediction requires a comprehensive consideration that combines user personalization under sparse behaviors and real-time content modeling to achieve optimal recommendation effectiveness.

To address these challenges, we propose **MMBee**: an efficient live streaming gifting prediction method based on real-time Multi-Modal Fusion and Behaviour Expansion. Specifically, we first design a Multi-modal Fusion Module with Learnable Query (MFQ). It helps the model to perceive the ***real-time content changes*** in live streaming through processing the complex visual frames, comments and audio in each streaming segment. In addition, aiming to address the ***sparsity problem*** in gifting prediction, we propose a novel Graph-guided Interest Expansion (GIE) approach. We first construct large-scale gifting graphs based on the history of gifting interactions. Then a graph pre-training scheme via contrastive learning (GraphCL) is adopted to learn general and robust streamer and user representations. Apart from these learned self-supervised embeddings, we further extend behavior sequences through metapaths with the graph structural information and optimize the representations in an end-to-end manner with online recommendation model. Both of the self-supervised and end-to-end learning schemes help model jump out of the specific historical gifting behaviors for potential preferences exploration and largely enrich the behavior representation. Finally, to meet the low latency requirements of the online serving system, we propose a decoupled graph offline training and online inference strategy. MMBee has now been deployed on the live-streaming recommendation system of Kuaishou, serving millions of active users every day.

Overall, our contributions are shown as follows:

- We propose a Multi-modal Fusion with Learnable Query (MFQ) module which fully leverages the dynamic multimodal content of live streaming and captures the distinct characteristics among streamers.
- Graph-guided Interest Expansion (GIE) module largely enriches the observed history behaviors of users and streamers with both self-supervised graph representation learning and metapath-based behavior expansion to alleviate the sparsity problem.
- We validate the effectiveness of MMBee through extensive offline experiments on Kuaishou's 3 billion scale industrial dataset and public dataset. Online A/B tests further show that MMBee brings significant online benefits and we build efficient industrial infrastructure to deploy MMBee on the real-world online live streaming recommendation.

2 Related Work

2.1 Live Streaming Gifting Recommendation

Existing works on live streaming gifting recommendation systems primarily view the whole live room as recommendation target and model the interaction between streamers and viewers only with categorical data. For instance, MARS [11] proposes a novel recommendation scenario called Multi-Stream Party (MSP) and designs two-phase methods to jointly maximize the reciprocal response of donations and optimize MSP personal satisfaction. LSEC-GNN [36] models the live stream e-commerce scenario using GNN and fully leverages the interaction information among streamers, users, and products. However, previous research ignores that dramatic content changes can occur even within the same live room thus it is vital to make full use of the multi-modal feature in live streaming. Aiming to solve this issue, MTA [32] introduces a novel orthogonal projection model to capture the cross-modal information interaction of real-time content. However, MTA formulates the gifting prediction task as a time series prediction problem and neglects the personalization modeling of users' interests. In conclusion, there still exists great room for improvement in existing methods for live-streaming gifting prediction.

2.2 Personalized Recommendation

The most widely adopted personalized recommendation methods in the industry are based on deep neural networks. For instance, DIN [42] models users' diverse interests in different target items by introducing attention mechanisms. SIM [20] proposes an online two-stage retrieval method that models relevant behaviors from a user's long-term history based on the features of the current candidate item However, in live-streaming gifting scenarios, it is challenging to achieve satisfactory results with these methods due to the sparsity issue in streamers and user interactions. Recently, several works combined with GNNs have introduced multimodal features to enrich the embedding of graph nodes. For instance, MMGCN [30] captures user preferences across different modalities by constructing a modal-specific user-item bipartite graph. EgoGCN [4] introduces a novel EGO fusion operation that enables inter-modal

message spreading. However, the aforementioned methods all rely on recursive graph convolution to study the node embedding, which can result in exponential computation cost and significant inference latency, especially in live streaming gifting recommendation scenarios where the model needs to handle millions of nodes and ensure low latency during inference. Therefore, it is crucial to design an efficient graph architecture for training and inference.

3 Preliminaries

In live streaming platforms, we use users to represent the viewers who watch live streaming and use authors to represent streamers. $V_u = \{u_1, u_2, \cdots, u_m\}$ is the set of users and $V_a = \{a_1, a_2, \cdots, a_k\}$ is the set of authors who are broadcasting at the current time, where m is the numbers of users and k is the numbers of authors. Previous studies treat the whole live streaming room as the recommendation target while ignoring the real-time change of streaming content. Thus, different from traditional recommendation tasks, we divide each live room into multiple consecutive 30s live segments and the live segment of author a at the current time is denoted by δ_a. We formulate that all live streaming segments of the current moment are the recommendation target and $M_a = \{v_a, s_a, t_a\}$ is the multi-modal raw data tuple, where v_a, s_a and t_a represent the visual frames, speech and comment text gathered from frame δ_a. Given a set of triples $\langle u_j, a_j, y_j \rangle$, $y_j = 1$ means that u_j send gift to a_j, otherwise $y_j = 0$. Thus, the gift-through-rate (GTR) prediction problem is to predict whether user u_i will send gift to a_i given the multi-modal raw data $M_{a_i} = \{v_{a_i}, s_{a_i}, t_{a_i}\}$ in the current live streaming segment δ_{a_i}:

$$p = f\left(a_i, u_i, M_{a_i}\right) \tag{1}$$

where p is termed as the gift through rate (GTR) and $f(\cdot)$ is the GTR prediction model. In this work, we choose SIM [20] as our foundational model, considering its widespread usage in the industry and its online efficiency and effectiveness. The objective function utilized in our method is the negative log-likelihood function, which is defined as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log p_i + (1 - y_i)\log(1 - p_i))) \tag{2}$$

where y_i denotes the ground truth label indicating whether current segment gets donation and $p_i \in [0, 1]$ is the predicted GTR.

4 Multi-modal Fusion with Learnable Query

For each live streaming segment, three frames are evenly sampled from each segment and necessary filtering process is conducted to clean the gathered ASR (Automatic Speech Recognition) and comment text. Then, we extract the multi-modal feature of raw data with Kuaishou's internal pre-trained 8 billion parameters multi-modal model K7-8B[1] and the extracted multi-modal feature sequences tuple of visual, speech and comment at the current moment of author a are represented with X_v, X_s and X_t, respectively.

Since processing and integrating information from different modalities is quite important [2, 10, 38], we propose a multi-modal fusion with learnable queries to ensure efficient modality interactions. Inspired by [32, 33], we adopt the orthogonal projection (OP) operation to maximize the complementation effects between different modalities. For example, take X_v as target modality, we calculate the relevant scores between the visual modality X_v with another two modalities by using correlation operations:

$$\begin{aligned} Corr_{vs} &= \text{Softmax}(X_v \cdot X_s) \\ Corr_{vt} &= \text{Softmax}(X_v \cdot X_t) \end{aligned} \tag{3}$$

where Softmax$(\cdot)$ is the softmax operation. Then, the irrelevant parts are obtained through $1 - x$ operation. Finally, the fused latent feature of visual modality Y_v is performed with:

$$\begin{aligned} Y_v = OP(X_v, X_s, X_t) = X_v + X_s \cdot (1 - Corr_{vs}) \\ + X_t \cdot (1 - Corr_{vt}) \end{aligned} \tag{4}$$

Note that $1 - Corr$ represents the dissimilarity vector that measures the difference between two modes' representation. It helps to preserve the parts of other modalities that are orthogonal to the target modality and remove duplicate information to prevent redundancy.

Then, as shown in the online stage of Figure 2, we utilize the orthogonal latent features in a hybrid fusion [23] manner applied with cross-attention and self-attention [27] alternately. The fused feature $\boldsymbol{h}_f$ is gotten with:

$$\begin{aligned} \boldsymbol{h}_v &= \text{CrossAttention}(X_v \boldsymbol{W}_v^Q, Y_v \boldsymbol{W}_v^K, Y_v \boldsymbol{W}_v^V), Y_v = OP(X_v, X_s, X_t) \\ \boldsymbol{h}_s &= \text{CrossAttention}(X_s \boldsymbol{W}_s^Q, Y_s \boldsymbol{W}_s^K, Y_s \boldsymbol{W}_s^V), Y_s = OP(X_s, X_t, X_v) \\ \boldsymbol{h}_t &= \text{CrossAttention}(X_t \boldsymbol{W}_t^Q, Y_t \boldsymbol{W}_t^K, Y_t \boldsymbol{W}_t^V), Y_t = OP(X_t, X_s, X_v) \\ \boldsymbol{h}_f^{'} &= \boldsymbol{h}_v \oplus \boldsymbol{h}_s \oplus \boldsymbol{h}_t \\ \boldsymbol{h}_f &= \text{SelfAttention}(\boldsymbol{h}_f^{'} \boldsymbol{W}_f^Q, \boldsymbol{h}_f^{'} \boldsymbol{W}_f^K, \boldsymbol{h}_f^{'} \boldsymbol{W}_f^V) \end{aligned} \tag{5}$$

However, the fused feature $\boldsymbol{h}_f$ can only reflect the content-level representation, thus lacking the connection to distinctive characteristics across various types of authors. To address this issue, we produce several learnable query[13, 43] tokens $\boldsymbol{q}_m \in \mathbb{R}^{N \times d}$ to extract streamer-aware content patterns. Note that each author keeps a set number of learnable query embeddings which are randomly initialized. N represents the number of query tokens for each author. The learnable query first interacts with fused multi-modal features through cross-attention layers as

$$\boldsymbol{h}_m^{'} = \text{CrossAttention}(\boldsymbol{q}_m \boldsymbol{W}_c^Q, \boldsymbol{h}_f \boldsymbol{W}_c^K, \boldsymbol{h}_f \boldsymbol{W}_c^V) \tag{6}$$

Then the queries interact with each other through self-attention layers to fuse the necessary information among different patterns:

$$\boldsymbol{h}_m = \text{SelfAttention}(\boldsymbol{h}_m^{'} \boldsymbol{W}_s^Q, \boldsymbol{h}_m^{'} \boldsymbol{W}_s^K, \boldsymbol{h}_m^{'} \boldsymbol{W}_s^V) \tag{7}$$

The multi-modal fusion module benefits from the learnable queries in two major aspects: 1) Each author has learnable tokens that store their specific highlight content patterns. The tokens can be activated at certain moments of awesome content, which is quite useful for gifting prediction. 2) These queries help align the multimodal representations with the ID embedding based recommendation space, thereby maximizing their mutual information. Consequently, the integration of learnable queries further enhances model's ability to capture real-time content.

5 Graph-guided Interest Expansion

5.1 User-to-Author and Author-to-Author Graph

Based on the users' donation history, we first construct a User-to-Author(U2A) graph $G_1(V_u \cup V_a, E_1)$ that represents the correlation between users and authors, where V_u and V_a are the sets of users and

[1] https://ir.kuaishou.com/news-releases/news-release-details/kuaishou-receives-leading-innvoation-digitial-economy-and-other

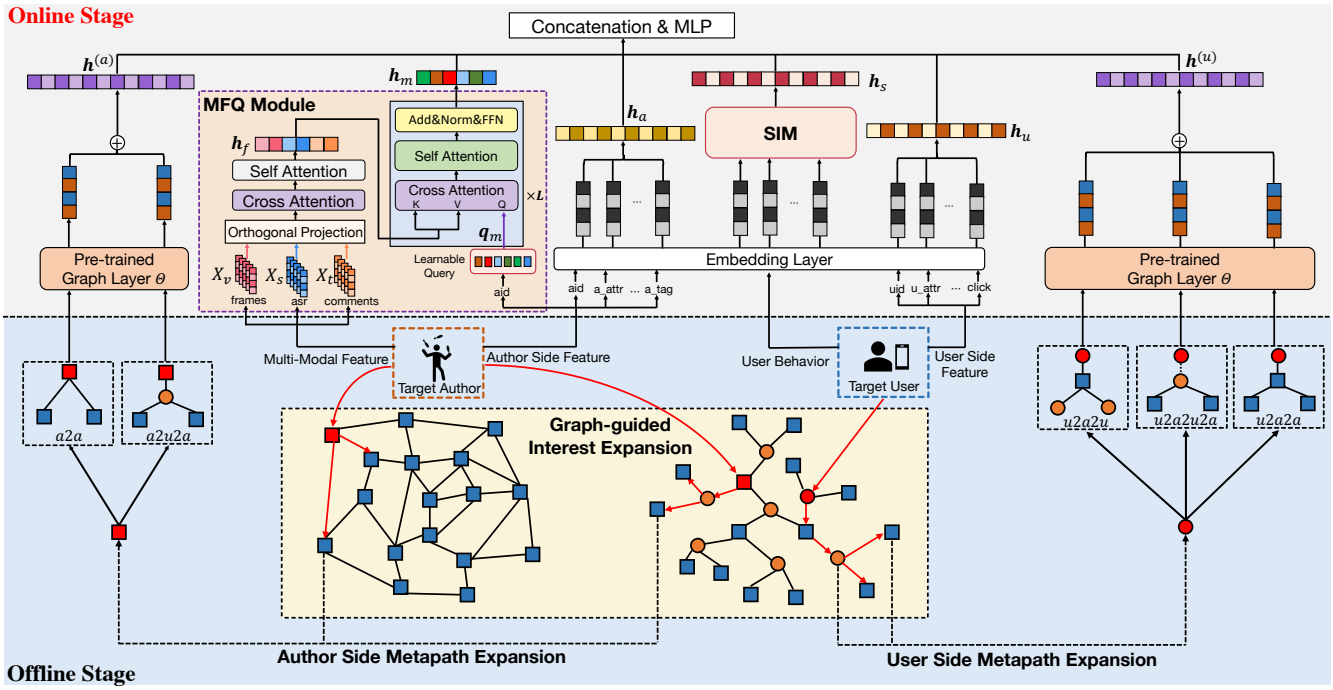
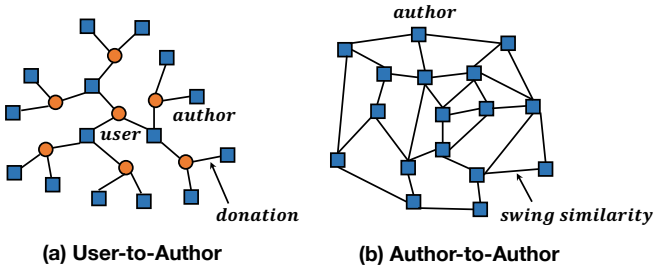


Figure 2: The overall framework of MMBee, consists of two stages: (i) the offline Graph-guided Interest Expansion (GIE) stage conducts the behavior expansion based on the target user and author; (ii) the online GTR prediction stage aggregates the real-time mutli-modal content and expanded behavior for end-to-end training.

authors respectively and E_1 represents the donation relationship between users and authors. As illustrated in Figure 3 (a), the circle represents the user, and the square represents the author. If a user has previously made a donation, an edge exists between the user and the donated author in this graph. The weight of the edge is the amount of donated money and an author node has the attribute of aggregated multi-modal feature. In this way, the large User-to-Author graph is constructed.



(a) User-to-Author (b) Author-to-Author

Figure 3: User-to-author and author-to-author donation graph construction with donation history.

Based on the aforementioned U2A graph, we further construct the Author-to-Author (A2A) Graph $G_2(V_a, E_2)$ to represent the interdependence among authors, where E_2 denotes the Swing similarity [34] relationship among authors. In this graph, each node represents an author, and the edge weight represents the Swing similarity between the authors. The similarity between author i and author j is given below:

$$s(i,j) = \sum_{u \in U_i \cap U_j} \sum_{v \in U_i \cap U_j} \frac{1}{\alpha + |I_u \bigcap I_v|} \tag{8}$$

where U_i is the set of users who have made donations on author i and I_u is the set of authors that donated by user u.

A2U graph is established through donation relationship between users and authors. The design of edge weights and sampling strategies helps enrich the representations of authors who have a rich history of being donated. However, there are some new or cold-start authors. Their limited donation history makes it difficult to benefit from the A2U graph. Fortunately, A2A graph is built from the swing similarity defined in Equation 8, which finds substitutable authors based on the substructures of user-author donation bi-partitive graph. It is useful for linking cold-start author to warm-start author and encouraging the engagement of cold-start authors, so A2A graph is quite necessary.

After constructing U2A and A2A graphs, we first leverage the graph node representation learning approach to train graph embedding layer in Section 5.2. Next, we propose metapath based behavior expansion process to enrich sparse behavior sequences in Section 5.3. To provide a precise demonstration of the abovementioned methods, we first establish the following definition:

Definition 1 (Metapath[5]). *Metapath is defined as a relation sequence to capture the specific structural relation between objects. In A2U and A2A graph, we define five metapaths: three metapaths ρ_{u2a2u}, $\rho_{u2a2u2a}$, ρ_{u2a2a} begin from target user, for example $\rho_{u2a2a} = User \rightarrow Author \rightarrow Author$ metapath indicates that user make donation to authors in U2A graph, and these authors further retrieve*

Algorithm 1: GraphCL

Initialize $\mathcal{L} \leftarrow 0$;
Graph $G_1(V_u \cup V_a, E_1)$, graph node embedding layers parameter $\Theta \in \mathbb{R}^{|V_u \cup V_a| \times d}$, walks epoch γ;
for $i = 0$ *to* γ **do**
  O = Shuffle($V_u \cup V_a$);
  for $v_t \in O$ **do**
    $V_p \leftarrow \{\}, V_n \leftarrow \{\}$;
    if $v_t \in V_u$ **then**
      $V_p \leftarrow \mathcal{N}^{(2)}_{\rho_{u2a2u}}(v_t)$;
      $V_n \leftarrow V_n \cup RandomSample(V_u)$;
    end
    if $v_t \in V_a$ **then**
      $V_p \leftarrow \mathcal{N}^{(2)}_{\rho_{a2u2a}}(v_t)$;
      $V_n \leftarrow V_n \cup RandomSample(V_a)$;
    end
  end
  $\mathcal{L} \leftarrow \mathcal{L}_{CE} + \lambda \mathcal{L}_{NCE}$;
  $\Theta \leftarrow \Theta - \alpha \frac{\partial \mathcal{L}}{\partial \Theta}$;
end
Output: Trained graph node embedding layers parameter Θ

similar authors in A2A graph, and we define two metapath ρ_{a2a} and ρ_{a2u2a} which begin form target author.

Definition 2 (Metapath-guided Neighbors[5]). *Given a node o and a metapath ρ (start from o) in the gragh, the metapath-guided neighbors is defined as the set of all visited nodes when the node o walks along the given metapath. We denote the i-th step neighbors of object o as $\mathcal{N}^{(i)}_{\rho}(o)$. For example, give the metapath $\rho_{u2a2u} = User \rightarrow Author \rightarrow User$, we can get metapath-guided neighbors as $\mathcal{N}^{(1)}_{\rho_{u2a2u}}(u_t) = \{a_1, a_2\}$, $\mathcal{N}^{(2)}_{\rho_{u2a2u}}(u_t) = \{u_1, u_2, u_3\}$.*

5.2 Node Representation Pre-training with GraphCL

Previous studies [3, 18, 19, 37] have shown that graph node embedding algorithms are beneficial for recommendation systems for tackling data sparsity problem because these methods are able to effectively capture the user-author relatedness from graph structures. To leverage the connectivity information of the whole graph, we apply the graph contrastive learning (GraphCL) framework to train the graph embedding layer. Aiming to cluster similar nodes together while pushing away dissimilar ones, we loop through all nodes in the whole graph G_1 and obtain positive sample set V_p through the metapath-guided neighbor process and the negative nodes set V_n are sampled randomly. The positive and negative nodes are utilized with the Cross-Entropy loss $\mathcal{L}_{CE}$ and InfoNCE [17] $\mathcal{L}_{NCE}$ loss for optimizing the parameters of the node embedding layers. Algorithm 1 shows the core of our approach and the trained graph node embedding implies the connectivity information from the whole graph. The InfoNCE loss is defined with Equation 9.

$$\mathcal{L}_{NCE} = -\frac{1}{|V_p|} \sum_{v_i \in V_p} \log \left(\frac{\exp \Theta(v_t)^T \Theta(v_i)}{\exp \Theta(v_t)^T \Theta(v_i) + \sum_{v_j \in V_n} \exp \Theta(v_t)^T \Theta(v_j)} \right) \quad (9)$$

5.3 Metapath-guided Behavior Expansion through End-to-End Training

When analyzing the node number distribution of the constructed A2U graph, we observe that the average outdegree of user nodes is 0.32. It becomes difficult for widely used behavior-based models like SIM to study meaningful representations and explore potential gifting preferences. Furthermore, the graph embedding in Section 5.2 is trained in a self-supervised manner which is not directly optimized for the recommendation model. To address these challenges, we expand the behavior sequence of the target user and author using various pre-defined metapaths [5]. Due to the computation cost, we perform up to 3-hop neighbors on both U2A and A2A Graph. We enumerated all possible metapaths and five metapaths with the highest scores are selected using commonly used feature importance filtering methods as follows:

- $\mathcal{N}^{(2)}_{\rho_{u2a2u}}(u_t)$ begins with the target user u_t and follow this metapath. The retrieved behavior sequence is a set of users who share the same authors as the target user. Therefore, this metapath gets similar users who share the similar interests of the target user.
- $\mathcal{N}^{(3)}_{\rho_{u2a2u2a}}(u_t)$ helps identify potential authors that may reflect the interest of the target user, excluding the authors they have already donated to in the past.
- $\mathcal{N}^{(2)}_{\rho_{u2a2a}}(u_t)$ is based on the target user's donated authors history and it retrieves similar authors in the A2A graph to find similar authors with respect to the target user.
- $\mathcal{N}^{(1)}_{\rho_{a2a}}(a_t)$ begins with the target author a_t, it retrieves the similar authors in the A2A graph. Therefore, this metapath helps obtain similar authors to the target author.
- $\mathcal{N}^{(2)}_{\rho_{a2u2a}}(a_t)$ indicates that a group of users donates to the target author in the U2A graph, and these users subsequently donate to another group of authors. Therefore, this metapath helps identify potential interest authors for the target author.

Based on these metapath-guided neighbors, we significantly enrich the behavior sequence of the target user and author. During the offline GIE stage, we store the pre-aggregated embeddings of the metapath-guided expanded neighbors of each user and author on the graph into memories or key-value databases to be further utilized in the online training stage.

In order to eliminate the gap between pre-trained node representation and online recommendation model, we gather the expanded sequences and optimize them with GTR prediction objective in recommendation model for end-to-end training. The generation of user side expanded graph representation $\mathbb{E}^{(u)}$ can be formalized as:

$$\mathbb{E}^{(u)} = \left\{ \Theta(v_i) | v_i \in \mathcal{N}^{(2)}_{\rho_{u2a2u}}(u_t) \cup \mathcal{N}^{(3)}_{\rho_{u2a2u2a}}(u_t) \cup \mathcal{N}^{(2)}_{\rho_{u2a2a}}(u_t) \right\} \quad (10)$$

And the generation of author side expanded graph representation $\mathbb{E}^{(a)}$ can be formalized as:

$$\mathbb{E}^{(a)} = \left\{ \Theta(v_i) | v_i \in \mathcal{N}^{(1)}_{\rho_{a2a}}(a_t) \cup \mathcal{N}^{(2)}_{\rho_{a2u2a}}(a_t) \right\} \quad (11)$$

where $\Theta(\cdot)$ represents the graph node embedding layer operation and author multi-modal attribute retrieval operation. Then we implement the mean pooling and concatenate operation in the recommendation model to get the final graph embedding $\boldsymbol{h}^{(u)}$ and $\boldsymbol{h}^{(a)}$ for end-to-end training:

$$\boldsymbol{h}^{(u)} = \text{MeanPooling}(\mathbb{E}^{(u)}) \quad \boldsymbol{h}^{(a)} = \text{MeanPooling}(\mathbb{E}^{(a)}) \quad (12)$$

5.4 System Deployment

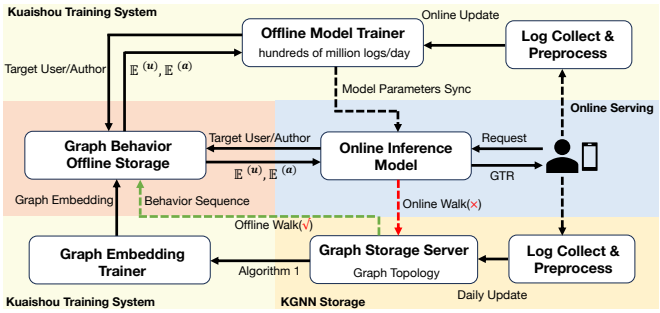


Figure 4: The deployment of MMBee in online live streaming GTR prediction system.

As shown in Figure 4, our recommendation model and graph embedding layer are trained on Kuaishou's large-scale distributed training system. Each day, hundreds of millions of users visit Kuaishou, actively watching and interacting with live-streaming content, resulting in the generation of hundreds of millions of logs for watching and interaction. These logs are collected, preprocessed in real-time, and utilized for training the model. Our training system incrementally updates the model parameters by incorporating the latest user-author interactions, multi-modal content features, and trained graph embedding. The trained parameters are synchronized to the online inference model for online serving. To train graph embedding, we first gather the users' historical donation behavior and utilize it to build the User-Author and Author-Author donation graphs. The topology of these two graphs is stored in a key-value based storage system called KGNN[2]. Then the graph embedding trainer requests the KGNN server with Algorithm 1 for training the node embedding layer and the KGNN storage updates once a day.

During the training and inference processes of the recommendation model, it needs to request the metapath-guided neighbors of the target user and author. As shown by the red dashed line in Figure 4, one approach is to dynamically request the KGNN storage. However, this method can impose significant computational overhead on the KGNN server and result in great time delays when walking on the entire graph. To address this issue, as shown by the green dashed line in Figure 4, we apply the pre-requested expansion manner and store the metapath-guided neighbors of all nodes in the graph in the Graph Behavior Offline Storage in advance. As a result, the online recommendation model can directly access the Graph Behavior Offline Storage to retrieve the sequence without having to walk on the graph.

6 Experiment

6.1 Dataset

6.1.1 Kuaishou Dataset We first test our method on company internal dataset called Kuaishou Dataset. It includes about 3 billion user interaction logs with live-streaming content in Kuaishou App. This dataset is collected as follows: if the user requests the recommendation service and makes a donation at time t, then only the segment containing t will be taken as the positive training sample while other samples will be ignored. On the contrary, if the recommended live broadcast has impressed but users' donation behavior does not occur until exiting, the segment when user exit will be adopted as negative sample [14]. With this process, the sparsity of of Kuiashou dataset is 99.969% which is reasonable. Kuaishou dataset is composed of two parts: D_{train} and D_{test}, where D_{train} is users' real interaction logs from 7 days of all live streaming content during that period. The D_{train} is used for the training phase. The D_{test} is sampled from the following one-day logs after D_{train} is collected, which is used to test model's performance.

6.1.2 Public Dataset To prove the effectiveness of our proposed MFQ and GIE module, we also compare our method on two public short video recommendation datasets: TikTok and MovieLens. The statistics of datasets are shown in Table 3.

Table 3: The statistics of public datasets. V, A, and T represent the dimensions of visual, acoustic and textual features.

Dataset	#Interactions	#Items	#Users	Sparsity	V	A	T
Tiktok	726,065	76,085	36,656	99.97%	128	128	128
Movielens	1,239,508	5,986	55,485	99.63%	2048	128	100

Movielens[3] [7] is a widely used dataset [9, 22, 24, 31] for the recommendation task. The raw data is initially acquired by collecting movie descriptions from Movielens-10M and crawling the corresponding trailers from YouTube. Textual features are subsequently extracted from the descriptions using the Sentence2Vector [1]. In terms of visual modality, key frames are initially extracted from the retrieved videos and then processed by a pre-trained ResNet50 model [9] to obtain visual features. The acoustic features are obtained using VGGish [12], following a soundtrack separation procedure implemented with the FFmpeg software.

TikTok[4] is published by TikTok, a micro-video sharing platform that enables users to create and share micro-videos with durations ranging from 3 to 15 seconds. TikTok comprises users, micro-videos, and their interactions, such as clicks. The features of the micro-videos in each modality are extracted and made available without providing the raw data. Specifically, the textual characteristics are extracted from the micro-video captions provided by users.

6.2 Baseline

On the Kuaishou dataset, we choose two widely used baselines MMoE [16] and SIM [20] for comparison. We evaluate the performance of our method by comparing it with the following recommendation method that is integrated with MMoE and SIM:

- **BDR** [39] consists of User-to-User and Author-to-Author graphs, enabling simultaneous prediction from both perspectives.
- **MTA** [32] leverages multimodal time-series analysis to effectively integrate information from different modalities. This approach does not consider modeling personalized preferences.
- **EgoFusion** [4] allows the spread of inter-modal messages in EgoGCN. In our work, we apply the Ego fusion operation to the multi-modal feature of node attribution to generate the multi-modal embedding and we exclude the MFQ module for a fair comparison in this baseline.

On the public dataset, we compare the performance of our method with the following GCN-based models:

[2]https://www.jiqizhixin.com/articles/2020-12-08

[3]https://grouplens.org/datasets/movielens/

[4]http://ai-lab-challenge.bytedance.com/tce/vc/

Table 1: Performances of different methods on Kuaishou dataset. * represents the absolute improvement.

Methods	GTR					
	AUC	**Impr.***	UAUC	**Impr.***	GAUC	**Impr.***
MMoE [16]	0.956230	-	0.730186	-	0.746711	-
MMoE+BDR [39]	0.956908	+0.0678 %	0.730625	+0.0439 %	0.747136	+0.0425 %
MMoE+MTA [32]	0.957095	+0.0865 %	0.731450	+0.1264 %	0.747327	+0.0616 %
MMoE+EgoFusion [4]	0.956952	+0.0722 %	0.731418	+0.1232 %	0.747275	+0.0564 %
MMoE+MFQ	0.956902	+0.0672 %	0.731975	+0.1789 %	0.747275	+0.1764 %
MMoE+GIE	0.957064	+0.0834 %	0.733853	+0.3667 %	0.751239	+0.4528 %
MMoE+Ours(MFQ+GIE)	**0.95723**	**+0.1001 %**	**0.735776**	**+0.5590 %**	**0.753017**	**+0.6306 %**
SIM [20]	0.958656	-	0.732239	-	0.748383	-
SIM+BDR [39]	0.958419	-0.0237 %	0.734757	+0.2518 %	0.750684	+0.2301 %
SIM+MTA [32]	0.958867	+0.0211 %	0.734921	+0.2682 %	0.750802	+0.2419 %
SIM+EgoFusion [4]	0.959387	+0.0085 %	0.735608	+0.3369 %	0.751669	+0.3286 %
SIM+MFQ	0.959202	+0.0546 %	0.735717	+0.3478 %	0.751780	+0.3397 %
SIM+GIE	0.959802	+0.1146 %	0.738309	+0.6070 %	0.755154	+0.6771 %
SIM+Ours(MFQ+GIE)	**0.960302**	**+0.1646 %**	**0.743678**	**+1.1439 %**	**0.76044**	**+1.2057 %**
p-value	$1.02e^{-3}$		$2.01e^{-3}$		$5.12e^{-3}$	

Table 2: Performances of different methods on Tiktok and Movielens datasets.

Methods	TikTok			Movielens		
	Recall@10	Precision@10	NDCG@10	Recall@10	Precision@10	NDCG@10
NGCF [28]	0.0292	0.0045	0.0156	0.1198	0.0289	0.0750
LightGCN [8]	0.0448	0.0082	0.0261	0.1992	0.0479	0.1324
MMGCN [30]	0.0544	0.0089	0.0297	0.2028	0.0506	0.1361
GRCN [29]	0.0392	0.0065	0.0221	0.1402	0.0338	0.0882
EgoGCN [4]	0.0569	0.0093	0.0330	0.2155	0.0524	0.1444
DIN [42]	0.0403	0.0074	0.0235	0.1372	0.0330	0.0912
SASRec [9]	0.0435	0.0043	0.0215	0.1914	0.0191	0.1006
SIM [20]	0.0413	0.0079	0.0245	0.1470	0.0429	0.1011
MMMLP [15]	0.0509	0.0081	0.0297	0.1842	0.0484	0.1328
MMSSL [20]	0.0553	0.0055	0.0299	**0.2482**	0.0170	0.1113
Ours	**0.0605**	**0.0097**	**0.0347**	0.2317	**0.0566**	**0.1573**
p-value	$1.29e^{-5}$	$6.23e^{-6}$	$7.29e^{-5}$	$2.75e^{-5}$	$2.81e^{-3}$	$1.61e^{-2}$

- **NGCF** [28] exploits high-order connectivity and collaborative signal by propagating embeddings on user-item graph structure.
- **LightGCN** [8] remove feature transformation and nonlinear activation from standard GCNs to construct a lightweight structure for collaborative filtering.
- **MMGCN** [30] captures modality-specific user preferences and integrates them to form user representations, which is used to evaluate their affinities towards the content features of the items.
- **GRCN** [29] refines the user-item bipartite sub-graphs for different modalities and adjusts the representation of the user and item accordingly to improve the prediction of their interactions.
- **EgoGCN** [4] improves the user-item interactions through an effective graph fusion approach called EGO fusion.

We also further compare the performance of our method with with well-known recommendation methods besides graph and recent methods integrated with multi-modal features:

- **DIN** [42] captures temporal interests from history behavior sequence with GRU and attentional update gate.
- **SASRec** [9] is a classic transformer-based sequential recommender.
- **SIM** [20] models life-long behavior in the two cascading stages with General Search Unit (GSU) and Exact Search Unit (ESU).
- **MMMLP** [15] adapts MLP-Mixer for modelling multi-modal feature in sequential recommendation .
- **MMSSL** [20] addresses the sparsity issue by introducing self-supervised tasks that maximize the mutual information between multiple content-augmented views.

6.3 Evaluation Metrics

For offline evaluation on Kuiashou dataset, we use the training set D_{train} to train all methods and evaluate the performance of all methods on the test set D_{test}. We report the average performance over hours. We adopt three widely adopted metrics: AUC, UAUC and GAUC [40] to evaluate the performance of different methods. AUC represents the probability that the score of a positive sample is higher than that of a negative sample, reflecting the ranking capability of a model. UAUC is the average of AUC values calculated for different users and GAUC is the weighted average of UAUC considering the impressions. They are defined as follows:

$$\text{UAUC} = \frac{1}{N}\sum_{i=1}^{N}\text{AUC}_i \quad \text{GAUC} = \frac{\sum_{i=1}^{N} impression_i * \text{AUC}_i}{\sum_{i=1}^{N} impression_i} \tag{13}$$

where N refers to the number of active users in the testing set. UAUC and GAUC alleviate the bias among users and consider the effect of impression to evaluate the model's performance in a finer and fair manner.

6.4 Overall Performance

Table 1 shows the performance of all models on the Kuaishou dataset. Note that given the large number of users and samples in Kuaishou dataset, an improvement of 0.5% in AUC, UAUC, and GAUC during offline evaluation holds significant value to bring obvious online gains for business. Table 2 presents the performance of several competitors on public Tiktok and Movielens datasets.

First, our method surpasses all baselines by a significant margin on Kuaishou dataset. Our method MFQ significantly outperforms traditional live streaming recommendation models BDR and MTA in UAUC and GAUC for two main reasons. Firstly, BDR ignores the modeling of multi-modal content, while MTA lacks the connection to distinctive characteristics across various types of authors. In contrast, our MFQ successfully leverages the multi-modal content of the target live-streaming room and adopts learnable queries to extract streamer-aware content patterns. Additionally, our method GIE also outperforms the graph-based method EgoFusion which provides evidence that the metapath-guided behavior expansion process greatly enhances behavior representation and explores potential donation preferences.

Secondly, our method exhibits generalizability to a common behavior-based model. Our method has seamlessly integrated into two widely used behavior-based methods, MMoE and SIM, both of which demonstrate significant performance improvements. Moreover, MMBee is not limited to these two behavior-based models and can be easily adapted to other methods such as DIN [42] and DIEN [41] as well.

Thirdly, our method is not restricted to gifting prediction tasks and it also proves effectiveness in multi-modal recommendation tasks. As shown in Table 2, our method exhibits great improvement when compared to several strong multi-modal recommendation baselines. This gain mainly comes from two folds: (1) The metapath-guided neighbors in our method enable better capture of user preferences, but other graph-based methods only rely on implicit learning from graph embeddings. (2) The MFQ module enhances the fusion of multi-modal features from short videos and clusters different videos with learnable queries initialized with item embedding, thereby benefiting further performance improvement of the recommendation model.

6.5 Ablation Study

Graph-level Ablation: In order to investigate the importance of different metapath neighbors and the effect of graph embedding training, we remove five expanded sequences in turn and evaluate the performance of ablated graph embedding features. The results are presented in Table 4, where we use $(-)$ to represent the removed part or feature. For example, $\boldsymbol{h_{u2a2u}(-)}$ means removing the metapath neighbors $\mathcal{N}^{(2)}_{\rho_{u2a2u}}(u_t)$ in recommendation model, $\boldsymbol{\Theta(-)}$ denotes removing the learned graph node embedding layers but remaining the expanded sequence and $\boldsymbol{h_g(-)}$ represents removing all features of graph modeling. From table 4, we can observe that $\boldsymbol{h_g(-)}$ drops -0.1100% of AUC and $\boldsymbol{\Theta(-)}$ also leads to a significant drop in performance which means that the GIE modeling is a very important supplement to the observed history behaviors. This suggests that the explicit metapath-based behavior expansion process and implicit graph node embedding learning are all beneficial to model's performance. Furthermore, among five expanded behavior sequences, we observed the metapath of ρ_{a2u2a} and $\rho_{u2a2u2a}$ are the most important sequences among them.

Multi-modal Ablation: We also investigate the influence of the multi-modal feature in MFQ module. Specifically, $\boldsymbol{h_m(-)}$ denotes removing all multi-modal content and $\boldsymbol{q_m(-)}$ represents removing the learnable query and cross attention. From the table we also observe significant performance drops when removing the multi-modal feature and the learnable query is also necessary. We also further study the influence of different modalities, including visual frame X_v, speech X_s and comment X_t and report the ablation results compared with MMBee on Kuaishou dataset in Table 5. We find visual modality has the most important impact, causing the most performance degradation when removed. The speech and comment modality have a lesser impact factor but still show an innegligible effect on the model's overall performance. So we choose all three modalities in MMBee.

Table 5: Ablation study on different modality impact.

Methods	X_v	X_s	X_t	AUC Impr.	UAUC Impr.	GAUC Impr.
MMBee	√	√	√	0.0000%	0.0000%	0.0000%
$\boldsymbol{X_v(-)}$	-	√	√	-0.1101%	-0.2069%	-0.2939%
$\boldsymbol{X_s(-)}$	√	-	√	-0.1090%	-0.1565%	-0.1383%
$\boldsymbol{X_t(-)}$	√	√	-	-0.0839%	-0.0933%	-0.1790%

Hyperparameters Ablation: We provide further experiment results about hyperparameters as follows:

- **Dimension of MFQ.** We compare 32/64/128 dimensions of MFQ on Kuaishou dataset and the speed is tested on 20*Tesla T4 (15GB) GPUs measured in examples/second. As shown in Table 6, the 64 dimension holds the best trade-off with computation efficiency and accuracy.

Table 6: The influence of dimension of MFQ.

Dimension	Speed	FLOPs	AUC Impr.
32	144.17K	154.61M	0.0000%
64	141.76K	190.27M	0.1744%
128	132.73K	229.30M	0.2105%

- **Segment Length.** We additionally choose 10/20 consecutive live segments and compared them with 5 segments on Kuaishou dataset. Table 7 shows that 10 live segments get obvious gain but when it comes to 20 the further gain is modest. However, 10 segments significantly increase resource costs (including storage, training and serving) making it infeasible to deploy in production. So we use 5 segments in MMBee.

Table 7: The influence of segments length..

Length	AUC Impr.	UAUC Impr.	GUC Impr.	FLOPs	Speed
5	0	0	0	190.27M	141.76K
10	0.0237%	0.2037%	0.2384%	194.09M	122.60K
20	0.0733%	0.2369%	0.2500%	203.04M	108.17K

Table 4: Ablation Study on Graph and Mutli-modal level. The number in bold indicates a significant performance degradation.

Category	Operator	AUC	Impr.	UAUC	Impr.	GAUC	Impr.
-	SIM	0.958656	-0.1646%	0.732239	-1.1439%	0.748383	-1.2057 %
Graph	$h_{u2a2u}(-)$	0.959842	-0.0460 %	0.743492	-0.0186 %	0.76014	-0.0300 %
	$h_{u2a2u2a}(-)$	0.959706	**-0.0596 %**	0.738322	**-0.5356 %**	0.755081	**-0.5359 %**
	$h_{u2a2a}(-)$	0.960162	-0.0140 %	0.743248	-0.0430 %	0.75976	-0.0680 %
	$h_{a2a}(-)$	0.960002	-0.0300 %	0.742931	-0.0747 %	0.759818	-0.0622 %
	$h_{a2u2a}(-)$	0.959462	**-0.0840 %**	0.738378	**-0.5300 %**	0.754722	**-0.5718 %**
	$\Theta(-)$	0.959782	**-0.0520%**	0.736832	**-0.6846 %**	0.752625	**-0.7815 %**
	$h_g(-)$	0.959202	**-0.1100%**	0.735608	**-0.8070 %**	0.751669	**-0.8771 %**
Multi-modal	$h_m(-)$	0.959802	**-0.0500 %**	0.738309	**-0.5369 %**	0.755154	**-0.5286 %**
	$q_m(-)$	0.960091	-0.0211%	0.740996	-0.2682 %	0.758021	-0.2419 %
-	**Ours**	0.960302	0.0000 %	0.743678	0.0000 %	0.76044	0.0000 %

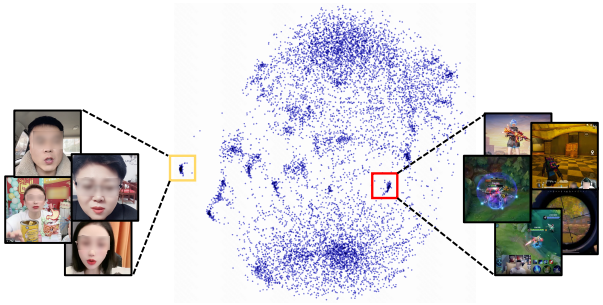


Figure 5: Visualization of the learnable query distribution in MFQ, where each point indicates an author.

6.6 Visualization Study

We conduct experiment to visualize the learnable query representations in MFQ. We randomly sample 10,000 authors and visualize these representations using t-SNE [26] in 2 dimensions, as illustrated in Figure 5. The points in this graph represent the sampled authors, and it is obvious that there are several distinct clustering centers and we mark two of them by the yellow and red boxes. To demonstrate the characteristics of each clustering center, we provide some visual frames for further explanation. We observe that authors in the yellow box tend to be chatting authors, while gaming authors tend to appear in the red box. These phenomena support our assumption that learnable query can represent distinctive characteristics of various types of authors.

6.7 Study of Online Response Time

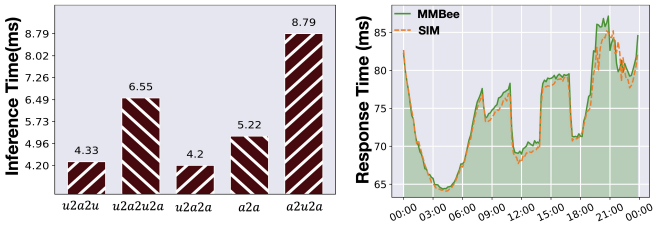


Figure 6: Left shows the response time of different metapaths and right shows the system's overall response time change during one day.

We investigate the online response time when recommendation requests the KGNN server and Figure 6 (left) shows the different response time when requesting different metapath behaviors. It is obvious that the max lag can reach 8.79 ms but this is not allowed in real-world applications. So we applied the pre-request of expansion behaviors and stored it in advance (described in Section 5.4) so the online recommendation model could access the embedding server instead of walking through the graph on the fly. We evaluate the efficiency of offline storage by comparing the time cost between the baseline system and the system equipped with MMBee. The response time (in milliseconds) with millions of queries per second during Jan. 24, 2024 is presented in Figure 6 (right), where the yellow and green lines represent the response time of the baseline system and MMBee. Empirical evidence shows that the response time of MMBee is only about 1 ms more than that of the baseline system on average, which is brought by the extra expanded graph behavior retrieving and computational overhead of inference.

6.8 Online Result

To evaluate the online performance of MMBee, we conduct strict online A/B tests on Kuaishou's business scenarios of live streaming main page spanning from 2023/10/05 to 2023/10/09 and we compare the performance of MMBee and SIM with 1% main traffic for experiments. Note that MMBee integrates our proposed MFQ and GIE into SIM backbone. We use **NGU** (Number of users who sent gifts) and **NGC** (the total number of gifts sent) as main online metrics. Online evaluation shows that MMBee has achieved **2.862%** on NGU and **4.775%** lift on NGC metric, which indicates that MMBee achieves much better recommendation results and brings considerable revenue increments for the platform.

7 Conclusion

In this paper, we propose a novel real-time multi-modal fusion and behavior expansion model called MMBee for live streaming gifting prediction. The model efficiently leverages real-time multi-modal features and effectively exploits metapath-guided expanded behaviors to enhance the performance of GTR prediction. We address two important challenges in live streaming gifting prediction, namely the multi-modal modeling and behavior sparsity, by introducing the Multi-modal Query Fusion (MFQ) and Graph-guided Interest Expansion (GIE) modules. Extensive experiments on real-world datasets demonstrate the excellent performance of MMBee.

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*.

[2] Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. 2019. Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 8102–8109.

[3] Rose Catherine and William Cohen. 2016. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM conference on recommender systems*. 325–332.

[4] 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.

[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 the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2478–2486.

[6] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 311–320.

[7] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[8] 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 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[9] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[10] Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard Product for Low-rank Bilinear Pooling. In *The 5th International Conference on Learning Representations*.

[11] Hsu-Chao Lai, Jui-Yi Tsai, Hong-Han Shuai, Jiun-Long Huang, Wang-Chien Lee, and De-Nian Yang. 2020. Live multi-streaming and donation recommendations via coupled donation-response tensor factorization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 665–674.

[12] Hsu-Chao Lai, Philip S Yu, and Jiun-Long Huang. 2023. Learning the Co-evolution Process on Live Stream Platforms with Dual Self-attention for Next-topic Recommendations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1158–1167.

[13] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023).

[14] Fengqi Liang, Baigong Zheng, Liqin Zhao, Guorui Zhou, Qian Wang, and Yanan Niu. 2024. Ensure Timeliness and Accuracy: A Novel Sliding Window Data Stream Paradigm for Live Streaming Recommendation. *arXiv preprint arXiv:2402.14399* (2024).

[15] Jiahao Liang, Xiangyu Zhao, Muyang Li, Zijian Zhang, Wanyu Wang, Haochen Liu, and Zitao Liu. 2023. MMMLP: multi-modal multilayer perceptron for sequential recommendations. In *Proceedings of the ACM Web Conference 2023*. 1109–1117.

[16] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[17] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[18] Enrico Palumbo, Giuseppe Rizzo, and Raphaël Troncy. 2017. Entity2rec: Learning user-item relatedness from knowledge graphs for top-n item recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 32–36.

[19] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Elena Baralis, Michele Osella, and Enrico Ferro. 2018. Knowledge graph embeddings with node2vec for item recommendation. In *The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers 15*. Springer, 117–120.

[20] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[21] Jérémie Rappaz, Julian McAuley, and Karl Aberer. 2021. Recommendation on live-streaming platforms: Dynamic availability and repeat consumption. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 390–399.

[22] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*. 519–527.

[23] Chenguang Song, Nianwen Ning, Yunlei Zhang, and Bin Wu. 2021. A multimodal fake news detection model based on crossmodal attention residual and multichannel convolutional neural networks. *Information Processing & Management* 58, 1 (2021), 102437.

[24] 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.

[25] Wei Tu, Chen Yan, Yiping Yan, Xu Ding, and Lifeng Sun. 2018. Who is earning? Understanding and modeling the virtual gifts behavior of users in live streaming economy. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. IEEE, 118–123.

[26] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[27] 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).

[28] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[29] 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.

[30] 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.

[31] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[32] Dinghao Xi, Liumin Tang, Runyu Chen, and Wei Xu. 2023. A multimodal time-series method for gifting prediction in live streaming platforms. *Information Processing & Management* 60, 3 (2023), 103254.

[33] Shuaiyong Xiao, Gang Chen, Chenghong Zhang, and Xiangge Li. 2022. Complementary or substitutive? A novel deep learning method to leverage text-image interactions for multimodal review helpfulness prediction. *Expert Systems with Applications* 208 (2022), 118138.

[34] Xiaoyong Yang, Yadong Zhu, Yi Zhang, Xiaobo Wang, and Quan Yuan. 2020. Large scale product graph construction for recommendation in e-commerce. *arXiv preprint arXiv:2010.05525* (2020).

[35] Dung-Ru Yu, Chiao-Chuan Chu, Hsu-Chao Lai, and Jiun-Long Huang. 2020. Social Attentive Network for Live Stream Recommendation. In *Companion Proceedings of the Web Conference 2020*. 24–25.

[36] Sanshi Yu, Zhuoxuan Jiang, Dong-Dong Chen, Shanshan Feng, Dongsheng Li, Qi Liu, and Jinfeng Yi. 2021. Leveraging tripartite interaction information from live stream e-commerce for improving product recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3886–3894.

[37] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 283–292.

[38] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 1821–1830.

[39] Shuai Zhang, Hongyan Liu, Jun He, Sanpu Han, and Xiaoyong Du. 2021. A deep bi-directional prediction model for live streaming recommendation. *Information Processing & Management* 58, 2 (2021), 102453.

[40] Yujing Zhang, Zhangming Chan, Shuhao Xu, Weijie Bian, Shuguang Han, Hongbo Deng, and Bo Zheng. 2022. KEEP: An industrial pre-training framework for online recommendation via knowledge extraction and plugging. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3684–3693.

[41] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[42] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[43] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020).