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

Jiaxin Deng[1,2], Shiyao Wang[3], Yuchen Wang[3], Jiansong Qi[3], Liqin Zhao[3], Guorui Zhou[3*], Gaofeng Meng[1]

[1] MAIS, Institute of Automation, Chinese Academy of Science

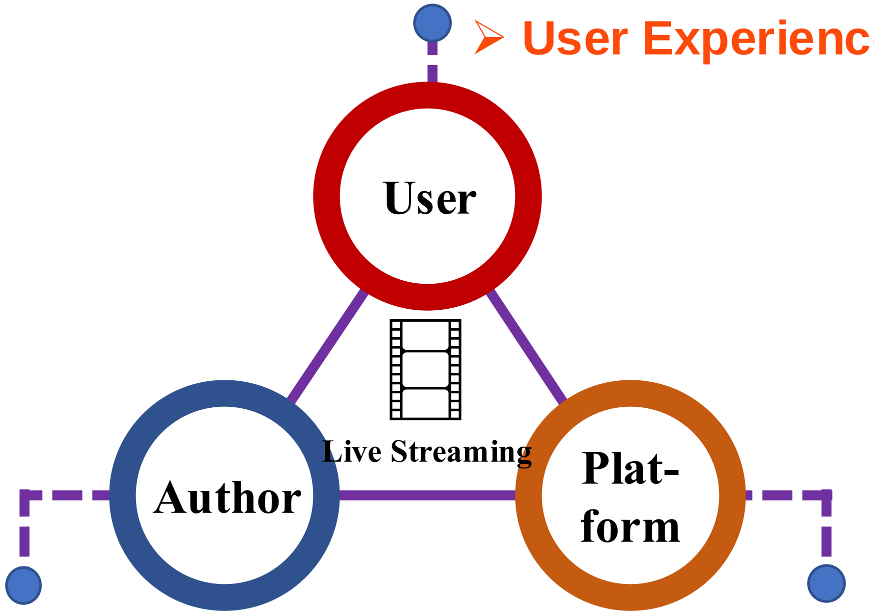
[2] University of Chinese Academy of Science

[3] Kuaishou Inc.

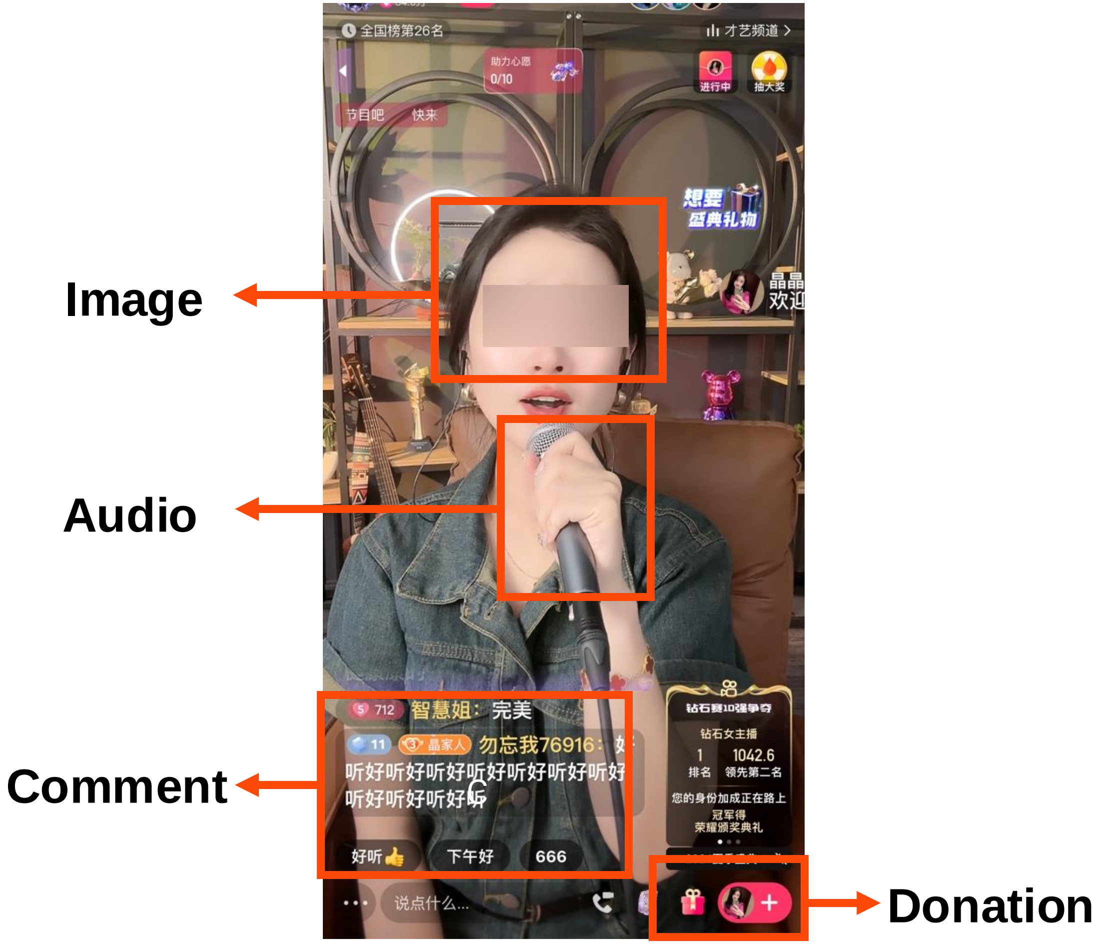
[*]Corresponding Author

Background: Live Streaming Gifting

Rapid Development of Various Information-sharing Platforms on the Internet



- ➢ User Experience
- ➢ More Fans
- ➢ More Reward
- ➢ User Stickiness
- ➢ Advertising Revenue



The Screenshot of the Kuaishou APP

Motivation

➢ Exiting Methods

- MARS[1] 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.

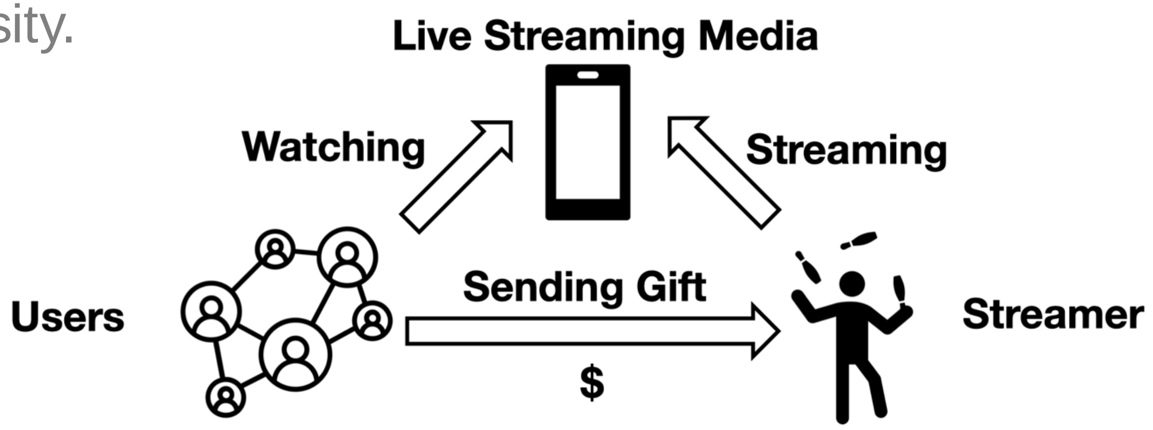
  It ignores the close connection between users' gifting behavior and the rapidly changing live content in the living room.
- MTA[2] designs a novel orthogonal module that fully utilizes the multi-modal features in live streaming.

  It treats the gift prediction as a time series prediction problem which does not consider users' personalization.
- SIM[3] leverages user behavior retrieval techniques to enhance the recommendation performance

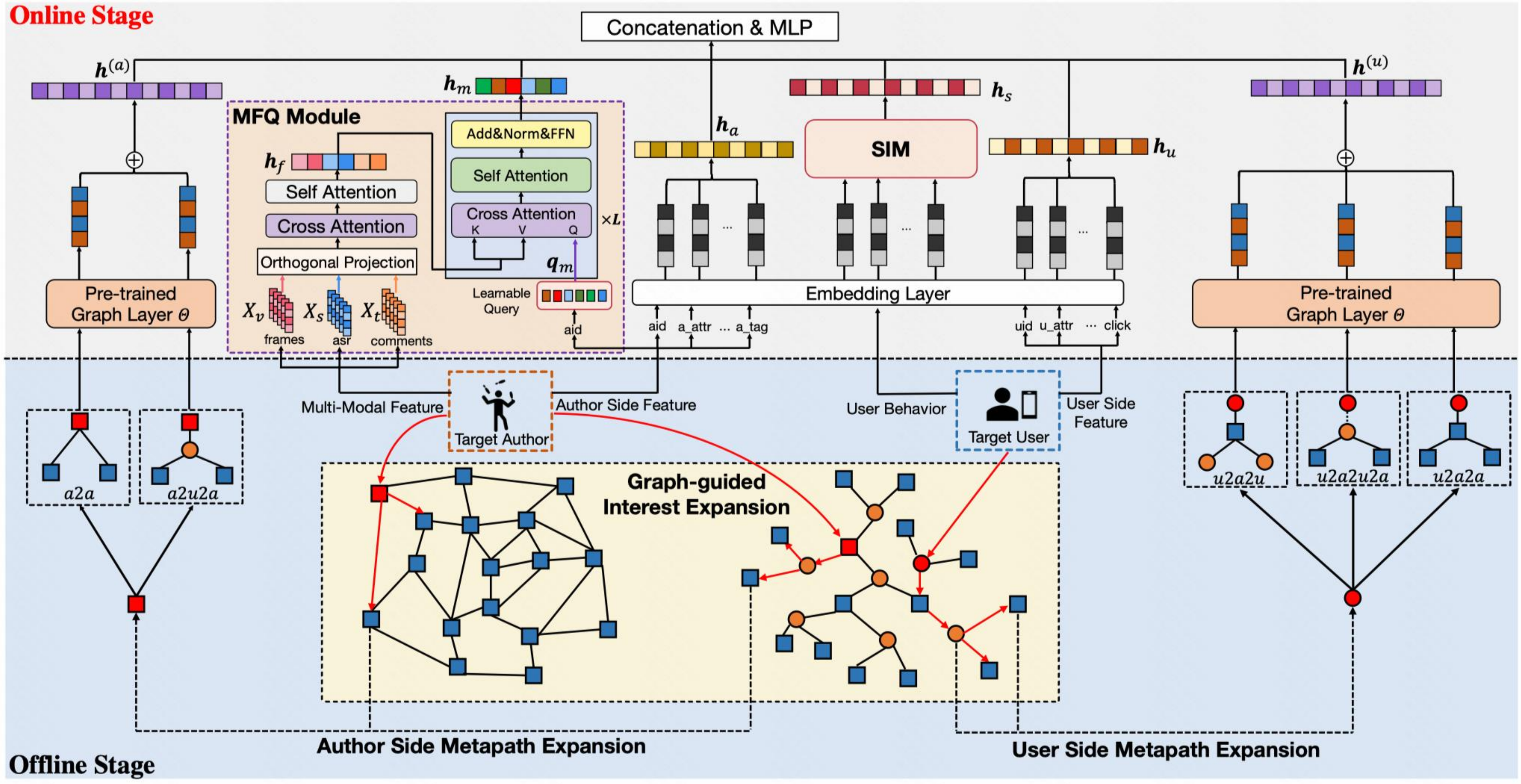
  It may face the challenge of gifting behavior sparsity.

➢ Two Challenges

- Precisely describe the real-time content changes in live streaming using limited categorical information.
- The the sparsity problem in gifting prediction.



Method: Detangled System Deployment



Method: Multi-modal Fusion with Learnable Query

➢ Author Side Feature

- We leverages the visual frames, ASR and comments feature.
- The orthogonal projection is proposed to maximize the complementation effects between different modalities.
- The learnable query from aid helps align the multimodal representations with the ID embedding

$$h_v = \text{CrossAttention}(X_v W_v^Q, Y_v W_v^K, Y_v W_v^V), Y_v = OP(X_v, X_s, X_t)$$

$$h_s = \text{CrossAttention}(X_s W_s^Q, Y_s W_s^K, Y_s W_s^V), Y_s = OP(X_s, X_t, X_v)$$

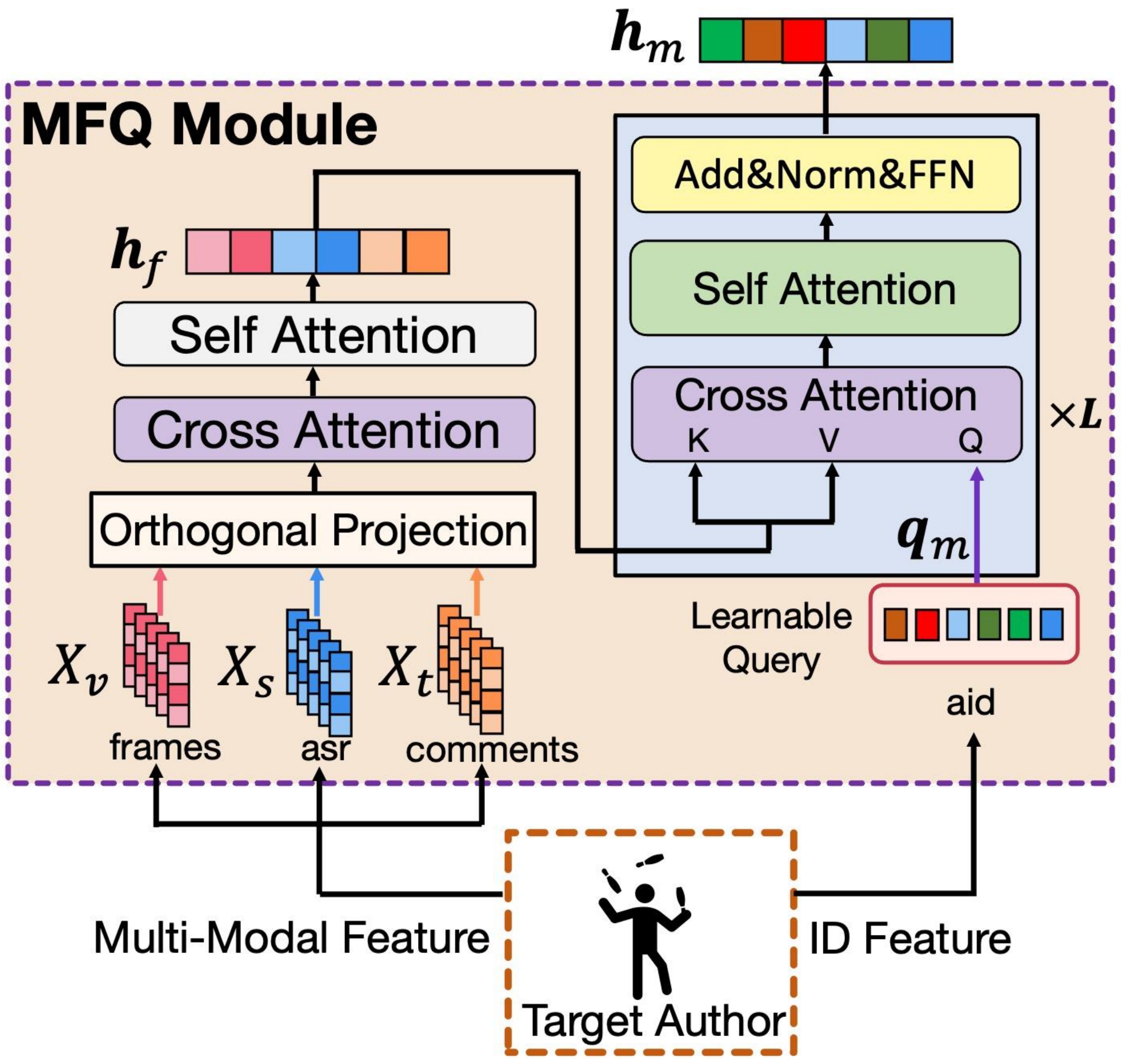
$$h_t = \text{CrossAttention}(X_t W_t^Q, Y_t W_t^K, Y_t W_t^V), Y_t = OP(X_t, X_s, X_v)$$

$$h_f' = h_v \oplus h_s \oplus h_t$$

$$h_f = \text{SelfAttention}(h_f' W_f^Q, h_f' W_f^K, h_f' W_f^V)$$

$$h_m' = \text{CrossAttention}(q_m W_c^Q, h_f W_c^K, h_f W_c^V)$$

$$h_m = \text{SelfAttention}(h_m' W_s^Q, h_m' W_s^K, h_m' W_s^V)$$



Method: Graph-guided Interest Expansion

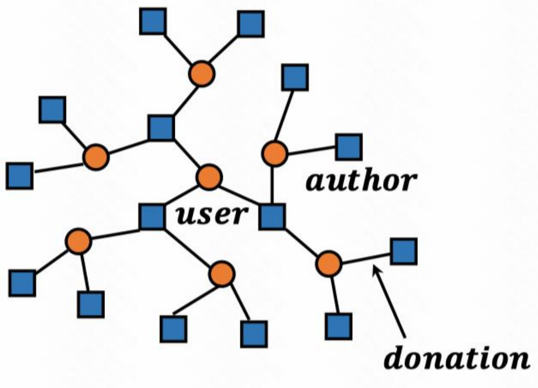
➢ User Side Feature

- We build two U2A and A2A graphs.
- We design five metapath-guided behavior expansion sequences through end-to-end training

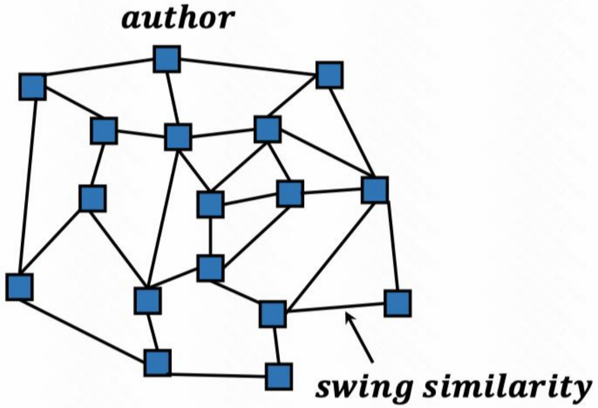
- $\mathcal{N}^{(2)}_{\rho_{u2a2u}}(u_t)$ begins with the target user u_t and follow this metapath. The retrieved behavior sequence is a set of users who share the same authors as the target user. Therefore, this metapath gets similar users who share the similar interests of the target user.
- $\mathcal{N}^{(3)}_{\rho_{u2a2u2a}}(u_t)$ 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.

Algorithm 1: GraphCL

```
Initialize L ← 0;
Graph G_1(V_u ∪ V_a, E_1), graph node embedding layers
  parameter Θ ∈ R^{|V_u ∪ V_a| × d}, walks epoch γ;
for i = 0 to γ do
    O = Shuffle(V_u ∪ V_a);
    for v_t ∈ O do
        V_p ← {}, V_n ← {};
        if v_t ∈ V_u then
            V_p ← N^{(2)}_{ρ_u2a2u}(v_t);
            V_n ← V_n ∪ RandomSample(V_u);
        end
        if v_t ∈ V_a then
            V_p ← N^{(2)}_{ρ_a2u2a}(v_t);
            V_n ← V_n ∪ RandomSample(V_a);
        end
    end
    L ← L_CE + λ L_NCE;
    Θ ← Θ − α ∂L/∂Θ;
end
Output: Trained graph node embedding layers parameter Θ
```



(a) User-to-Author

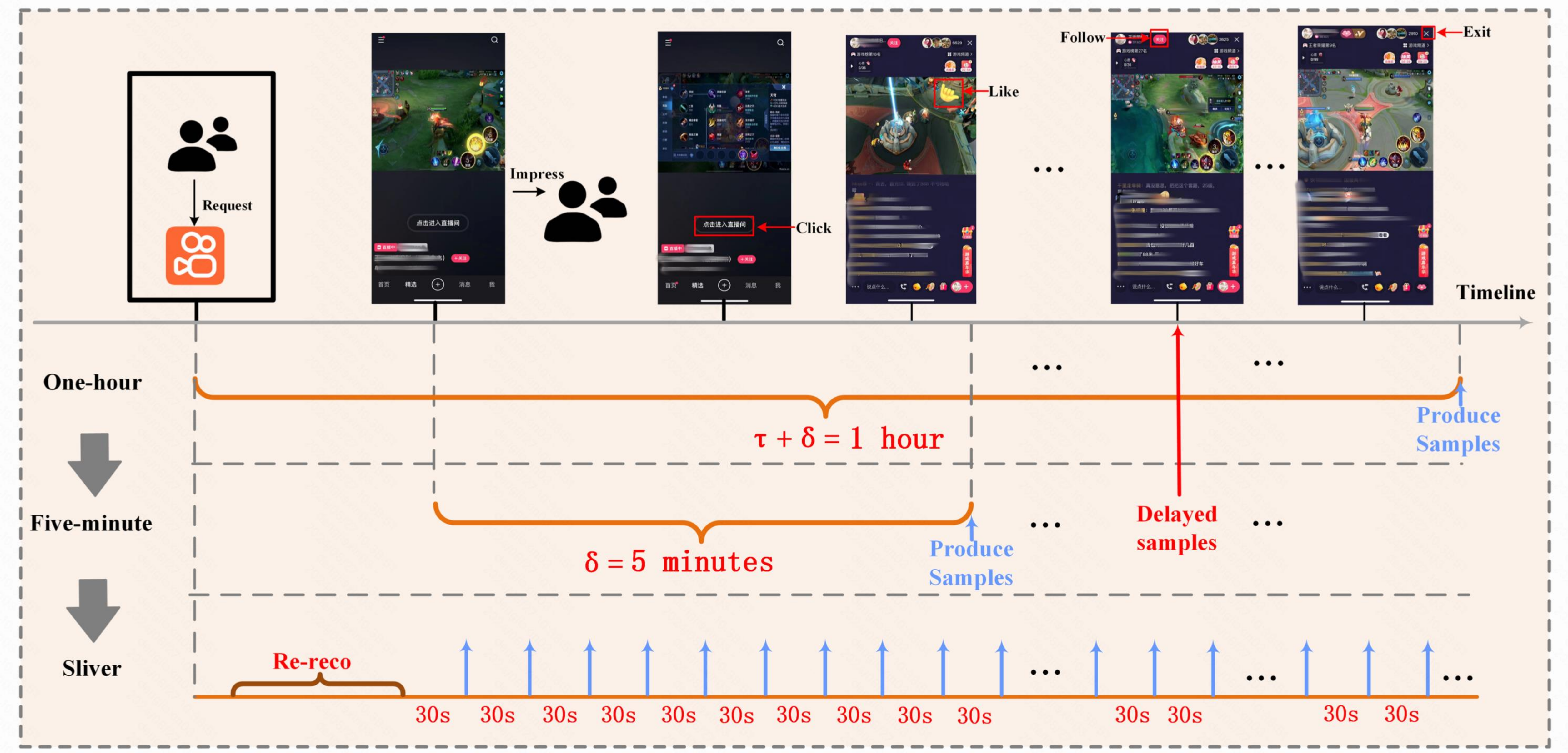


(b) Author-to-Author

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

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

Dataset: Live Streaming Sample Generation



Experiments: Results on Kuaishou dataset

Table 2: 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	$1.02e^{-3}$		$2.01e^{-3}$		$5.12e^{-3}$	

Experiments: Results on Tiktok and ML datasets

Table 3: 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]	<u>0.0569</u>	<u>0.0093</u>	<u>0.0330</u>	0.2155	<u>0.0524</u>	<u>0.1444</u>
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**	<u>0.2317</u>	**0.0566**	**0.1573**
p-value	$1.29e^{-5}$	$6.23e^{-6}$	$7.29e^{-5}$	$2.75e^{-5}$	$2.81e^{-3}$	$1.61e^{-2}$

Experiments: Ablation Study

Table 7: 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 %

Table 4: 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%
$X_v(-)$	-	√	√	-0.1101%	-0.2069%	-0.2939%
$X_s(-)$	√	-	√	-0.1090%	-0.1565%	-0.1383%
$X_t(-)$	√	√	-	-0.0839%	-0.0933%	-0.1790%

Table 6: 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

System Response Time Optimization

- We apply the pre-request of expansion behaviors and stored it in advance.
- The offline walk strategy significantly reduces the response time latency.

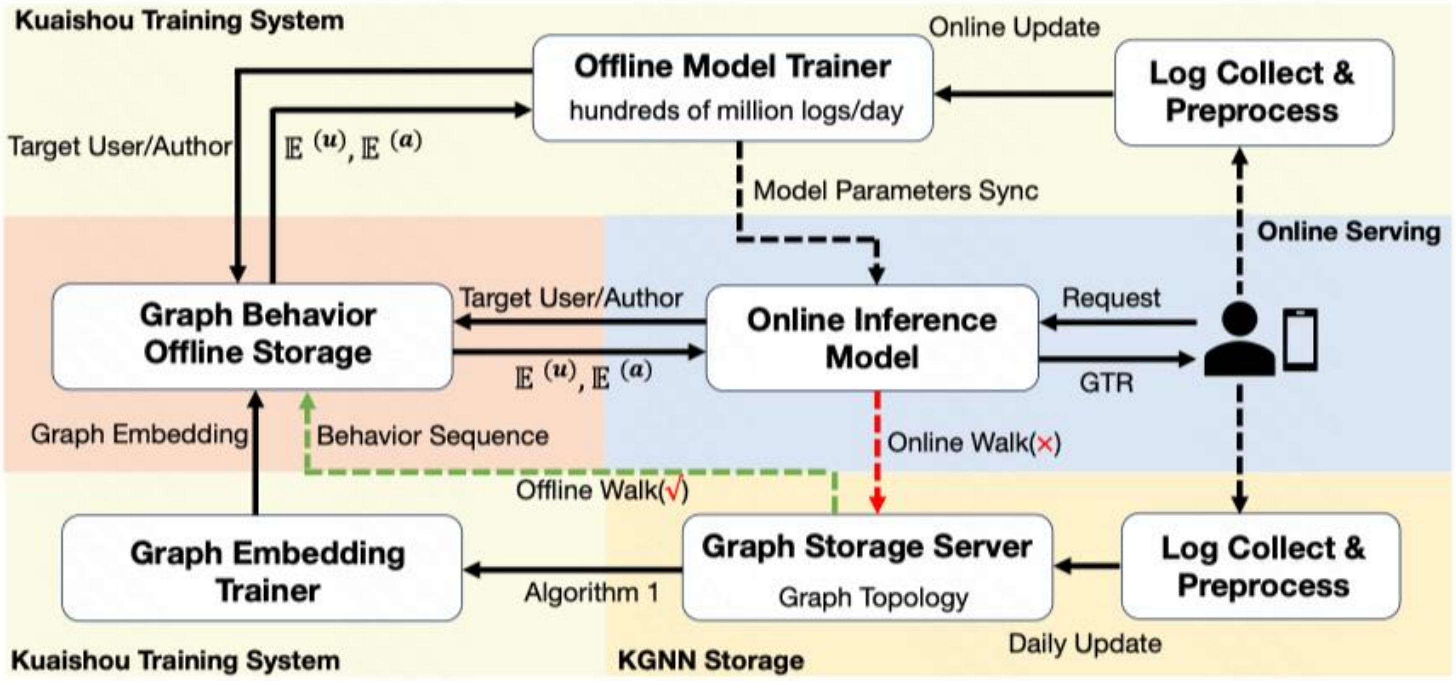
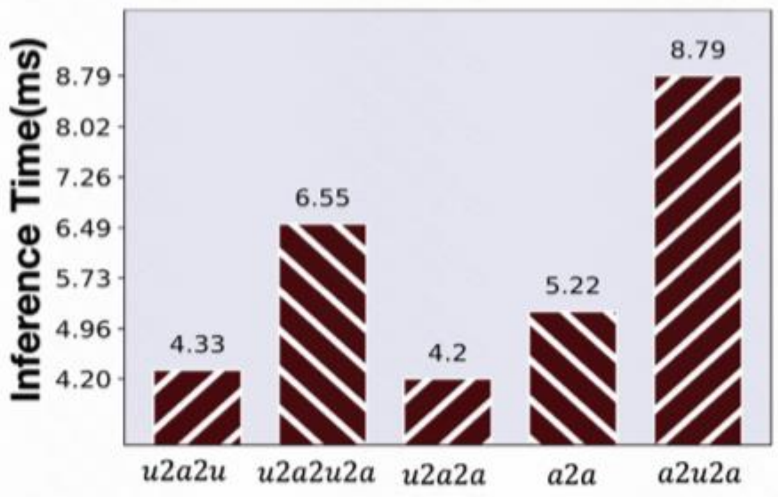
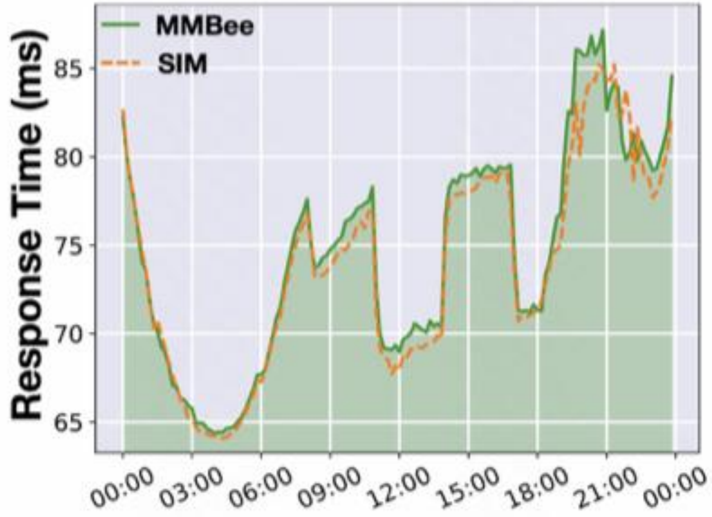


Figure 4: The deployment of MMBee in online live streaming GTR prediction system.





Thank You

- We proposed **Multi-modal Fusion with Learnable Query** (MFQ) module leverages the dynamic multimodal content of live streaming and captures the distinct characteristics among streamers.
- The proposed **Graph-guided Interest Expansion** (GIE) module largely enriches the observed history behaviors of users and streamers with both self-supervised graph representation learning and metapathbased 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.