

FROG: Effective Friend Recommendation in Online Games via Modality-aware User Preferences

Qiwei Wang*
Tencent
Shenzhen, China

Dandan Lin*
Shenzhen Institute of Computing
Sciences
Shenzhen, China
lindandan@sics.ac.cn

Wenqing Lin†
Ziming Wu
Tencent
Shenzhen, China
edwlin@me.com

Abstract

Due to the convenience of mobile devices, the online games have become an important part for user entertainments in reality, creating a demand for friend recommendation in online games. However, none of existing approaches can effectively incorporate the multi-modal user features (e.g., images and texts) with the structural information in the friendship graph, due to the following limitations: (1) some of them ignore the high-order structural proximity between users, (2) some fail to learn the pairwise relevance between users at modality-specific level, and (3) some cannot capture both the local and global user preferences on different modalities. By addressing these issues, in this paper, we propose an end-to-end model FROG that better models the user preferences on potential friends. Comprehensive experiments on both offline evaluation and online deployment at Tencent have demonstrated the superiority of FROG over existing approaches. The source code of this paper can be found at <https://github.com/socialalgo/FROG>.

CCS Concepts

• Information systems → Recommender systems.

Keywords

Recommender Systems, Social Network, Multi-Modal

ACM Reference Format:

Qiwei Wang, Dandan Lin, Wenqing Lin, and Ziming Wu. 2025. FROG: Effective Friend Recommendation in Online Games via Modality-aware User Preferences. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, July 13–18, 2025, Padua, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3726302.3730198>

1 Introduction

Due to the convenience of mobile devices, online games have become a significant component for user entertainments in reality [2, 8, 12–14, 16, 17, 19, 24–27]. In the online games, a player u might want to connect with the other players for the purpose of sociality to interact with interesting users, or gaming requirements that encourage players to play the games together [12, 13, 24, 27].

*Qiwei Wang and Dandan Lin are the co-first authors.

†Wenqing Lin is the corresponding author.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGIR '25, Padua, Italy*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1592-1/2025/07

<https://doi.org/10.1145/3726302.3730198>

However, it is difficult for player u to search among billions or millions of players in the online game platforms, which has prompted a need for *friend recommendation* in the online games. Specifically, given a user (i.e., player) u in an online game platform, the friend recommendation task aims to recommend a *short* list of new potential friends with whom u might be interested to have a connection. It also has been empirically verified that friend recommendation in the online games brings the growth of social network, and further increases the number of active interactions between users like game-playing or chatting [13, 27], leading to an increase of the total revenues in the game providers.

To tackle friend recommendation task, a straightforward approach is to utilize the natural graph structure behind the social networks where the nodes represent the users in a specific platform, and the edges between two nodes u and v denote that users u and v are friends in the platform. Based on the friendship graph, a plenty of *traditional proximity-based methods* can be adopted by firstly computing the user-user proximity scores based on the topological information, and then returning the top- k users that have the highest proximity scores with respect to (w.r.t) a given user u as the potential friends for user u , e.g., Personalized PageRank [10, 12], the common friends-based triadic closure principle [3] or the path-based Katz centrality [9]. These proximity-based methods are based on the rationale that two users are more likely to connect if they have many common friends. However, this kind of methods *fail* to consider the valuable **multi-modal** information in the online games, e.g., the user attributes and the in-game attributes, leading to inferiority in the performance. The multi-modal information has an important effect on users' decision to accept the recommendations. For example, a user whose profile image is a cartoon character is more likely to be interested in the users who have the cartoon profile image than those who have the profile image with natural scenery. Besides, advanced game players with high game level are more inclined to play with peers of comparable gaming ability, rather than those with vastly different game levels, as it leads to a more enjoyable gaming experience. Thus, it is equally important for friend recommendation to capture the user preferences by exploiting the multi-modal data.

Nevertheless, utilizing the multi-modal information for friend recommendation is not trivial due to the following two challenges.

Challenge 1: How to jointly process the multi-modal data of different scales? Multi-modality data can take various forms with different scales as they are collected from diverse domains. Although the use of user profile features has become pervasive in

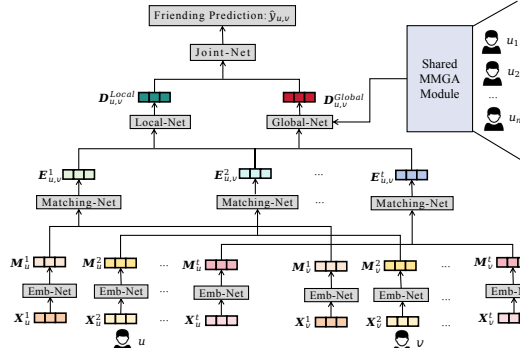


Figure 1: Framework of our proposed model FROG.

product/item recommendation applications by learning the interactions between input features explicitly or implicitly, these *feature-based methods* [4, 6, 15, 18, 22, 28], e.g., FM [18], DEEPFM [4] and AutoInt [22], struggle to deliver comparable results when applied to friend recommendation tasks, since they not only *ignore the high-order structural proximity between users* revealed from the friendship graph, but also *fail to inject modality-aware signals*. While the state-of-the-art approach GRAFRANK [20] for friend recommendation exploits Graph Neural Network (GNN) to fuse the structural information, it *cannot effectively capture the pairwise relevance between users* since it focuses on the updating of individual user embedding.

Challenge 2: How to effectively discriminate the effect of each modality for personalized recommendation? Each modality contributes *differently* to a successful friend recommendation due to the following two reasons. (i) The information contained within each modality is distinct, leading to significant variations in their discriminative capabilities. (ii) The user preferences towards different modalities are *highly personalized*, e.g., some users might focus on the profile images only, while some might be interested in the game-playing histories. Thus, it necessitates an effective methodology to discern the attentions of each user on each modality. Although previous work GRAFRANK [20] aggregates the neighbor information in a modality-specific manner, it *misses the global preferences* on different modality. The case becomes more challenging when combining the personalized attentions on modalities with the pairwise relevance between users.

To tackle these issues, we consider to develop an effective model for multi-modal friend recommendation in online games that have the following three abilities simultaneously. (1) **End-to-end learning (EE)**. The model learning should be end-to-end to generate the probability that any pair users will be friends for final friend recommendation, instead of emphasis on individual user embeddings. (2) **Modality-aware pairwise learning with high-order topological information (MP)**. The model can learn the implicit relationships between users at the fine-grained modality-specific level by incorporating the high-order structural proximity revealed in the friendship graph. (3) **Holistic personalized learning (HP)**. The model can learn the personalized user attentions on multi-modalities from both the *local* and *global* views. To the best of our knowledge, all of existing approaches in friend/item recommendation scenarios fail to satisfy above three key abilities *simultaneously*. In this paper, we propose a novel end-to-end model for multi-modal friend recommendation in online games, termed FROG, that can fulfill the

above three capabilities simultaneously. FROG has been deployed in the online games in Tencent and supports various friend recommendation scenarios. The following shows our **contributions**.

- We devise a Matching-Net inside FROG to learn the pairwise relationship at the fine-grained modality-specific level.
- We combine a Local-Net and a Global-Net inside FROG for holistic personalized learning.
- Comprehensive experiments demonstrated that the proposed model FROG significantly outperforms the state-of-the-art methods on two real datasets **by up to 15.82% and 14.59%** in terms of Hit-Rate and NDCG, respectively.
- We have developed FROG in a friend recommendation scenario of an online game in Tencent and conducted online A/B test to show its superiority over existing approaches.

2 Problem Definition

Given an online game with a massive number of users, let V be the set of all users in the game. In the online game, each user $u \in V$ has a profile that contains the in-game attributes, such as game levels, personal descriptions, and profile images. Let $G(V, E)$ denote the friendship graph, where V is the set of users and $E \subseteq V \times V$ is the set of friendship edges between users in G . In this paper, we only consider the *static* friendship graph that is captured before the daily model training phase. Let n denote the number of users in the game platform before the daily model training.

Let t be the number of modalities used for user features. The multi-modal data of u is denoted by $X_u = [X_u^1, X_u^2, \dots, X_u^i, \dots, X_u^t]$, where X_u^i is the data from the i -th modality of user u and i is an integer in $[1, t]$. The data from each modality might have different forms, e.g., a vector or a graph (which will be elaborated in Section 3). We formally define the problem of *multi-modal friend recommendation* in a large-scale game platform as follows:

Definition 1 (Multi-Modal Friend Recommendation). Given the multi-modal user features X of all users in V , and a user $u \in V$, it recommends a list L of k users from V , such that the probability $\hat{y}_{u,v}$ of a user $v \in L$ that u would establish a new friendship with v is larger than other users that are not in the recommended list L , i.e., $\hat{y}_{u,v} > \hat{y}_{u,w}$ for all $v \in L$ and $w \notin L$.

3 Proposed Method

3.1 Overview

Given two users u and v , and their multi-modal data X_u and X_v , our FROG generates the friending probability $\hat{y}_{u,v}$ that u would establish a new friendship with v . Figure 1 shows the overall framework of our FROG that consists of five key components, i.e., (1) the Emb-Net, (2) the Matching-Net, (3) the Local-Net, (4) the Global-Net, and (5) the Joint-Net. To be specific, FROG firstly feeds the multi-modal data X_u (w.r.t X_v) into the Emb-Net to transform each modality of X_u (w.r.t X_v) into a unified representation for the subsequent steps. Secondly, the obtained representations M_u (w.r.t M_v) are coupled in the Matching-Net to learn the pairwise implicit similarity between u and v at the modality-specific level. Specifically, it jointly learns the mutual relevance between u and v for each modality, instead of regarding the modality data of each user individually. After that, the Local-Net and the Global-Net are processed simultaneously where the Local-Net facilitates the personalized preference

of user u on user v and the Global-Net utilizes a shared multi-modal global attention mechanism (MMGA) to capture the global information of all users. Finally, the obtained local and global information is fused in the Joint-Net, which produces the friending prediction.

3.2 Details of FROG

In this section, we elaborate the five key components one by one.

(1) **Emb-Net.** The Emb-Net processes the multi-modality features into a unified representation. Specifically, given the multi-modal data X_u of a user u , for the i -th modality X_u^i , it generates a modality-specific embedding M_u^i by considering the properties of the specific modality. After projecting each modality-specific embedding into a d -dimensional vector via three full-connected layers, it yields \mathcal{M}_u as a new representation of u by concatenating all vectors together, i.e., $\mathcal{M}_u = \{M_u^1, \dots, M_u^i, \dots, M_u^t\}$ where M_u^i is a $1 \times d$ vector. Due to space limit, more details about how features in each modality are generated could be found in our technical report [23].

(2) **Matching-Net.** To satisfy the capability of modality-aware pairwise learning, the Matching-Net module jointly learns the pairwise similarity between two users on a specific modality. Specifically, given a pair of two users u and v , and their representations \mathcal{M}_u and \mathcal{M}_v returned by the Emb-Net, the Matching-Net outputs a set $\mathcal{E}_{u,v}$ that contains the pairwise similarity embedding $\mathbf{E}_{u,v}^i$ between two users w.r.t the i -th modality, i.e., $\mathcal{E}_{u,v} = \{\mathbf{E}_{u,v}^1, \dots, \mathbf{E}_{u,v}^i, \dots, \mathbf{E}_{u,v}^t\}$.

A straightforward approach to learn the pairwise similarity between two users is to compute the similarity between two representations \mathcal{M}_u and \mathcal{M}_v by using traditional \mathcal{L}^p -norm-based similarity measure, e.g. Euclidean Distance or Inner Product. However, this approach fails to discover the difference of mutual similarity. That is, the similarity score of user u w.r.t v is different from that of user v w.r.t u since u and v might have different preferences in making friends. To solve this problem, our Matching-Net module computes the pairwise similarity embedding by considering the mutual similarity. Formally, $\mathbf{E}_{u,v}^i$ is computed as follows:

$$\mathbf{E}_{u,v}^i = M_u^i \circ R_{u \rightarrow v}^i + M_v^i \circ R_{v \rightarrow u}^i, \quad (1)$$

where $R_{u \rightarrow v}^i$ and $R_{v \rightarrow u}^i$ are the $1 \times d$ relevance vectors of v w.r.t u and u w.r.t v on the i -th modality, respectively, and \circ is Hadamard product that performs the element-wise multiplication of two vectors.

Computation of relevance vectors. To avoid being trapped in the local-level data, we utilize the attention mechanism to better model the mutual similarity. Specifically, the relevance vectors $R_{u \rightarrow v}^i$ and $R_{v \rightarrow u}^i$ are computed in three steps. Firstly, we measure the attention values C_u^i (w.r.t C_v^i) of the i -th modality representation M_u^i (w.r.t M_v^i) as below:

$$C_u^i = P_u^i M_u^i Q_v^i \text{ and } C_v^i = P_v^i M_v^i Q_u^i, \quad (2)$$

where P_u^i , Q_u^i , P_v^i and Q_v^i are learnable parameters, P_u^i and P_v^i are $d \times 1$ vectors, Q_u^i and Q_v^i are $d \times d$ matrices. Notice that P_u^i and Q_u^i work together to refine the granularity of the relationships inherent in M_u^i . Similarly for P_v^i and Q_v^i . Secondly, the affinity matrix $G_{u,v}^i$ between u and v is computed by using the tanh function:

$$G_{u,v}^i = \tanh \left((C_u^i)^T C_v^i \right). \quad (3)$$

Finally, the relevance vectors are calculated by using the mean-pooling on the affinity matrix:

$$R_{u \rightarrow v}^i = \sigma_1 \left(\text{RowMean} \left(G_{u,v}^i \right) \right)^T, \quad (4)$$

Dataset	#Training	#Validation	#Testing	#Total
Game1	697,979	174,495	3,210,833	4,083,307
Game2	1,085,124	271,281	8,986,352	10,342,757

Table 1: Statistics of the datasets.

$$R_{v \rightarrow u}^i = \sigma_1 \left(\text{ColMean} \left(G_{u,v}^i \right) \right), \quad (5)$$

where $\sigma_1(\cdot)$ is an activation function.

(3) **Local-Net.** The Local-Net module explores the extent of how user u is interested to be friends with user v by considering the pairwise similarity from the perspective of each modality. Specifically, given the set $\mathcal{E}_{u,v}$ of pairwise similarity embeddings, a simple multi-layers perceptron (MLP) [6] is used to generate the local personalized preference as follows:

$$D_{u,v}^{local} = \text{MLP} \left(\phi_{1 \leq i \leq t} (\mathbf{E}_{u,v}^i) \right), \quad (6)$$

where ϕ is a element-wise concatenation function and the output of $\phi_{1 \leq i \leq t} (\mathbf{E}_{u,v}^i)$ is a $1 \times td$ vector. Since MLP can handle the interaction of features in the input data automatically, the locally discriminative information can be well observed.

(4) **Global-Net.** However, Local-Net learns the local personalized preference from a single training instance, which misses the global preference on different modalities. For example, the image data might be more important in the success of recommendation than the textual data, which can be revealed by all users. To address this issue, we propose the Global-Net module to inject the global user preferences on different modals by using a shared MMGA mechanism. To be specific, we use a trainable $1 \times d$ vector A to represent a global training sample decision plane, namely, A is treated as a global key shared with all training samples. Formally, given the set $\mathcal{E}_{u,v}$ of pairwise similarity embeddings, it calculates the global preference $D_{u,v}^{global}$ by projecting $\mathcal{E}_{u,v}$ on the global sample decision plane A as follows:

$$D_{u,v}^{global} = \left(\sum_{1 \leq i \leq t} \mathbf{E}_{u,v}^i A^T \right) A. \quad (7)$$

With this paradigm, the global information provided by all users is integrated into the attention mechanism, augmenting the model with the capability of capturing the global pattern.

(5) **Joint-Net.** After obtaining both the local and global user preferences, the Joint-Net module is used to generate the friending probability $\hat{y}_{u,v}$ that u would like to be friends with v , as follows:

$$\hat{y}_{u,v} = \sigma_3 \left(W_2 \cdot \sigma_2 \left(W_1 \cdot \phi(D_{u,v}^{local}, D_{u,v}^{global}) + b_2 \right) + b_1 \right), \quad (8)$$

where σ_2 and σ_3 are activation functions, W_1 and W_2 are the trainable weights while b_1 and b_2 are trainable biases.

3.3 Loss Function and Complexity Analysis

In online games, some users might accept or turn down several the friend recommendations, which should be well distinguished. To facilitate that, we exploit the FocalLoss [11] to construct the loss of prediction $\hat{y}_{u,v}$, as follows:

$$L = -\alpha (1 - \hat{y})^\gamma \log(\hat{y}) y - (1 - \alpha) \hat{y}^\gamma \log(1 - \hat{y}) (1 - y), \quad (9)$$

where y is the true label, \hat{y} represents $\hat{y}_{u,v}$, $\alpha \in [0, 1]$ and $\gamma \geq 0$ are hyper-parameters.

Complexity. Let T_i be the unit cost for computing M_u^i by Emb-Net for the i -th modality for user u . FROG takes $O(\sum_{i=0}^t T_i + td^3 + (td)^2 h + td + (h + d)^2)$ total time for each pair where h is the number of dimensions of $D_{u,v}^{local}$. See the proof in technical report [23].

Model	Game1						Game2					
	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20
LR	0.2032	0.2534	0.2382	0.2663	0.2469	0.2777	0.0166	0.0124	0.0189	0.0131	0.0402	0.0210
MLP	0.2043	0.254	0.2395	0.2673	0.2483	0.2779	0.0167	0.0121	0.0191	0.0128	0.0406	0.0215
FM	0.209	0.2555	0.2404	0.2661	0.2477	0.2769	0.0145	0.0106	0.0163	0.0111	0.0357	0.0188
DEEPM	0.2067	0.2549	0.2401	0.2666	0.2487	0.2781	0.0149	0.012	0.0169	0.0126	0.0380	0.0208
AUTOINT	0.2082	0.2558	0.2405	0.2669	0.2476	0.2782	0.0188	0.0148	0.0212	0.0155	0.0444	0.0237
DMF	0.2065	0.2543	0.2400	0.2654	0.2465	0.2772	0.0154	0.0115	0.0174	0.0121	0.0364	0.0198
SAGE+MAX	0.2056	0.2573	0.2397	0.2656	0.2470	0.2772	0.0173	0.0136	0.0196	0.0143	0.0438	0.0225
SAGE+MEAN	0.2074	0.2553	0.2404	0.2667	0.2475	0.2779	0.0176	0.0141	0.0199	0.0148	0.0439	0.0225
GRAFRANK	0.2088	0.2559	0.2400	0.2659	0.2475	0.2778	0.0182	0.0144	0.0204	0.0151	0.0438	0.0235
EBR	0.209	0.2562	0.2406	0.2668	0.2477	0.2781	0.0181	0.0142	0.0204	0.0149	0.0441	0.0235
FROG	0.2179	0.2637	0.2495*	0.2715*	0.2582*	0.2885*	0.0219	0.0163	0.0245*	0.0171*	0.0493*	0.0271*
Improvement	4.26%	2.79%	3.70%	1.57%	3.82%	3.70%	16.49%	10.14%	15.81%	10.32%	11.08%	14.59%

Table 2: Recommendation results of evaluated methods. The best and second-best results of each metric are highlighted in a bold font and underlined, respectively. The improvement is computed as the gains of the best result over the second-best result.

Model	Game1					
	HR@5	NDCG@5	HR@10	NDCG@10	HR@20	NDCG@20
WITHOUT Matching-Net	0.2129 (2.3%↓)	0.2587 (1.9%↓)	0.2415 (3.31%↓)	0.2669 (1.72%↓)	0.2487 (3.82%↓)	0.2787 (3.52%↓)
WITHOUT Local-Net	0.2113 (3.1%↓)	0.2553 (3.3%↓)	0.2395 (4.18%↓)	0.2657 (2.18%↓)	0.2471 (4.49%↓)	0.2774 (4.00%↓)
WITHOUT Global-Net	0.2127 (2.4%↓)	0.2572 (2.5%↓)	0.2407 (3.66%↓)	0.2673 (1.57%↓)	0.2489 (3.74%↓)	0.2777 (3.89%↓)
FROG	0.2179	0.2637	0.2495	0.2715	0.2582	0.2885

Table 3: The effect of each module in FROG for friend recommendation performance on Game1 dataset.

Model	accRate	intRate
AUTOINT	42.86%	50.01%
FROG (Ours)	57.14%	62.51%

Table 4: Results of A/B testing in an online game of Tencent.

4 Experiments

4.1 Experiment Setup

Datasets. We used two *real* datasets [23] collected from Tencent: (1) Game1 and (2) Game2. Table 1 shows the statistics of each dataset. **Baselines.** We compared our method with *eleven* baselines including *Logistic Regression* (LR) [6], *MLP* [6] that consists of three fully-connected layers, *Factorization Machine* (FM) [18], *DEEPM* [4], *AUTOINT* [22], *AUTOFIS+DEEPM* [15] that removes redundant feature intersections inside DEEPM, *DMF* [7], *SAGE+MAX* [5] that uses the element-wise *max pooling* in GRAPH-SAGE model, *SAGE+MEAN* [5] that uses the element-wise *sum-pooling* in GRAPH-SAGE model, *GRAFRANK* [20] and *EBR* [21].

Default parameters. We exploit the Adam optimizer for training the models. Besides, we set the learning rate as 0.001, the max epochs as 50, and the batch size as 1024. For activation functions used in FROG, σ_1 , σ_2 and σ_3 are the softmax ReLU and Sigmoid functions, respectively. For a fair comparison, we used the same embeddings of multi-modal data obtained by Emb-Net and the same training strategy for both our model and the competitors. Besides, for each evaluated method, we choose the model that performs the best in the validation set, to be evaluated for the testing set. Source code of this paper can be found at [1].

Environment. We run the experiments on a machine with a Tesla V100 GPU, 22 CPU cores, and 90 GB shared CPU memory. We implemented each evaluated method by using TensorFlow.

Evaluation metrics. Following previous work [20], we used two widely-used metrics to evaluate the performance of the proposed approach, i.e., *Hit-Rate* (HR@ k) and *Normalized Discounted Cumulative Gain* (NDCG@ k), where k is varied from {5, 10, 20}. For each experiment, we repeated 5 times and reported the average results.

4.2 Experimental Results

Overall performance. Table 2 shows the results on two datasets where the best and second-best results of each metric are highlighted in a bold font and underlined, respectively. From the results,

we can see that our proposed model FROG achieves the best performance on all datasets when k is varied from 5 to 20. To be specific, on the largest dataset Game2, FROG has better recommendation performance than the second-best baseline AUTOINT by up to 15.82% and 14.59% in terms of HR and NDCG, respectively. It is because FROG considers both the pairwise modality-aware signals between users while AUTOINT fails. Moreover, compared with GRAFRANK that utilizes both the multi-modalities and the social topology information, the performance of FROG is higher no matter how k is changed, showing the effectiveness of FROG by considering both the local and global personalized user preferences on different modalities.

Ablation study. We evaluated the effect of each module in FROG with its three degraded variants. Table 3 shows the results on the Game1 dataset by varying k from 5 to 20. It demonstrates that each module has the essential influence on the friend recommendation, *namely*, FROG using all three modules has the best performance by up to 4.49% and 4.00% in terms of HR and NDCG, respectively.

Online deployment. We deployed it in an online *First Personal Shooter* (FPS) game of Tencent, which is a multiplayer online game with billions of users. We selected AUTOINT as the competitor since it achieves the second-best results in most of cases. Due to space limit, the deployment settings could be found in [23]. We used two evaluation metrics *i.e.*, the *acceptance rate* (accRate) that user accepts the recommendations and *interaction rate* (intRate) that user interacts with the newly-connected friends after acceptance. Due to space limit, the formal definitions are in technical report [23]. Table 4 shows the results of A/B tests. FROG outperforms AUTOINT by up to 33.3% and 25.5% in terms of both accRate and intRate, respectively, showing the effectiveness of FROG in reality.

5 Conclusion

In this paper, we propose an end-to-end model FROG for multi-modal friend recommendation in online games. The model focuses on the modality-aware pairwise learning from both local and global user preferences by utilizing the multi-modal user features. Comprehensive experiments have demonstrated its effectiveness for friend recommendation scenarios in real-world online games.

References

- [1] 2024. Source codes of FROG. <https://github.com/socialalgo/FROG>.
- [2] Linah Aburahmah, Hajar AlRawi, Yamamah Izz, and Liyakathunisa Syed. 2016. Online social gaming and social networking sites. *Procedia Computer Science* 82 (2016), 72–79.
- [3] David Easley, Jon Kleinberg, et al. 2012. Networks, crowds, and markets. *Cambridge Books* (2012).
- [4] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 1725–1731.
- [5] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *NeurIPS* 30 (2017), 1024–1034.
- [6] Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- [7] Di Hu, Chengze Wang, Feiping Nie, and Xuelong Li. 2019. Dense multimodal fusion for hierarchically joint representation. In *ICASSP*. IEEE, 3941–3945.
- [8] Melinda Jacobs and Tanja Sihvonen. 2011. In perpetual beta? On the participatory design of Facebook games. In *Proceedings of DiGRA 2011 Conference: Think Design Play*.
- [9] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.
- [10] Dandan Lin, Raymond Chi-Wing Wong, Min Xie, and Victor Junqiu Wei. 2020. Index-free approach with theoretical guarantee for efficient random walk with restart query. In *ICDE*. 913–924.
- [11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *ICCV*. 2980–2988.
- [12] Wenqing Lin. 2019. Distributed Algorithms for Fully Personalized PageRank on Large Graphs. In *WWW*. 1084–1094.
- [13] Wenqing Lin. 2021. Large-Scale Network Embedding in Apache Spark. In *SIGKDD*. 3271–3279.
- [14] Wenqing Lin, Feng He, Faqiang Zhang, Xu Cheng, and Hongyun Cai. 2020. Initialization for Network Embedding: A Graph Partition Approach. In *WSDM*. 367–374.
- [15] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincal Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *SIGKDD*. 2636–2645.
- [16] Chang Liu, Qiwei Wang, Wenqing Lin, Yue Ding, and Hongtao Lu. 2024. Beyond Binary Preference: Leveraging Bayesian Approaches for Joint Optimization of Ranking and Calibration. In *SIGKDD*. ACM, 5442–5453.
- [17] Chang Liu, Yuwen Yang, Yue Ding, Hongtao Lu, Wenqing Lin, Ziming Wu, and Wendong Bi. 2024. DAG: Deep Adaptive and Generative K-Free Community Detection on Attributed Graphs. In *SIGKDD*. ACM, 5454–5465.
- [18] Steffen Rendle. 2010. Factorization machines. In *ICDM*. 995–1000.
- [19] Luca Rossi. 2010. Playing your network: gaming in social network sites. *Available at SSRN 1722185* (2010).
- [20] Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. 2021. Graph neural networks for friend ranking in large-scale social platforms. In *WWW*. 2535–2546.
- [21] Jiahui Shi, Vivek Chaurasiya, Yozen Liu, Shubham Vij, Yan Wu, Satya Kanduri, Neil Shah, Peicheng Yu, Nik Srivastava, Lei Shi, Ganesh Venkataraman, and Jun Yu. 2023. Embedding Based Retrieval in Friend Recommendation. In *SIGIR*. 3330–3334.
- [22] 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 *CIKM*. 1161–1170.
- [23] Qiwei Wang, Dandan Lin, Wenqing Lin, and Ziming Wu. 2025. FROG: Effective Friend Recommendation in Online Games via Modality-aware User Preferences. arXiv:2504.09428 [cs.SI] <https://arxiv.org/abs/2504.09428>
- [24] Shiqi Zhang, Yiqian Huang, Jiachen Sun, Wenqing Lin, Xiaokui Xiao, and Bo Tang. 2023. Capacity Constrained Influence Maximization in Social Networks. In *SIGKDD*. 3376–3385.
- [25] Shiqi Zhang, Jiachen Sun, Wenqing Lin, Xiaokui Xiao, Yiqian Huang, and Bo Tang. 2024. Information Diffusion Meets Invitation Mechanism. In *WWW*. ACM, 383–392.
- [26] Shiqi Zhang, Jiachen Sun, Wenqing Lin, Xiaokui Xiao, and Bo Tang. 2022. Measuring Friendship Closeness: A Perspective of Social Identity Theory. In *CIKM*. ACM, 3664–3673.
- [27] Xingyi Zhang, Shuliang Xu, Wenqing Lin, and Sibao Wang. 2023. Constrained Social Community Recommendation. In *SIGKDD*. 5586–5596.
- [28] Jieming Zhu, Qinglin Jia, Guohao Cai, Quanyu Dai, Jingjie Li, Zhenhua Dong, Ruiming Tang, and Rui Zhang. 2023. Final: Factorized interaction layer for ctr prediction. In *SIGIR*. 2006–2010.