

Large-scale Personalized Video Game Recommendation via Social-aware Contextualized Graph Neural Network

Liangwei Yang

lyang84@uic.edu

University of Illinois at Chicago
USA

Chen Wang

cwang266@uic.edu

University of Illinois at Chicago
USA

Zhiwei Liu

zhiweiliu@salesforce.com

Salesforce Research
USA

Ziwei Fan

zfan20@uic.edu

University of Illinois at Chicago
USA

Yu Wang

ywang617@uic.edu

University of Illinois at Chicago
USA

Philip S. Yu

psyu@cs.uic.edu

University of Illinois at Chicago
USA

ABSTRACT

Because of the large number of online games available nowadays, online game recommender systems are necessary for users and online game platforms. The former can discover more potential online games of their interests, and the latter can attract users to dwell longer in the platform. This paper investigates the characteristics of user behaviors with respect to the online games on the Steam platform. Based on the observations, we argue that a satisfying recommender system for online games is able to characterize: personalization, game contextualization and social connection. However, simultaneously solving all is rather challenging for game recommendation. Firstly, personalization for game recommendation requires the incorporation of the dwelling time of engaged games, which are ignored in existing methods. Secondly, game contextualization should reflect the complex and high-order properties of those relations. Last but not least, it is problematic to use social connections directly for game recommendations due to the massive noise within social connections. To this end, we propose a Social-aware Contextualized Graph Neural Recommender System (SCGRec), which harnesses three perspectives to improve game recommendation. We conduct a comprehensive analysis of users' online game behaviors, which motivates the necessity of handling those three characteristics in the online game recommendation.

CCS CONCEPTS

- Information systems → Personalization; Recommender systems; Collaborative filtering.

KEYWORDS

Game Recommendation, Graph Neural Network, Personalization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9096-5/22/04...\$15.00

<https://doi.org/10.1145/3485447.3512273>

ACM Reference Format:

Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S. Yu. 2022. Large-scale Personalized Video Game Recommendation via Social-aware Contextualized Graph Neural Network. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3485447.3512273>

1 INTRODUCTION

Recent years have witnessed the rapid growth of the video game industry. According to statistics¹, the global video game market value in 2020 is \$155.89 billion and is expected to reach \$268.81 billion in 2025. Along with the rising of the market, values are massive new releasing video games. It is reported that on Steam platform², 10,263 new games are released in 2020 while the number was 276 in 2010³. Due to the time limitation of users in playing video games, a game recommender system is necessary to assist users to discover new games of their potential interests. In this way, we can improve users' experience in-game engagement and can thus attract players to stay longer within the platform.

Though designing game recommender systems attracts recent research attentions [1, 3, 6], they are still under-explored. A satisfying game recommender system should be able to characterize the following three perspectives: 1) **personalization**, modelling the game engagement interests of users, 2) **game contextualization**, revealing complex relations among games and 3) **social connection**, interpreting the game engagement of users as a type of social activities. Though these factors have already been investigated in other recommendation scenarios [8–10, 18–20, 42, 55], directly adopting existing methods from other domains is problematic due to the unique challenges in tackling game recommendations.

Firstly, the personalization of users should reveal their interests in games, which requires the awareness of game engagements. A comprehensive game engagements should incorporate: 1) *which games* users have engaged in and 2) *how long* users dwell in those games [50]. The former is widely tackled as the collaborative signals [5, 14] between users and games, while the latter is seldom studied. On one hand, dwelling time reflects the loyalty of users to games, which should be of high vitality in revealing the users' preference. On the other hand, because of the design of games (e.g.,

¹<https://www.statista.com/statistics/292056/video-game-market-value-worldwide/>

²<https://store.steampowered.com/>

³<https://www.statista.com/statistics/552623/number-games-released-steam/>

RPG v.s. MOBA), the dwelling time of different games may not be comparable [34], which also leads to the difficulty.

Moreover, game contextualization is to comprehend the relatedness among games. Contextually similar games are recommended to users based on their historical game engagements. Existing methods [5, 6] construct the context of games by formatting that side information as features, e.g., the developer and price of a game will be processed as one-hot vector and float number features, respectively. This enables the recommender system to harness the side information of games rather than solely relying on the collaborative signals. Nevertheless, digesting context as features is unable to characterize complex and high-order relations between games. For example, if game A has the same developer as game B, and game B is usually co-purchased with game C according to users transactions, game A and game C should also be highly related when conducting recommendations. However, the complex co-purchase relation and the transitivity of relations are both neglected by feature-based methods. Hence, we should explicitly tackle the relations between games for contextualization.

Last but not least, social connections are rather crucial in game recommendations because many online video game engagements are not isolated individual behaviors but social activities. In this sense, discovering potential interests of users from their friends enables the recommender system to identify the game engagements of users from a social perspective. Nevertheless, the challenge is that due to social inconsistency [49], users' friends may be of different impacts concerning distinct games. For example, users usually play MOBA games *Dota 2*⁴ with friends in a team, whereas single-player RPG games *Portal*⁵ are played individually. As a result, directly recommending those games equally based on the engagements of friends [28] yields sub-optimal performance. Therefore, we should infer the impacts of friends on users.

To this end, we propose a novel game recommendation framework named SCGRec. SCGRec can be split into three parts, **time-aware context aggregation**, **context-aware social aggregation** and **personalized predictor**. User's contextual embedding is learned in time-aware context aggregation module. It aggregates information from user's engaged games with consideration on dwelling time. User's social embedding is obtained from context-aware social aggregation module. This module firstly learns attention weight of neighbors from user's contextual embedding, and obtains social embedding by weighted sum of neighbors' personalized embedding. In personalized predictor, contextual embedding and social embedding are fused with user's personalized embedding by weighted sum. The prediction score is obtained by a dot product of aggregated user embedding and personalized item embedding. The code and data for experiment is available online at <https://github.com/YangLiangwei/game-recommendation>.

The contributions of this paper are as follows:

- We conduct comprehensive data analyses of the game engagement based on a large-scale Steam dataset containing nearly billions of interactions. We investigate both the statistical characteristics and underlying patterns of the game engagements.

⁴https://store.steampowered.com/app/570/Dota_2/

⁵<https://store.steampowered.com/app/400/Portal/>

- We propose a unified framework to incorporate personalization, game contextualization and social connections, which improves the game recommendation performance compared with existing methods, especially those state-of-the-art graph neural networks.
- We conduct extensive ablation studies to demonstrate the effectiveness of leveraging all those three perspectives.

2 RELATED WORK

2.1 Game Recommendation

Designing game recommender systems is not a academic focus until recent years [1, 3, 6] due to the rapid developments of game industries [24]. We categorize existing works as: 1) the collaborative filtering (CF)-based methods, 2) the content-based methods, and 3) the hybrid methods.

Firstly, collaborative filtering (CF)-based methods [35, 56] harness the interactions between users and games and assume users with similar behaviors are of similar game preference. R. Sifa etc. [33, 34], are pioneering researchers solving game recommendation problems via archetypal, which factorizes the interaction matrix into the linear combination of archetypes. GAMBIT [1] investigates both the item-based and user-based CF methods for game recommendation. G. Cheque etc. [5] demonstrate the game recommendation performance on Steam with a simple CF-based method via the ALS algorithm [36]. CF-based methods are effective in reflecting the interactive signals between users and games.

Secondly, content-based methods [27] leverage the profile of users and the description of games and predict their interaction likelihood. B. Paul etc. [3] predict the next game that users will interact by fitting their time-series patterns. J. Kim etc. [15] conduct the ranking-based recommendation by sequentially considering user historical behaviors.

Finally, hybrid methods characterize both the CF signals and content-based information for the recommendation. G. Cheque etc. [5] investigate the performance of FM [29] and DeepFM [12] towards game recommendation by leveraging both the interactions and feature information of users and games. J. Pérez-Marcos etc. [28] model feature-based relations between games and then apply the CF for the recommendation. They also firstly study the possibility of using social networks over users to improve performance. Hybrid methods are of both flexibility and adaptability regarding the game recommendation task. Our proposed model is also a hybrid recommender system.

2.2 Graph-based Recommendation

We review some graph-based recommendation methods as we adopt the graph neural network (GNN) [38, 43, 46, 54] for the game recommendation. Graph-based recommendation methods model user-item interactions as a bipartite graph [2, 14, 23, 32, 51, 53], with potential extensions to the heterogeneous graph with additional user-user social graph [25, 49] and item knowledge graph [39, 40]. Graph-based methods mostly adopt GNN for learning nodes (users and items) embeddings. GNNs aggregate information from neighbors and can incorporate first-order and even higher-order information if multiple layers are stacked. Such unique characteristics allow GNNs to effectively capture high-order collaborative signals,

Table 1: Number of different nodes in Raw Data

users	games	genres	engagements	social connections
3.8M	4,079	22	384.3M	392.7M

which are crucial signals for learning user and item embeddings in recommendation [14, 41].

Heterogeneous GNNs [20, 40] are applied for recommendation for the combination of more side information such as knowledge graph and social network. Knowledge-aware GNNs recommendation [40] increasingly attracts attentions, intending to improve recommendation by connecting items with knowledge entities by various relationships. KGAT [40] introduces a unified collaborative knowledge graph from the user-item interaction graph and knowledge graph and proposes a GNN for learning both recommendation and knowledge tasks. KGNN-LS [39] introduces label smoothing regularization on knowledge graphs to further enhance the item embeddings learning.

In addition to knowledge graph, GNNs also enables the modelling of user-user social graph as side information in recommendation. GraphRec [7] learns graph attention network to assign different attention weights to different social neighbors. DiffNet [44] models information diffusion process [4] in social graph to enlarge user's influence scope [11]. DANSER [45] performs dual graph attention networks on social and user-item interaction networks separately and fuses the learned embedding by a policy-based fusion layer. ConsisRec [49] dynamically samples informative neighbors and performs aggregation considering different relation types. FeSoG [21] proposes a GNN-based social recommendation system under graph federated learning setting [13]. Our work considers the social information as the social context via attention mechanism.

Our work aims to enhance game recommendation with additional contextual information. Different from previous works, SCGRec is flexible enough to integrate both time-aware contextual information and social connection information. Instead of using explicit message passing as existing works, SCGRec models those side information as contextual embeddings. As such, we can improve the recommendation performance by leveraging both game engagement information and additional contextual information.

3 DATA ANALYSIS

In this section, we first present an overview of the data statistics, then dive deep into the data characteristics by visualizing data distributions from personalization, game contextualization, and social connections perspectives.

3.1 Overview of Data Statistics

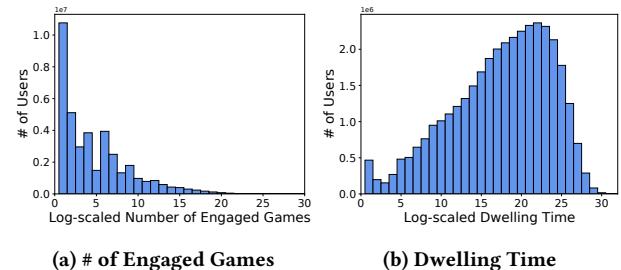
The raw data⁶ is crawled by Mark et al. [26] through Steam Web API⁷. Players and games form a highly unbalanced bipartite graph. Millions of players can play each game while each player only plays a small number of games, which makes the aggregated information highly unbalanced on the two sides. The following analysis is based on all the collected data.

⁶<https://steam.internet.byu.edu/>

⁷https://developer.valvesoftware.com/wiki/Steam_Web_API

3.2 Personalization

User engagement activities uncover their personalized interests in online video game platforms. We demonstrate the necessity and challenges of introducing personalization for game recommendation by investigating distributions of user engagements in both the numbers of engaging games and dwelling time. We visualize the user distributions with respect to the number of engaging games in Figure 1a and dwelling time in Figure 1b. From Figure 1a, we can observe the long-tail distribution on the number of engaging games. We also show the number of users with respect to different number of interacted games in Table 5. It shows a large number of users only interact with limited number of games. Note that, users' personalized information is not restricted to historical game interactions. Users' social friends and dwelling time of each game all count as personalized information. SCGRec achieves personalized recommendation by considering all such information. In addition to the number of engaging games, we plot the user count distribution over different dwelling times in Figure 1b. Dwelling time uncovers how much time users engage in games and is also a crucial signal of user interests. To illustrate this, we visualize the user count distribution on log-scaled dwelling time in Figure 1b. The dwelling time distribution almost follows the normal distribution, which is different from the long-tail distribution of the number of engaging games. It reveals different behavioral patterns on whether and how long to engage in a game, which indicates that predictive modelings on these two signals demand distinct mechanisms.

**Figure 1: User Game Engagement Distributions**

3.3 Game Contextualization

Game contextualization is to comprehend the relations between games, which can further interpret the recommendation [47]. We investigate the game contextualization from both the genre level and the game level. Genre-level contextualization is to analyze whether games from different genres are co-played by users. We select the five most popular genres. We calculate the probability that conditioned users engaged in one genre, how likely those users also play those other genres. The relations between genres are illustrated in Figure 2. We observe that the conditional probability from different genres are diverse, which indicates the correlations between genres are distinct. Moreover, we observe that users playing simulation games are reluctant to play other game types because of the low probability in other genres.

The second investigation is designed for game-level contextualization. Here, we present the co-purchase relation between games.

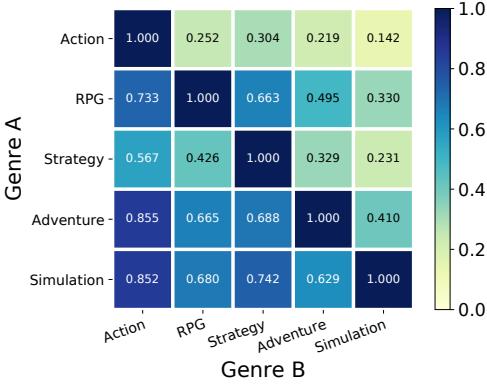


Figure 2: Genre Conditional Probability $P(A|B)$ Visualization

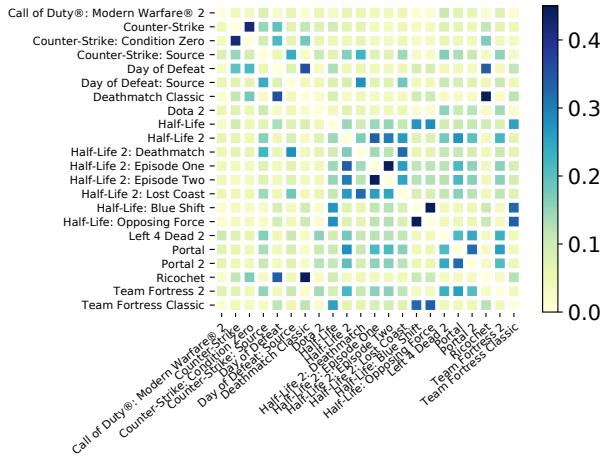


Figure 3: Game-Game Co-Purchase

Precisely, the co-purchasing relation score c_{ij} between game i and j is calculated as follows:

$$c_{ij} = \frac{\# \text{ of users engaged in both } i \text{ and } j}{\# \text{ of users engaged in } i + \# \text{ of users engaged in } j}. \quad (1)$$

High relation scores represent that two games are played by the larger group of shared users, which implies their high relatedness. We present the results for part of the games in Figure 3. We can observe that some game pairs have a high correlation, e.g. the Counter-Strike and Counter-Strike: Condition Zero. This will be a strong signal for game contextualization. We also present the co-dwelling relation scores between those games in the appendix.

3.4 Social Connections

Social activities also influence online video game engagements. To verify this, we conduct two types of data analyses over the game engagements of users and their friends. The first analysis is to demonstrate whether game engagements are social activities. To achieve this, we select the top-10 popular games and calculate the

Pearson correlation coefficient between the dwelling time of users and the average dwelling time of their friends. The results are in Table 2. For comparison, we randomly select the same number of users with no friendships with the associated users and calculate their Pearson correlation coefficient. Observations are twofold. Firstly, game engagements are influenced by social connections because the correlation scores between friends are higher than between non-friends on all the games. Secondly, the intensity of social impacts is different across games. For example, the MOBA game *Dota 2* has a higher correlation score compared with the single-player game *Portal*. We also include the Pearson correlation coefficient concerning top-5 genres, and distribution of friend number in the appendix.

Table 2: Pearson correlation coefficient w.r.t. games.

Game	user-friend	user-random
Counter-Strike	0.5816	0.1846
Dota 2	0.5422	0.1649
Team Fortress Classic	0.4705	0.0678
Team Fortress 2	0.4659	0.1489
Day of Defeat	0.4325	0.0609
Left 4 Dead 2	0.4304	0.0917
Ricochet	0.3084	0.0246
Deathmatch Classic	0.2376	0.0142
Half-Life	0.2264	0.0217
Portal	0.1216	0.0254

The second analysis is to investigate whether friends are of similar game preference. To calculate this similarity score, we first define the user game preference vector as the genre-wise engagement distribution $[t_1, t_2, \dots, t_k]$, where each entry t_k denotes the log-scaled dwelling time in genre k . Due to the power-law distribution of users in genres, we choose the top-5 genres. Then, the preference similarity is calculated as the cosine similarity between users, computed over all social connections. Furthermore, for comparison, we also randomly sample the same number of non-friendship links between users. The histogram of the similarity score is distributed in Figure 4. We can observe that in the range of [0.9, 1.0], the number of friendship connections is more than non-friendship connections, suggesting that friends are more likely to have a similar preference.

4 PRELIMINARIES

In this section, we present several basic definitions before formulating the problem of game recommendation.

DEFINITION 4.1. (Game Engagement). Given the game set \mathcal{I} , the game engagement of a user u is defined as a set of (game, time) pairs as $\mathcal{E}_u = \{(i_1, t_1), (i_2, t_2), \dots, (i_k, t_k)\}$, where $i_1, i_2, \dots, i_k \in \mathcal{I}$ and $t_1, t_2, \dots, t_k \geq 0$. k is the total number of engaged games of u and t_k denotes the dwelling time in game i_k .

The personalization requires the game engagements of all users \mathcal{U} , which forms a bipartite attributed graph. The nodes are users and games, and the edges are their interactions with dwelling time. Dwelling time is calculated as how long the user has played a game, reflecting the intensity of the game engagement.

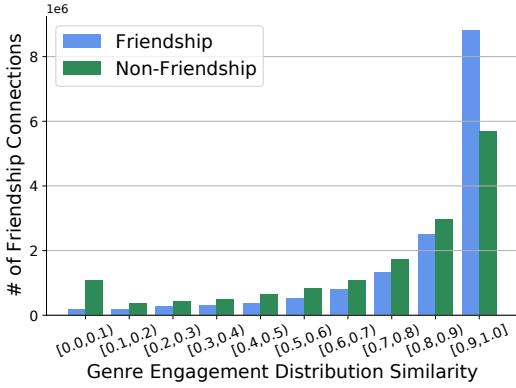


Figure 4: The comparison between counting distributions of friendship edges and non-friendship edges within different intervals of game genres engagement cosine similarity.

Besides the interactions between users and games, we also leverage the context information for games, e.g., the developer of games. Moreover, instead of directly processing context information, we form relations between games via shared context. In other words, if two games share the same context, we construct a link between them. In this way, we define the game context graph as follows:

DEFINITION 4.2. (Game Context Graph). Given the game set \mathcal{I} , the game context graph for context c is defined as $\mathcal{G}_i^{(c)} = \{\mathcal{I}, \mathcal{E}_c\}$, where \mathcal{I} is the node set and \mathcal{E}_c denotes edges. For two games $i, j \in \mathcal{I}$, $(i, j) \in \mathcal{E}_c$ if they share context c .

The definition of context is rather flexible. Context shows some relevance between games. It can be the game attributes, co-engaged relations and game mechanisms [22]. Each relevance can be represented as one kind of edge in the game context graph. Those contexts constitute a multiple relation game context graph as $\mathcal{G}_i = \{\mathcal{G}_i^{(c)}\}_{c=1}^C$, where C is the total number of contexts.

Moreover, users are connected by their friendship links, with which we can define the social graph of users:

DEFINITION 4.3. (Social Graph). Given users \mathcal{U} , the social graph is defined as $\mathcal{G}_u = \{\mathcal{U}, \mathcal{S}\}$, where \mathcal{S} denotes edges. For two users $u, v \in \mathcal{U}$, $(u, v) \in \mathcal{S}$ if they are friends.

Figure 5 shows the graph structure. Game context graph and social graph act like side information to game engagement graph.

Thereafter, we can formulate game recommendations as follows:

DEFINITION 4.4. (Game Recommendation). Supposing a set of users \mathcal{U} and games \mathcal{I} , given the game engagement of all users $\{\mathcal{E}_u\}_{u \in \mathcal{U}}$, the game context graph \mathcal{G}_i and the social graph \mathcal{G}_u , we should recommend a user $u \in \mathcal{U}$ a ranking list of games that u has no engagements in.

5 PROPOSED MODEL

In this section, we introduce SCGRec. It has four crucial components: game context graph neural network, time-aware context aggregation, context-aware social aggregation, and the personalized predictor. The proposed model is shown in Figure 6.

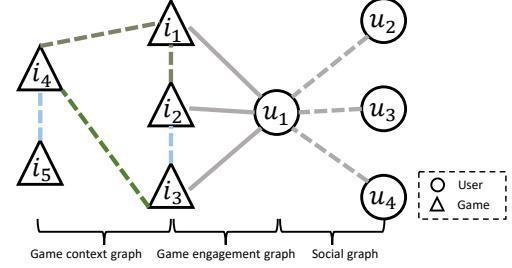


Figure 5: Graph structure of Steam data

5.1 Game context graph neural network

Game context graph \mathcal{G}_i is built to propagate information among similar games. In this dataset, we build \mathcal{G}_i by 5 game relations including 3 feature-based relations and 2 behavior-based relations.

Feature-based relations include co-genre, co-developer, and co-publisher based on different game features. For example, edges in co-developer relation indicate the same game developer develops the connected games. Behavior-based relations are constructed from users' behavior information. Co-purchase relation measures whether two games are likely to be purchased together. The co-purchase score between game i and j is calculated by Equation 1. Then the edges with a co-purchase score larger than threshold τ_p remain in the co-purchase graph. The co-dwelling relation between two games reflects whether two games can attract users to play a similar period length. The co-dwelling score is computed by:

$$\text{Co-dwelling} = e^{-\frac{|t_i - t_j|}{T}}, \quad (2)$$

where t_i and t_j are the average dwelling time for users co-purchased game i and game j , respectively. T is a constant to normalize the time. Then the edges with co-dwelling scores larger than threshold τ_t are kept in the co-dwelling graph.

As shown in Figure 7, one graph convolution is performed for each relation type. Then the game contextual embedding $\mathbf{e}_i^c \in \mathbb{R}^d$ is obtained after a pooling layer to aggregate information from all relation types. Specifically, we use mean aggregation as:

$$\mathbf{e}_i^c = \frac{1}{|C|} \sum_{c \in C} \sum_{j \in \mathcal{N}^c(i)} \frac{1}{m_{i,j}^c} \mathbf{W}_c \mathbf{h}_j + \mathbf{b}_c, \quad (3)$$

where $\mathcal{N}^c(i)$ is the neighbor set of node i w.r.t. relation c . \mathbf{h}_j denotes the hidden feature vector of node j . $m_{i,j}^c = \sqrt{|\mathcal{N}^c(i)|} / \sqrt{|\mathcal{N}^c(i)|}$ is the normalization factor. \mathbf{W}_c and \mathbf{b}_c are graph convolution parameters.

5.2 Time-aware context aggregation

Dwelling time is a critical information for game recommendation, which signifies user engagements. However, time information can not be directly used because the dwelling time of different games are incomparable. Instead, we calculate the user dwelling time percentile to measure user's engagement in each game.

User's contextual embedding $\mathbf{e}_u^c \in \mathbb{R}^d$ is obtained from game contextual embedding \mathbf{e}_i^c with the consideration of user engagement

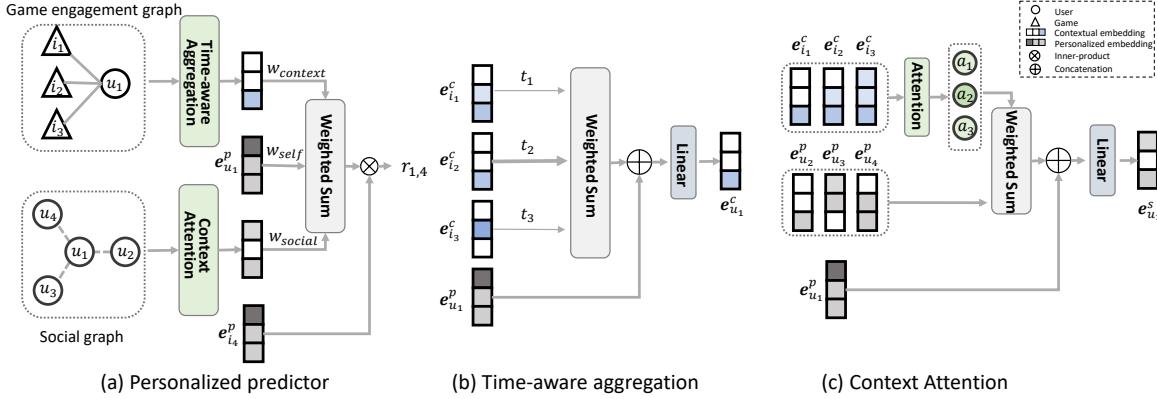


Figure 6: Framework of SCGRec

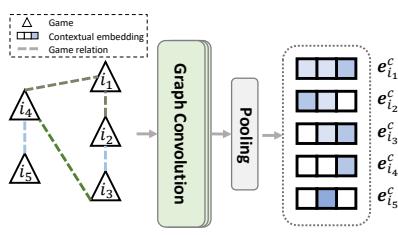


Figure 7: Game context graph aggregation

\mathcal{E}_u . For u , the time-aware context aggregation is calculated by:

$$\mathbf{e}_{\text{agg}}^c = \sum_{(i,t) \in \mathcal{E}_u} \gamma_{i,t} \mathbf{e}_i^c, \quad (4)$$

where $\gamma_{i,t}$ is the time-aware weight, which is calculated as follows:

$$\gamma_{i,t} = \frac{\text{percentile}(i, t)}{\sum_{(i',t') \in \mathcal{E}_u} \text{percentile}(i', t')}, \quad (5)$$

where $\text{percentile}(i, t)$ returns time t 's percentile in all dwelling time occurred in game i .

Then the contextual embedding of user u is calculated by:

$$\mathbf{e}_u^c = \mathbf{W}(\mathbf{e}_u^p \oplus \mathbf{e}_{\text{agg}}^c), \quad (6)$$

where $\mathbf{W} \in \mathbb{R}^{d \times 2d}$ is a linear projection matrix, \oplus is the concatenation operator and \mathbf{e}_u^p is u 's personalized embedding.

5.3 Context-aware social aggregation

In Table 2, we observe higher correlation between friends than random users, which implies that social neighbors are beneficial for inferring users' interests. Since contextual embeddings of users contain the game engagement information, the aggregation of context embedding from social connection perspectives can help explore more similar neighbors. We use context attention to obtain the attention weight for social aggregation:

$$\alpha_{u_i} = \text{softmax}_{u_j \in \mathcal{N}(u)}(e_{u_i}) = \frac{\exp(e_{u_i})}{\sum_{u_j \in \mathcal{N}(u)} \exp(e_{u_j})}, \quad (7)$$

$$\mathbf{e}_{u_i} = \text{LeakyReLU} \left(\mathbf{a}^\top (\mathbf{W} \mathbf{e}_u^c \oplus \mathbf{W} \mathbf{e}_{u_i}^c) \right), \quad (8)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathbf{a} \in \mathbb{R}^{2d}$ are parameters for calculating context attention weights. $\mathcal{N}(u)$ is the neighbor set of node u , \mathbf{e}_u^c is node u 's contextual embedding, and \oplus is the concatenation operator. We perform weighted aggregation on personalized embedding as:

$$\mathbf{e}_{\text{agg}}^p = \sum_{u_i \in \mathcal{N}(u)} \alpha_{u_i} \mathbf{e}_{u_i}^p, \quad (9)$$

then the social embedding of user u is calculated as:

$$\mathbf{e}_u^s = \mathbf{W}(\mathbf{e}_u^p \oplus \mathbf{e}_{\text{agg}}^p), \quad (10)$$

where $\mathbf{W} \in \mathbb{R}^{d \times 2d}$ is a linear transformation, \oplus is the concatenation operator and \mathbf{e}_u^p is u 's personalized embedding.

5.4 Personalized Predictor

For user u , we have context embedding \mathbf{e}_u^c from time-aware context aggregation, social embedding \mathbf{e}_u^s from context-aware social aggregation and personalized embedding \mathbf{e}_u^p . To simultaneously incorporating all these information, we assign different weights and sum them to obtain the final embedding of users as follows:

$$\mathbf{e}_u = w_{\text{context}} \mathbf{e}_u^c + w_{\text{social}} \mathbf{e}_u^s + w_{\text{self}} \mathbf{e}_u^p, \quad (11)$$

where w_{context} , w_{social} and w_{self} are scalar hyper-parameters to decide the importance of the corresponding information. Hereafter, the final rating score is calculated by the dot-product between user embedding \mathbf{e}_u and game personalized embedding \mathbf{e}_i^p as follows:

$$r_{u,i} = \mathbf{e}_u \cdot \mathbf{e}_i^p. \quad (12)$$

We adopt the Bayesian Personalized Ranking (BPR) loss [30] for optimizing all trainable parameters, which includes the embeddings and convolution weights. Loss function is defined as:

$$\mathcal{L} = - \sum_{(u,i) \in \mathcal{E}_u, j \in \mathcal{I} \setminus \mathcal{E}_u} \log \sigma(r_{u,i} - r_{u,j}) + \lambda \|\Theta\|_2^2, \quad (13)$$

where j denotes the negative games sampled from $\mathcal{I} \setminus \mathcal{E}_u$ that user u has no interactions with, λ is the hyper-parameter, and Θ includes all trainable parameters.

6 EXPERIMENTS

6.1 Experimental Setup

6.1.1 Steam Data Set. Due to the limitation of computational resources, we preprocess the raw data by two steps to make it feasible for training and evaluation. Firstly, we filter out the users with less than 5 game interactions or play games for less than 60 minutes. Then 30% of the users are randomly sampled to form the data set. The details of the filtered dataset are shown in Table 3. Given this dataset, We then build a validation set and a test set by randomly sampled 50,000 players. The validation set and test set are formed by 10% game interactions of these selected players. This processed data is publicly released⁸ for future reference.

Table 3: Statistics of filtered Steam Data

# Players	3,908,744
# Games	2,707
# Publishers	689
# Developers	1,170
# Interactions	95,441,434
# Social Connections	10,625,806

6.1.2 Baselines. To evaluate the intrinsic characteristics of game recommendation, we compare our proposed model with several baseline models, including Popularity (time), Popularity (count), LightGCN [14], RGCN [31], GIN [48], PinSAGE [51] and GAT [37]. Detailed description and comparison are given in the appendix.

6.1.3 Implementation Details. We implement SCGRec in Pytorch and deep graph library (DGL)⁹, an open-source framework for graph neural networks. Hyper-parameters include embedding size, learning rate, batch size, social embedding aggregation weight W_{social} and contextual embedding aggregation weight $W_{context}$. The personalized embedding aggregation weight is set to $1 - W_{social} - W_{context}$. The tuning range and best hyper-parameters setting is given in the appendix. Adam [16] is adopted as the optimizer. BPR loss is adopted for optimization with random negative sample. Early stop strategy is adopted to avoid over-fitting. The training is stopped if the model's performance does not increase in successive 10 epochs. During testing, we rank all the potential games from those 2,702 games for each user based on their interaction scores. Then we calculate the metrics including NDCG@K, Recall@K, Hit Ratio@K, and Precision@K, where K ranges in {5, 10, 20}

6.2 Performance Evaluation

Experiment results are shown in Table 4. from the results. We have the following observations:

- SCGRec significantly outperforms the state-of-the-art GNN models. In the top 5 recommendation, SCGRec outperforms the second-best model by more than 5% on NDCG, Recall, and Precision. By tackling the personalization, social connection, and game interpretation information accordingly, SCGRec can effectively fuse

⁸https://drive.google.com/file/d/1F9kr_YWimBtexJEH-zkDzC0wl1q7GmFp/view?usp=sharing

⁹<https://www.dgl.ai/>

all side information to enhance game recommendation. When the recommendation list gradually becomes longer, the improvement of SCGRec becomes more and more limited on all the metrics. For example, the improvement on Recall@5 is 5.96% while on Recall@20 is only 2.66%. It shows the difficulty of improving on a long recommendation list.

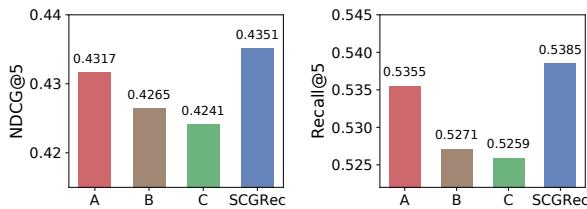
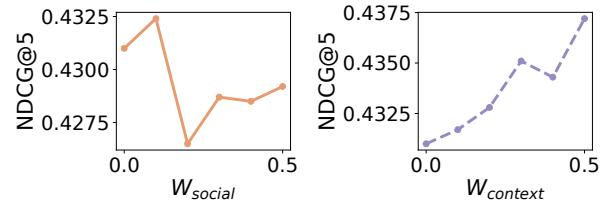
- The personalized recommendation is much better than the non-personalized recommendation. Popularity (time) and Popularity (count) are non-personalized recommendation methods. They generate the same recommendation list for all users based on game popularity. The worse performance of popularity-based methods shows the importance of personalized recommendations. Popularity (count) performs nearly twice as well as popularity (time). It shows that directly utilizing dwelling time length is not informative. The same dwelling time reflects different engagements on different games. Thus, dwelling time information is modeled as user engagement percentile in SCGRec.
- Nodes' self-information is important in GNN aggregation. LightGCN and RGCN have much lower performance than other GNN based models. Compared to other GNN models, the main difference between LightGCN and RGCN is that they directly aggregate all the neighbor information without considering the relative importance of self-information. PinSAGE firstly aggregates neighbor embedding and self-embedding, then it fuses the information by a learnable linear layer. GIN also focuses more on self-embedding. It adds the aggregated neighbor embedding as side information. GAT learns attention weight for each neighbor with consideration on nodes' self-embedding. In SCGRec, we explicitly add nodes' self-information controlled by hyper-parameter w_{self} .
- Side information such as social links and game context graphs are helpful to improve recommendation performance. Compared with LightGCN, RGCN builds social link and game context information as different relation types and aggregates information from all relations. LightGCN only aggregates on a user-game bipartite graph. The improvement of RGCN over LightGCN can be explained by utilizing more side information. SCGRec explicitly models game-side information as contextual embedding by time-aware aggregation. Social link information is also modeled as social embedding by context-aware social aggregation module.

6.3 Ablation Study

In this section, we perform an ablation study on SCGRec. We build three variant of SCGRec by removing part of its modules: 1) variant A is built by removing social embedding; 2) variant B removes those contextual embeddings; and 3) variant C is constructed by removing both social and contextual information. Experimental results are demonstrated in Figure 8. Compared with all three variants, SCGRec performs the best on all metrics. We observe that considering both social embedding and contextual embedding achieves the best performance. Moreover, variant A performs better than variant B, which indicates that removing contextual embedding has more negative effects on SCGRec. Hence, the context information is relatively more important than social information. Compared with variant C, A and B both have better performance. This justifies the effectiveness of both the social connections and game contextualization in game recommendation.

Table 4: Overall comparison, the best and second-best results are in bold and underlined, respectively

Method	NDCG			Recall			Hit Ratio			Precision		
	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
Popularity (time)	0.0896	0.1053	0.1256	0.1106	0.1584	0.2324	0.1600	0.2320	0.3290	0.0336	0.0255	0.0192
Popularity (count)	0.1654	0.2050	0.2380	0.2335	0.3527	0.4735	0.2960	0.4348	0.5745	0.0662	0.0512	0.0357
LightGCN	0.1861	0.2100	0.2475	0.2452	0.3174	0.4507	0.2849	0.3784	0.5504	0.0637	0.0447	0.0347
RGCN	0.2253	0.2616	0.2980	0.2888	0.4027	0.5359	0.3771	0.5107	0.6550	0.0863	0.0629	0.0441
GIN	0.3824	0.4179	0.4460	0.4851	0.5924	0.6872	0.5978	0.7163	0.8060	0.1375	0.0912	0.0582
PinSAGE	0.3963	0.4271	0.4529	0.4927	0.5863	0.6740	0.6025	0.7088	0.7926	0.1385	0.0899	0.0567
GAT	0.4053	0.4385	0.4655	0.5081	0.6082	0.6988	0.6209	0.7319	0.8158	0.1430	0.0936	0.0591
SCGRec	0.4351	0.4660	0.4921	0.5385	0.6311	0.7175	0.6519	0.7535	0.8331	0.1508	0.0969	0.0609
Improvement	6.98%	6.25%	5.70%	5.96%	3.75%	2.66%	4.98%	2.94%	2.11%	5.43%	3.52%	2.93%

**Figure 8: Ablation study of SCGRec****Figure 9: Social and Context influence**

It is also worth noting that SCGRec performs better than the best baseline even without the side information. The reason is that the number of nodes in the game engagements is rather imbalanced, millions of players v.s. thousands of games. Hence, the direct neighbor aggregation of existing GNN methods on the player-game interaction graph leads to the **over-smoothing problem**. However, SCGRec adopts personalized embeddings, which are optimized directly from interaction rather than aggregation from neighbors.

6.4 Social and Context Influence

In this section, we show the influence of social and context information by tuning the importance weight w_{social} and $w_{context}$. w_{self} is calculated by $w_{self} = 1 - w_{social} - w_{context}$. Experiment results are shown in Figure 9. When w_{social} gradually becomes larger, SCGRec's performance first increases and then decreases. It shows social information contains lots of noise. Similar observations are also shown in previous research. Junliang et al. [52] discuss that explicit social links are not all reliable due to the existence of spammers and bots. Also, the inconsistent social neighbors [49] can also be seen as noise. When we pay much importance weight to social embedding, the noise in social neighbors is also amplified, which may lower down the performance. It can also be observed from Figure 9 that when $w_{context}$ gradually becomes larger, SCGRec performs better. It shows the learned information from game context graph is beneficial. Providing the flexibility in constructing game context graph, SCGRec can easily model multi-type game context information to boost model's performance.

7 CONCLUSION AND FUTURE WORK

In this paper, we comprehensively analyze nearly billion of user behaviors on the Steam online gaming platform, including dwelling time, social impacts, and game context information. Based on our analysis, we argue that 3 key characteristics should be harnessed in the game recommendation, which are personalization, game contextualization and social connection. Therefore, we propose a new model SCGRec by tackling those characteristics simultaneously for the game recommendation. Experimental results demonstrate the superiority of SCGRec regarding game recommendation performance. It has better performance against all existing GNN based methods in game recommendation. The SCGRec model is simple and rather flexible to incorporate various side information for recommendation, which provides the game industry with an effective model to increase the game engagement of users.

We leave several research directions in game recommendations as to future work. Firstly, dwelling time measures user direct game engagement. But, how to effectively utilize dwelling time information is still an open question. Secondly, the user game engagement graph is a highly imbalanced bipartite graph. Differences in neighbor aggregation behavior on the two sides can be further investigated. Lastly, as shown in data analyses, there exist a large portion of cold-start players. How to effectively recommend games for those players is yet another challenging research problem.

8 ACKNOWLEDGMENTS

This work was supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

REFERENCES

- [1] Syed Muhammad Anwar, Talha Shahzad, Zunaira Sattar, Rahma Khan, and Muhammad Majid. 2017. A game recommender system using collaborative filtering (GAMBIT). In *2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE, 328–332.
- [2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
- [3] Paul Bertens, Anna Guitart, Pei Pei Chen, and África Periéñez. 2018. A machine-learning item recommendation system for video games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–4.
- [4] Lingjiao Chen and Jian Gao. 2018. A trust-based recommendation method using network diffusion processes. *CoRR abs/1803.08378* (2018). arXiv:1803.08378 <http://arxiv.org/abs/1803.08378>
- [5] Germán Cheque, José Guzmán, and Denis Parra. 2019. Recommender systems for Online video game platforms: The case of STEAM. In *Companion Proceedings of The 2019 World Wide Web Conference*. 763–771.
- [6] Lorenzo De Simone, Davide Gadia, Dario Maggiolini, and Laura Anna Ripamonti. 2021. Design of a Recommender System for Video Games based on In-Game Player Profiling and Activities. In *CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter*. 1–8.
- [7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13–17, 2019*. ACM, 417–426.
- [8] Ziwei Fan, Zhiwei Liu, Shen Wang, Lei Zheng, and Philip S. Yu. 2021. Modeling Sequences as Distributions with Uncertainty for Sequential Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 3019–3023. <https://doi.org/10.1145/3459637.3482145>
- [9] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S. Yu. 2022. Sequential Recommendation via Stochastic Self-Attention. *arXiv preprint arXiv:2201.06035* (2022).
- [10] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S. Yu. 2021. Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 433–442. <https://doi.org/10.1145/3459637.3482242>
- [11] Chungu Guo, Liangwei Yang, Xiao Chen, Duanbing Chen, Hui Gao, and Jing Ma. 2020. Influential Nodes Identification in Complex Networks via Information Entropy. *Entropy* 22, 2 (2020), 242. <https://doi.org/10.3390/e22020242>
- [12] Huifeng Guo, Ruiming Tang, Yuning Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017*, Carles Sierra (Ed.). ijcai.org, 1725–1731. <https://doi.org/10.24963/ijcai.2017/239>
- [13] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S. Yu, Yu Rong, et al. 2021. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145* (2021).
- [14] 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.
- [15] JaeWon Kim, Jeonga Wi, Soojin Jang, and YoungBin Kim. 2020. Sequential Recommendations on Board-Game Platforms. *Symmetry* 12, 2 (2020), 210.
- [16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- [17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=SLU4ayYg1>
- [18] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S. Yu. 2021. *Augmenting Sequential Recommendation with Pseudo-Prior Items via Reversely Pre-Training Transformer*. Association for Computing Machinery, New York, NY, USA, 1608–1612. <https://doi.org/10.1145/3404835.3463036>
- [19] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and S Yu Philip. 2020. Basket recommendation with multi-intent translation graph neural network. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 728–737.
- [20] Zhiwei Liu, Mengting Wan, Stephen Guo, Kannan Achan, and Philip S. Yu. 2020. Basconv: Aggregating heterogeneous interactions for basket recommendation with graph convolutional neural network. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 64–72.
- [21] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S. Yu. 2021. Federated Social Recommendation with Graph Neural Network. *arXiv preprint arXiv:2111.10778* (2021).
- [22] Tiago Machado, Dan Gopstein, Andy Nealen, and Julian Togelius. 2019. Pitako: recommending game design elements in cicero. In *2019 IEEE Conference on Games (CoG)*. IEEE, 1–8.
- [23] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.
- [24] André Marchand and Thorsten Hennig-Thurau. 2013. Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities. *Journal of interactive marketing* 27, 3 (2013), 141–157.
- [25] Dennis Nii Ayeh Mensah, Gao Hui, and Liangwei Yang. 2020. Approximation Algorithm for Shortest Path in Large Social Networks. *Algorithms* 13, 2 (2020), 36. <https://doi.org/10.3390/a13020036>
- [26] Mark O'Neill, Elham Vaziripour, Justin Wu, and Daniel Zappala. [n. d.]. Condensing Steam: Distilling the Diversity of Gamer Behavior. In *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14–16, 2016*, Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan (Eds.), 81–95.
- [27] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.
- [28] Javier Pérez-Marcos, Lucía Martín-Gómez, Diego M Jiménez-Bravo, Vivian F López, and María N Moreno-García. 2020. Hybrid system for video game recommendation based on implicit ratings and social networks. *Journal of Ambient Intelligence and Humanized Computing* 11, 11 (2020), 4525–4535.
- [29] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 1–22.
- [30] Steffen Rendle, Christoph Freudenthaler, Zeno Ganter, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18–21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25
- [31] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.
- [32] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. 2021. How Powerful is Graph Convolution for Recommendation? In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1619–1629.
- [33] Rafet Sifa, Christian Bauckhage, and Anders Drachen. 2014. Archetypal Game Recommender Systems. In *LWA*. 45–56.
- [34] Rafet Sifa, Christian Bauckhage, and Anders Drachen. 2014. The Playtime Principle: Large-scale cross-games interest modeling. In *2014 IEEE conference on computational intelligence and games*. IEEE, 1–8.
- [35] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* 2009 (2009), 421425:1–421425:19. <https://doi.org/10.1155/2009/421425>
- [36] Gábor Takács and Domonkos Tikk. 2012. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*. 83–90.
- [37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [38] Chen Wang, Yingtong Dou, Min Chen, Jia Chen, Zhiwei Liu, and Philip S. Yu. 2021. Deep Fraud Detection on Non-attributed Graph. In *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15–18, 2021*. IEEE, 5470–5473. <https://doi.org/10.1109/BigData52589.2021.9672028>
- [39] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 968–977.
- [40] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.
- [41] 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.
- [42] Yu Wang, Zhiwei Liu, Ziwei Fan, Lichao Sun, and Philip S. Yu. 2021. DSKReG: Differentiable Sampling on Knowledge Graph for Recommendation with Relational GNN. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3513–3517.
- [43] Yu Wang, Yuesong Shen, and Daniel Cremers. 2021. Explicit pairwise factorized graph neural network for semi-supervised node classification. In *Uncertainty in*

- Artificial Intelligence*. PMLR, 1979–1987.
- [44] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21–25, 2019*. ACM, 235–244.
- [45] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. 2091–2102.
- [46] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24. <https://doi.org/10.1109/TNNLS.2020.2978386>
- [47] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th international conference on web search and data mining*. 672–680.
- [48] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. OpenReview.net. <https://openreview.net/forum?id=ryGs6iA5Km>
- [49] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S. Yu. 2021. ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11–15, 2021*. Fernando Diaz, Chirag Shah, Torsten Stuel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2141–2145. <https://doi.org/10.1145/3404835.3463028>
- [50] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*. 113–120.
- [51] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.
- [52] Junliang Yu, Min Gao, Jundong Li, Hongzhi Yin, and Huan Liu. 2018. Adaptive Implicit Friends Identification over Heterogeneous Network for Social Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22–26, 2018*. Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selcuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 357–366.
- [53] Hengrui Zhang and Julian J. McAuley. 2020. Stacked Mixed-Order Graph Convolutional Networks for Collaborative Filtering. In *Proceedings of the 2020 SIAM International Conference on Data Mining, SDM 2020, Cincinnati, Ohio, USA, May 7–9, 2020*. Carlotta Domeniconi and Nitesh V. Chawla (Eds.). SIAM, 73–81. <https://doi.org/10.1137/1.9781611976236.9>
- [54] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S. Yu. 2021. From Canonical Correlation Analysis to Self-supervised Graph Neural Networks. *CoRR* abs/2106.12484 (2021). arXiv:2106.12484 <https://arxiv.org/abs/2106.12484>
- [55] Lei Zheng, Ziwei Fan, Chun-Ta Lu, Jiawei Zhang, and Philip S. Yu. 2019. Gated Spectral Units: Modeling Co-evolving Patterns for Sequential Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1077–1080.
- [56] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi, Evren Korpeoglu, Kannan Achan, and Jingrui He. 2021. PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2409–2419.

A DATA ANALYSIS

In this section, we present more details regarding the data analyses.

Table 5: Number of players w.r.t. different number of game interactions

Interacted game number	Player number
1	10,764,869
2	5,113,208
3	2,960,352
4	3,847,987
5	1,483,215
6	2,773,800
7	1,165,742
8	1,743,871
9	749,208
≥ 10	8,170,542

Table 5 shows the number of players with respect to different number of game interactions. As discussed in Section 3.2, a large portion of players only interact with limited number of games. The data sparsity causes challenges when recommending for these cold-start players. In SCGRec, we aim to mitigate this problem by players’ social graph and games’ context graph. Players’ social graph can link players to more games through social friends, and games’ context graph can learn the relevance information among games.

Table 6: Number of players w.r.t. different number of friends

Number of Social Friends	Number of players
1	9,025,326
2	4,286,860
3	2,769,634
4	2,012,605
5	1,564,368
6	1,269,972
7	1,056,377
8	895,596
9	770,913
≥ 10	9,468,130

As shown in Figure 10, players number of friends follows power-law distribution. Table 6 shows the exact number of players with respect to different number of friends. It can be observed that a large portion of players have limited number of friends. As side information, social graph can link more games through social neighbor, which is more important for cold-start players.

Table 7 shows pearson correlation coefficient between user-friend and user-random. Similar to Table 2 in Section 3.4, we can have two observations. Firstly, the pearson correlation coefficient between player and their friends is much higher than random players. It shows friends tend to have preference on the same game genre. Secondly, the social influence on different genres are different. For example, the pearson correlation coefficient of action genre

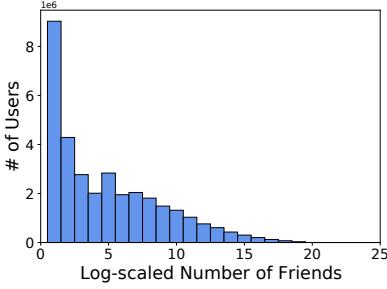


Figure 10: Players' number of friends distribution

Table 7: Pearson correlation coefficient w.r.t. genres

Genre	player-friend	player-random
Action	0.4597	0.1782
Adventure	0.1449	0.0546
RPG	0.2201	0.0661
Simulation	0.3722	0.0828
Strategy	0.3552	0.1070

is much larger than adventure genre. It shows social influence has a different level of impact on different genres.

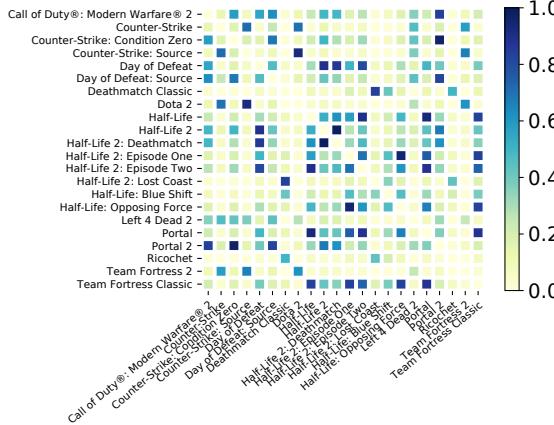


Figure 11: Game-Game Co-Dwelling

Figure 11 shows co-dwelling relation among part of the games. Similar to co-purchase relation shown in Figure 3, figure 11 also shows a high correlation between game pairs with respect to co-dwelling time. For example, co-dwelling time of game "Half-Life" and "Half-Life 2: Episode Two" is very long. It shows an apparent signal of game co-dwelling context.

B EXPERIMENTAL SETTING

Besides the data analyses, we also include more experimental settings.

Table 8: Best Hyper-parameter Setting

Hyper-parameter	Search range	Best setting
Embedding size	{4,8,16,32,64}	32
Learning rate	{0.03,0.01,0.001}	0.03
Batch size	{128,256,512,1024}	1024
λ	{1e-3, 1e-4, 1e-5}	1e-4
W_{social}	{0.0,0.1,0.2,0.3,0.4,0.5}	0.1
$W_{context}$	{0.0,0.1,0.2,0.3,0.4,0.5}	0.5

Table 9: Model comparison

Model	Social	Context	Personalization
Popularity (time)	✗	✗	✗
Popularity (count)	✗	✗	✗
LightGCN	✗	✗	✓
GIN	✗	✗	✓
PinSAGE	✗	✗	✓
GAT	✗	✗	✓
RGCN	✓	✓	✓
SCGRec	✓	✓	✓

The search range and best setting of hyper-parameters are given in Table 8. W_{self} is calculated by $W_{self} = 1 - W_{social} - W_{context}$. The same search range is applied to all the baselines.

Table 9 presents comparisons among all methods. Popularity (time) and Popularity (count) are non-personalized recommendation models. They directly recommend games based on popularity. LightGCN, GIN, PinSAGE and GAT utilize only user-game interaction data, and learn the node embedding by different GNN models. RGCN and our proposed SCGRec incorporate all the information including user-game interaction, social graph and game contextualization. A brief description of all the baselines are given as follows:

- Popularity (time): Same as Popularity (count), except that we measure the game popularity based on total dwelling time.
- Popularity (count): Directly rank all the games based on the number of players. Each player is recommended with the same game list with no personalization.
- LightGCN [14]: LightGCN simplifies Graph Convolution Network (GCN) [17] by removing transformation matrix, non-linear transformation, and self-loop, which is the SOTA GNN recommendation model.
- RGCN [31]: Relational Graph Convolution Network learns node embedding from all the side information. Each relation type is assigned one GCN layer for relation-specific aggregation.
- GIN [48]: Graph Isomorphism Network is a simple graph neural network that expects to achieve the ability as the Weisfeiler-Lehman graph isomorphism test.
- PinSAGE [51]: PinSAGE combines efficient random walks in graph convolution to learn node embedding on large-scale recommendation bipartite graphs.
- GAT [37]: Graph Attention Network firstly learns the attention scores for neighbors before its neighboring aggregation.