GMTL: A GART Based Multi-task Learning Model for Multi-Social-Temporal Prediction in Online Games

Jianrong Tao[†], Linxia Gong[†], Changjie Fan[†], Longbiao Chen[‡], Dezhi Ye[†], Sha Zhao[*]

[†]Fuxi AI Lab, NetEase Inc., Hangzhou, China
[‡]School of Information Science and Technology, Xiamen University, China
*College of Computer Science and Technology, Zhejiang University, China
{hztaojianrong, gonglinxia, fanchangjie, yedezhi}@corp.netease.com,
longbiaochen@xmu.edu.cn, szhao@zju.edu.cn

ABSTRACT

Multi-social-temporal (MST) data, which represent multi-attributed time series corresponding to the entities in multi-relational social network series, are ubiquitous in real-world and virtual-world dynamic systems, such as online games. Predictions over MST data such as social time series prediction and temporal link weight prediction are of great importance but challenging. They are affected by many complex factors, including temporal characteristics, social characteristics, collaborative characteristics, task characteristics and the intrinsic causality between them. In this paper, we propose a graph attention recurrent network (GART) based multi-task learning model (GMTL) to fuse information across multiple social-temporal prediction tasks. Experiments on an MMORPG dataset demonstrate that GMTL outperforms the state-of-the-art baselines and can significantly improve performances of specific social-temporal prediction task with additional information from others. Our work has been deployed to several MMORPGs in practice and can also expand to many related multi-social-temporal prediction tasks in real-world applications. Case studies on applications for multi-social-temporal prediction show that GMTL produces great value in the actual business in NetEase Games.

CCS CONCEPTS

• Information systems → Data mining; Massively multiplayer online games; • Computing methodologies → Multi-task learning; Neural networks; • Mathematics of computing → Graph algorithms; Time series analysis; Dynamic Graph algorithms.

KEYWORDS

Time series prediction; link weight prediction; multi-task learning; graph attention network; recurrent neural network; online game

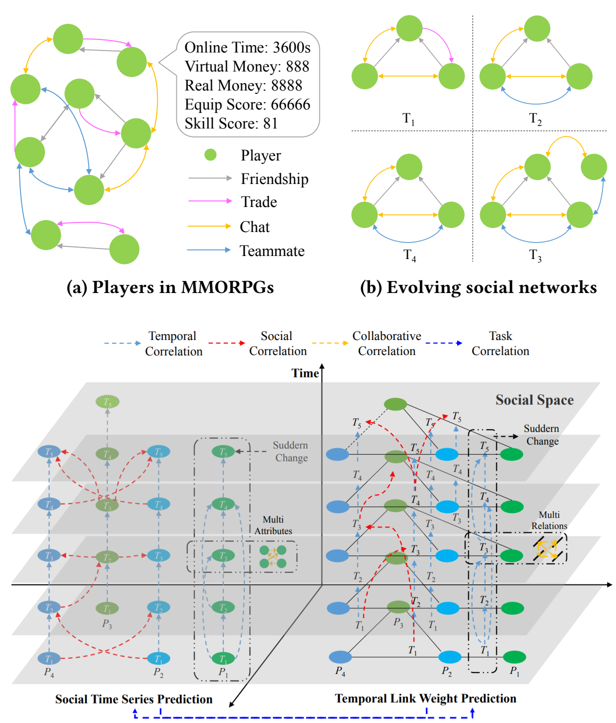
[†]NetEase Fuxi AI Lab: named after Fu Xi, the legendary Creator in China, and established to enlighten games with artificial intelligence. (https://fuxi.163.com/en/)

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11...$15.00
https://doi.org/10.1145/3357384.3357830

ACM Reference Format:
Jianrong Tao[†], Linxia Gong[†], Changjie Fan[†], Longbiao Chen[‡], Dezhi Ye[†], Sha Zhao[*]. 2019. GMTL: A GART Based Multi-task Learning Model for Multi-Social-Temporal Prediction in Online Games. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3357384.3357830

1 INTRODUCTION



(a) Players in MMORPGs **(b) Evolving social networks**

(c) Multi-social-temporal prediction

Figure 1: Illustration of correlations in multi-social-temporal prediction in online games.

Over the last decade, online games have become very popular, and one of the popular genres is **m**assively **m**ulti-player **o**nline **r**ole **p**laying **g**ames (MMORPGs). As shown in Figure 1a, an MMORPG provides a virtual world where players can participate in not only game-specific activities but also various social interactions such as making friends, trading assets, delivering messages or forming teams. As the entities of the specific social networks in the virtual world, millions of players share kinds of player attributes, like online time, (virtual or real) money, game scores etc. Many real social networks are dynamic in nature, where nodes, edges, and their attributes change over time. For example, as shown in Figure 1b, players may establish new social relations, remove old relationships, and new players may join the social network while old players can leave and their attributes like game scores can change over time. These multi-attributed time series corresponding to entities in multi-relational social network series, represented as **m**ulti-**s**ocial-**t**emporal (MST) data, are recorded on logs in the game servers.

MST data analysis in online games has become increasingly important in recent years. Social time series prediction and temporal link weight prediction are the two major tasks. Social time series prediction is to predict the future status of player conditions based on a fundamental variable of player attributes flow. Prediction over different types of player attributes helps game operations a lot. Online time prediction helps to predict churning (or returning) players, virtual money prediction prospects economic damage, real money prediction contributes to recharging (or consumption) forecasting and game scores prediction benefits player pursuits mining. Temporal link weight prediction is to predict the future network structure based on a sequence of observed networks. It's essential to recommend friends you may know or be interested in but not yet connected with, or to provide team formation suggestions for better gaming experience. It can also benefit to automatic friends list cleanup and produce players with a fast entry for delivering messages to close friends.

Despite the great efforts that have been made, multi-social-temporal prediction is still challenging, affected by the following factors which are shown in Figure 1c:

- ***Temporal correlation*** Attributes and relationships of players are affected by historical values of themselves both near and far and usually follow periodic and trending patterns. Players' attributes and relationships sometimes fluctuate tremendously and sharply, quickly decreasing the impact of observations in their previous time intervals.
- ***Social correlation*** Attributes and relationships of a player are affected by the nearby players as well as the distant players in the social space. The social correlation between different players' attributes and relationships is highly dynamic, changing over time. When modeling dynamic pairwise correlations, classical methods have extremely heavy computational cost due to their massive parameters.
- ***Collaborative correlation.*** Different types of attributes and relationships affect each other and contribute to single attribute or single relationship prediction as well.
- ***Task correlation.*** Social time series prediction and temporal link weight prediction may contribute to each other.

To tackle the aforementioned challenges, we propose a graph attention recurrent network based multi-task learning model for multi-social-temporal prediction in online games. The contributions of our study are five-fold:

(1) ***Multi-social-temporal prediction.*** To the best of our knowledge, this is the first work that introduces the multi-social temporal prediction problem into online games considering the aforementioned correlations.
(2) ***The GART.*** We propose a graph attention recurrent network that combines the multi-graph attention network and the recurrent neural network (long-short term memory network) to square up the social and temporal correlation.
(3) ***The GMTL.*** We propose a graph attention recurrent network based multi-task learning model to premeditate the collaborative correlation and the task correlation.
(4) ***Real evaluation.*** We evaluate our model based on a virtual-world MMORPG dataset. Extensive experiments show the advantages of our method against all baselines.
(5) ***Real application.*** We perform case studies on applications for multi-social-temporal prediction in NetEase MMPRPGs, the results show that our method is capable of making good predictions in the actual business.

2 PRELIMINARIES

2.1 MMORPG Notations

Suppose there are N_p^t active players in an online game at the timestamp t, each of which generates N_{rela} kinds of player relations and N_{attr} kinds of player attributes. We set $N_{rela} = 4$ and $N_{attr} = 10$ in our MMORPG dataset, which is discussed in section 3. We represent the multi-relational attributed network as a weighted directed graph $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}, \mathbf{W}, \mathbf{A})^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{W}^t, \mathbf{A}^t)$, where $\mathcal{V}^t$ is a set of nodes (players) and $|\mathcal{V}^t| = N_p^t$, $\mathcal{E}^t$ is a set of edges (relations) and $\mathbf{W}^t \in \mathbb{R}^{N_p^t \times N_p^t \times N_{rela}}$ is a weighted adjacency matrix representing the nodes proximity, $\mathbf{A}^t \in \mathbb{R}^{N_p^t \times N_{attr}}$ is a weighted matrix representing the nodes attributes.

2.2 Social Time Series Prediction

The social time series prediction problem aims to learn a function $h_s(\cdot)$ that maps M historical multi-relational attributed network information to future H network node attributes, where $H \geq 1$:

$$[\mathcal{G}^{t-M+1}, \cdots, \mathcal{G}^t] \xrightarrow{h_s(\cdot)} [\mathbf{A}^{t+1}, ..., \mathbf{A}^{t+H}] \tag{1}$$

Note that we can change the size of H for next-step or multi-steps prediction.

2.3 Temporal Link Weight Prediction

Formally, the problem of temporal link weight prediction is defined as: given a sequence of multi-relational attributed networks from timestamp $t-M+1$ to timestamp t, the task is to predict the different types of link weights at timestamp $t+H$, where $H \geq 1$:

$$[\mathcal{G}^{t-M+1}, \cdots, \mathcal{G}^t] \xrightarrow{h_t(\cdot)} [\mathbf{W}^{t+1}, ..., \mathbf{W}^{t+H}] \tag{2}$$

Note that a special case of this definition is to predict whether a new link will emerge or not.

3 DATASET DESCRIPTION

3.1 Game Logs in MMORPG

We use a dataset of JusticePC[1], which is a popular MMORPG released by NetEase Games[2] and creates a breathing virtual world. Various activities performed by the players and state change events are recorded in the form of structured game logs through the game servers. The game logs consist of the following information.

- Timestamp: when a specific activity or event occurs
- Player information: the player's role ID, class, level, online time, virtual (real) money and game scores, etc.
- Event ID: an ID that identifies the type of activity (sending messages, trading virtual money, adding friends, etc.) or state change event type (level up, virtual money increasing or decreasing, etc.)
- Target player information: the other player information if the log is related to the interaction with another player
- Detailed information: depending on each activity type or event type, detailed information related to the activity or event

We have recorded more than 100 billion different types of game logs which contain more than 20 million character creation activity logs up to now. Specifically, we use the game logs of more than 654,492 users in the most active game server from 22rd June to 10th December, 2018. The dataset of JusticePC is now available for downloading[3]. Considering the privacy, the players in our dataset are ensured by anonymizing all personal identifiable information.

3.2 Player Attributes Construction

We extract a total of three categories of player attributes which include online time, game monies and game scores as the temporal datasets shown in Figure 4.

Online Time. The *online time* records the aggregated active duration of each player in one day. Thus it is a metric generally adopted by the game designers and operation teams to describe a player's immersion, satisfaction and stickiness to the games.

Game Monies. Two types of game monies exist: *virtual monies* and *real monies*, consumed in the game for different assets and use cases. The virtual monies are produced by the game system, as a prize to missions completion or activities participation, while the real monies are charged with real-world currencies.

Game Scores. The *experience score* is accumulated by completing the in-game missions and used for players to upgrade their characters. The *equipment score* is the synthetic estimation of the equipment worn by the character, positive related to the fighting force. Similarly, the *skill score* gives an overall estimation of learned skills and positively contributes to the combat power.

3.3 Social Networks Construction

We construct four different types of social networks from the game logs which include a transaction network, a friendship network, a team network and a chat network as the social datasets. They are visualized in Figure 4 and build up a dynamic multi-relational social networks between players.

[1]https://n.163.com/
[2]http://game.163.com/
[3]https://github.com/fuxiAIlab/JusticePC

The transaction network shows assets exchange relations between characters in the virtual world. Edges indicate the virtual currency of established transactions between characters.

The friendship network is built upon unidirectional friendship in online games. A character can send an invitation to another character and remove friends from his friendship lists.

The team network is made up of collaborative relations between characters. A team is temporally formed with the same goal and is disbanded after achieving the goal.

The chat network expresses the communication relationship between characters. A character can send a private message to other characters for individual communications.

We compare the four different social networks shown in Table 1.

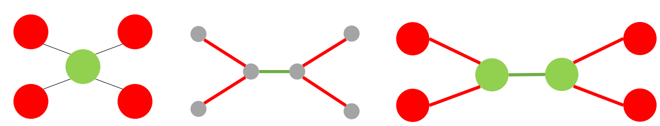
Table 1: Social Networks Comparison

Network	$\|V\|$	$\|E\|$	directionality	edges
transaction	149,328	14,473,464	directed	assets exchange
friendship	226,711	259,346,227	directed	friendship
team	223,877	10,985,058	undirected	teamwork
chat	179,981	14,029,059	directed	message delivery
total	329,405	298,833,808	both	multi-relations

4 PROPOSED GMTL

4.1 Enclosing Subgraph Construction

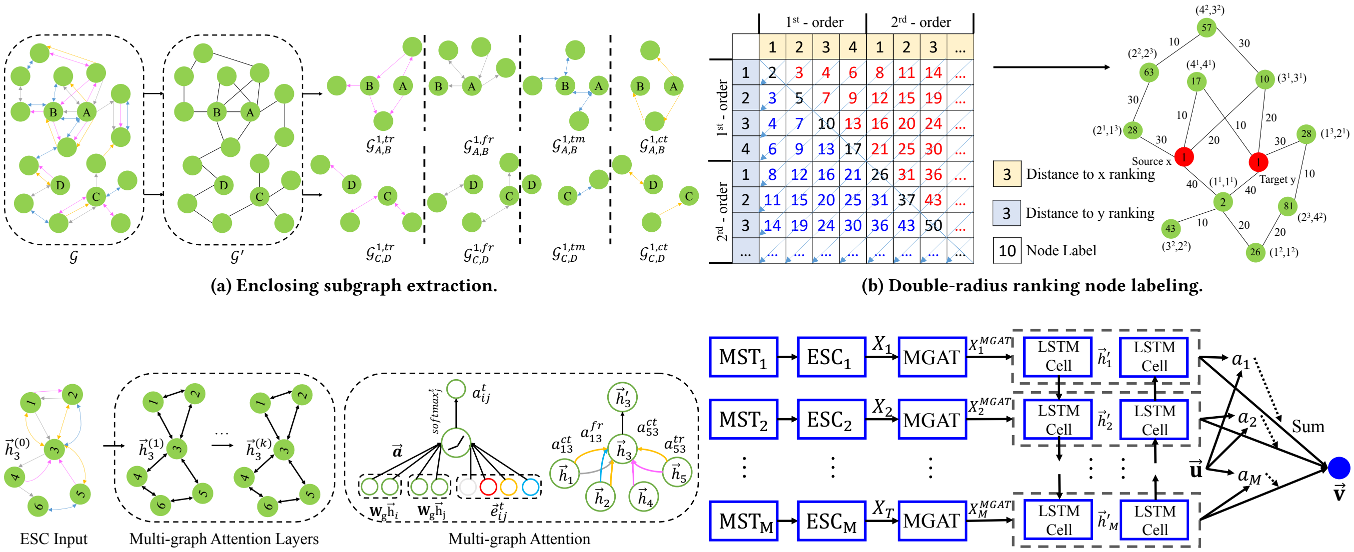
In Figure 7a, we consider the red nodes as the neighbours of the green node. Similarly, we consider the red edges as the neighbours of the green edge shown in Figure 7b. In multi-social temporal prediction tasks, we construct the enclosing subgraph shown in Figure 7c to extract both node neighbours and edge neighbours (the red ones) when predicting attributes of nodes or relations between nodes (the green ones).



(a) Node Neighbours (b) Edge Neighbours (c) Enclosing Subgraph

Figure 3: Illustration of enclosing subgraph.

Enclosing subgraph extraction. The first step is to extract enclosing subgraphs for a set of sampled source nodes and a set of sampled target nodes to construct the training data. Detailed illustration about enclosing subgraph can be seen in Section B. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we firstly extract an undirected and unweighted graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ where $e_{x,y}{}' \in \mathcal{E}'$ for each pair of nodes $x, y \in \mathcal{V}$ and each type of edges $e_{x,y}$ if $e_{x,y} \in \mathcal{E}$ while $e_{x,y}{}' \notin \mathcal{E}'$ if $e_{x,y} \notin \mathcal{E}$. Given two nodes $x, y \in \mathcal{V}$, the h-hop enclosing subgraph for (x, y) is the subgraph $\mathcal{G}^h_{x,y}$ induced from $\mathcal{G}'$ by the set of nodes $\{i|d(i, x) \leq h \ or \ d(i, y) \leq h\}$, where for any nodes $x, y \in \mathcal{V}$, $d(x, y)$ indicates the shortest path distance between x and y. Players share



(a) Enclosing subgraph extraction.

(b) Double-radius ranking node labeling.

(c) Multi-graph attention network.

(d) Graph attention recurrent network.

Figure 2: Illustration of the components of the proposed GMTL.

four different types of social relations, so we can construct four h-hop enclosing subgraphs $\mathcal{G}_{x,y}^{h,tr}, \mathcal{G}_{x,y}^{h,fr}, \mathcal{G}_{x,y}^{h,tm}, \mathcal{G}_{x,y}^{h,ct}$ for (x, y) with four different path distance functions $d_{tr}, d_{fr}, d_{tm}, d_{ct}$. Illustration of enclosing subgraph extraction can be seen in Figure 2a.

Node information extraction. The second step is to extract the node information matrix X for each enclosing subgraph. A graph neural network typically takes (W, X) as input, where W is the adjacency matrix of the input enclosing subgraph. X is the node information matrix, each row of which corresponds to a node's feature vector. This step is crucial for training a successful GNN model. The node information matrix X has three components: structural node labels, node attributes and node link weights. The purpose of node labeling is to use different labels to mark nodes' different roles in an enclosing subgraph. Figure 2b shows the details of our proposed double-radius ranking node labeling method. We ignore the graph directions and present a lookup table using (R_x^h, R_y^h) which represent the h-order distance ranking to the source and target nodes. Other than the structural node labels, the node information matrix X also provides an opportunity to include explicit attributes and links features. Given a node x, the node's attribute vector is composed of ten player attributes discussed in section 3.2. The node's link vector is associated with weights of different link types order by node labels. By concatenating each node's label/attribute/link vector to its corresponding row in X, we can make GNN simultaneously learn from three types of features. Detailed notations and equations are organized below:

$$
\begin{aligned}
L_x &= (L_x^{tr}, L_x^{fr}, L_x^{tm}, L_x^{ct}), \\
A_x &= (A_x^{ot}, A_x^{gm}, A_x^{gs}), \\
W_x &= (W_x^{tr}, W_x^{fr}, W_x^{tm}, W_x^{ct}), \\
X_x &= L_x || A_x || W_x.
\end{aligned}
\tag{3}
$$

4.2 Multi-graph Attention Network

Graph attention network (GAT)[20] specifies different weights to different nodes in a neighborhood, however, ignores the multiple edge types and edge weights information. Specifically, we conduct a multi-relational graph which leads to a **M**ulti-**G**raph **A**ttention ne**T**work (MGAT) that incorporates edge types and edge weights information. We will start by describing a single multi-graph attention layer to construct multi-graph attention network through stacking this layer as shown in Figure 2c.

The input to our layer is a set of nodes in enclosing subgraphs represented with node information matrix X. We use a four dimensional vector to represent the edge types and edge weights between two nodes. The four dimensions represent the four social relations between players and each edge constructs a vector:

$$
\begin{aligned}
&\overrightarrow{h}_i = X_i, \; i \in [1, N_{esc}], \\
&\overrightarrow{e}_{ij}^{\,t} = 4zeros(i = t, v = log(w_{ij}^t)), \; t \in \mathcal{N}_t.
\end{aligned}
\tag{4}
$$

where $4zeros(i, v)$ means a vector of 4 zeros and sets the i_{th} dimension to the value v, w_{ij}^t means the edge weight between player p_i and p_j with edge type t, $\mathcal{N}_t = \{tr, ct, fr, tm\}$.

In order to obtain sufficient expressive power to transform the input features into high-level features, at least one learnable linear transformation is required. To that end, as an initial step, a shared linear transformation, parametrized by a weight matrix $\mathbf{W_g}$, is applied to every node. We then perform a self-attention on the nodes—a shared attention mechanism computes attention coefficients that indicate the importance of node p_j's features to node p_i. We inject the graph structure into the mechanism by performing masked attention—we only use the first-order neighbors $\mathcal{N}_i$ of p_i

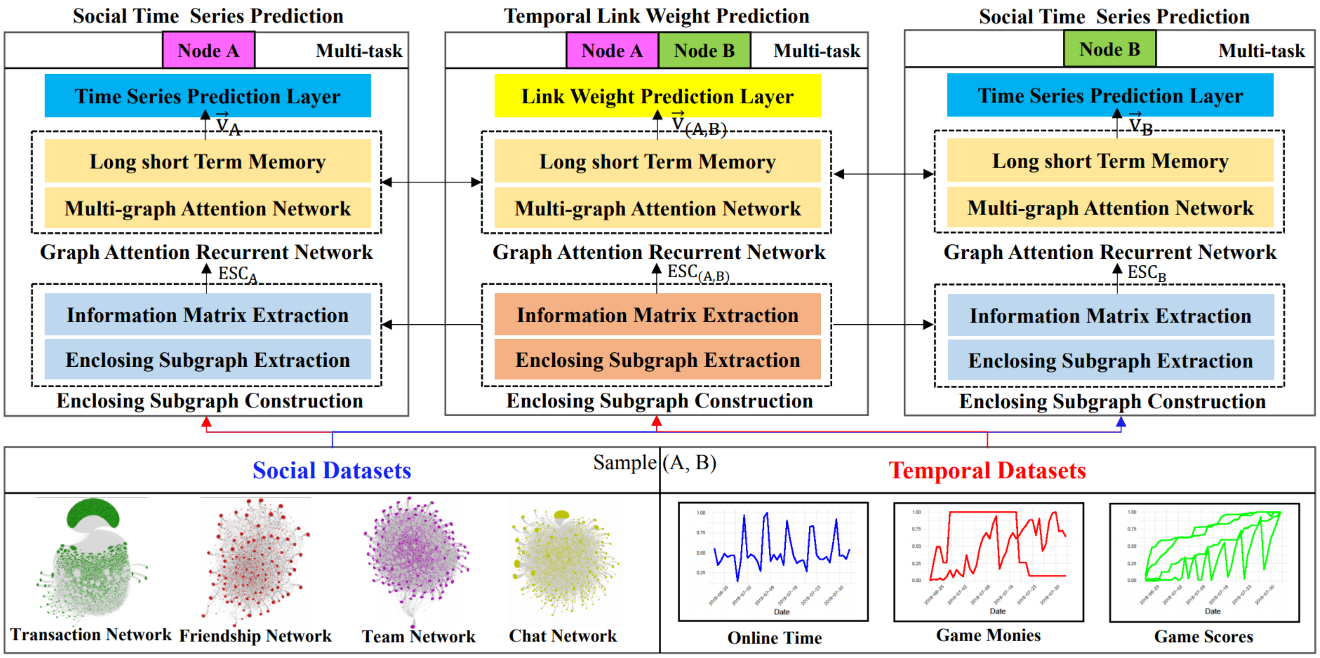


Figure 4: The proposed GMTL Model.

(including p_i):

$$a_{ij}^t = \frac{exp(\sigma(\overrightarrow{\mathbf{a}}^T[\mathbf{W_g}\overrightarrow{h}_i||\mathbf{W_g}\overrightarrow{h}_j||\overrightarrow{e}_{ij}^t]))}{\sum_{k\in\mathcal{N}_i, s\in\mathcal{N}_t} exp(\sigma(\overrightarrow{\mathbf{a}}^T[\mathbf{W_g}\overrightarrow{h}_i||\mathbf{W_g}\overrightarrow{h}_k||\overrightarrow{e}_{ik}^s]))}, \tag{5}$$

where σ is a nonlinear function like LeakyReLU, $.^T$ represents the transposition operation, $||$ is the concatenation operation and $\overrightarrow{\mathbf{a}}$ is a learnable context vector.

Once obtained, the normalized attention coefficients are used to compute a linear combination of the corresponding features, to serve as the final output features for every node (after potentially applying a nonlinearity σ like tanh):

$$\overrightarrow{h}'_i = \sigma(\sum_{j\in\mathcal{N}_i, t\in\mathcal{N}_t} a_{ij}^t \mathbf{W_g}\overrightarrow{h}_j). \tag{6}$$

After K multi-graph attention layers, we get X_i^{MGAT}.

4.3 Graph Attention Recurrent Network

In order to capture the data distribution by exploiting local stationarity and compositionality properties as well as the dynamic properties, we propose a **G**raph **A**ttention **R**ecurrent ne**T**work (GART) architecture shown in Figure 2d that are quite natural and stack a MGAT for feature extraction and a bi-directional long short term memory network (Bi-LSTM) with attention for sequence learning.

In the setting, X_t^{MGAT} is the output of the MGAT gate. We use a bi-directional long short term memory network to encode the feature learned by MGAT:

$$\begin{aligned} X_t^{MGAT} &= MGAT_{\mathcal{G}}(X_t) \\ \overrightarrow{h}_t &= \overrightarrow{\mathbf{LSTM}}(X_t^{MGAT}), \\ \overleftarrow{h}_t &= \overleftarrow{\mathbf{LSTM}}(X_t^{MGAT}), \\ \overrightarrow{h}'_t &= [\overrightarrow{h}_t||\overleftarrow{h}_t]. \end{aligned} \tag{7}$$

We finally apply the attention mechanism and introduce a context vector $\overrightarrow{\mathbf{u}}$ to measure the importance of the MSTs.

4.4 Multi-task Learning

The high level representation vector $\overrightarrow{\mathbf{v}}$ learned from GART can be used for both social time series prediction and temporal link weight prediction. We propose a multi-task learning model shown in Figure 4 that firstly samples source nodes and target nodes which show some interactions. Then we extract the enclosing subgraph and separate it into $\mathrm{ESC_{(A,B)}}, \mathrm{ESC_A}, \mathrm{ESC_B}$. We share parameters both in MGAT and LSTM model and finally make prediction with two different prediction layers: one for social time series prediction and the other for temporal link weight prediction. We use the linear transformation function to generate the prediction and train model by minimizing the mean square loss and square of mean absolute percentage loss:

$$\begin{aligned} \overrightarrow{\hat{Y}}_{stsp} &= \mathbf{W_{stsp}}\overrightarrow{\mathbf{v}} + \mathbf{b_{stsp}}, \\ \overrightarrow{\hat{Y}}_{tlwp} &= \mathbf{W_{tlwp}}\overrightarrow{\mathbf{v}} + \mathbf{b_{tlwp}}, \\ Loss &= \sum_{i=1}^{\xi}((Y_i - \hat{Y}_i)^2 + \gamma(\frac{Y_i - \hat{Y}_i}{Y_i})^2), \end{aligned} \tag{8}$$

Table 2: Social time series prediction performance comparison among different methods.

Algorithm	Online Time	Game Monies						Game Scores		
	online time	virtual balance	virtual income	virtual outcome	real balance	real income	real outcome	experience	equipment	skill
	4W / 1D,2D,3D / $RMSE_{stsp}$									
LAST	**11759,12413,13476**	4617,4947,5198	704,767,833	815,886,919	162,169,182	29.1,31.2,32.7	31.9,34.7,38.5	82,91,91	46,51,53	7.8,8.1,8.6
HA	10446,11718,12945	**4858,5189,5456**	**718,785,843**	**823,898,931**	**165,171,187**	**29.7,31.8,33.4**	**32.9,35.1,39.3**	**89,94,98**	**54,59,62**	**8.9,9.1,9.6**
ARIMA	9938,10989,12075	4417,4678,4981	697,759,821	809,878,905	161,164,179	28.4,29.9,31.7	31.1,33.4,37.5	86,93,93	50,56,59	8.4,8.8,9.1
GBRT	7985,9045,10997	2891,2989,3145	449,472,491	508,543,580	104,111,123	19.8,21.2,23.9	24.5,26.1,28.9	75,81,84	41,45,48	7.1,7.5,8.1
LSTM	7052,7932,9897	2851,2918,2997	403,446,476	478,502,534	97,102,109	15.6,17.9,21.3	20.1,23.9,25.1	65,69,77	33,35,40	6.3,6.7,6.9
$GART_{!mgat}$	6817,7589,9476	2617,2710,2789	378,401,434	456,481,502	90,97,99	13.9,16.1,19.9	18.7,22.1,24.5	60,62,71	29,31,36	5.8,6.1,6.4
$GART_{!lstm}$	7217,8276,9981	2883,2977,3091	423,454,482	489,513,551	100,109,117	16.4,18.9,21.8	21.9,24.8,26.7	69,76,81	38,41,45	6.6,7.0,7.6
$GART_{sin}$	6517,7312,9219	2457,2592,2671	354,378,411	422,454,490	88,92,96	13.2,15.3,18.9	17.8,20.1,22.3	52,58,65	25,27,35	5.5,5.9,6.1
$GART_{mul}$	6088,6965,8767	2189,2303,2387	331,346,359	398,412,443	81,84,89	12.5,13.9,16.5	16.4,18.8,20.1	49,54,59	21,24,29	5.1,5.4,5.8
GMTL	**5417,6432,7956**	**1798,1883,1976**	**297,312,327**	**354,379,413**	**72,76,83**	**11.4,12.4,14.3**	**15.1,16.6,17.8**	**42,46,51**	**18,20,27**	**4.5,4.9,5.1**
	4W / 1D,2D,3D / $MAPE_{stsp}$(%)									
LAST	**40.15,40.45,40.98**	28.61,29.02,29.33	18.96,19.22,19.86	15.15,15.66,15.80	13.84,14.28,14.40	10.47,10.65,10.85	11.26,11.57,11.88	5.23,5.34,5.41	14.73,14.81,14.96	5.58,5.78,5.83
HA	40.01,40.08,40.17	**29.78,30.12,30.78**	**20.11,20.76,21.33**	**16.76,17.11,17.45**	**14.76,15.01,15.45**	**11.56,11.98,12.04**	**12.09,12.34,12.67**	**6.19.6.27,6.54**	**15.01,15.25,15.65**	**6.12,6.33,6.58**
ARIMA	38.84,39.34,39.89.	26.24,26.90,27.18	17.79,18.12,18.43	14.01,14.43,14.87	12.75,13.11,13.49	10.12,10.29,10.37	10.93,11.02,11.17	5.49,5.77,5.81	14.42,14.54,14.71	6.01,6.45,6.51
GBRT	31.31,32.78,33.09	19.79,21.11,21.89	14.68,15.14,15.97	11.94,12.45,12.99	8.88,9.45,9.91	9.85,9.99,10.07	10.19,10.45,10.77	4.79,4.90,5.12	13.46,13.89,14.17	5.08,5.45,5.81
LSTM	27.56,27.67,27.90	19.52,19.66,19.71	13.18,13.44,13.60	11.23,11.32,11.46	8.29,8.35,8.49	7.76,7.89,8.01	8.36,8.55,8.90	4.15,4.28,4.33	10.84,11.28,11.92	4.51,4.59,4.67
$GART_{!mgat}$	26.64,26.89,27.26	17.92,18.44,18.90	12.36,12.77,12.98	10.72,10.98,11.02	7.69,7.81,7.90	6.91,7.12,7.45	7.78,7.91,8.12	3.83,3.91,4.04	9.52,10.02,10.44	4.15,4.26,4.43
$GART_{!lstm}$	28.20,28.87,29.18	19.74,19.81,20.10	13.83,14.09,14.25	11.49,11.57,11.68	8.54,8.66,8.72	8.16,8.55,8.91	9.11,9.35,9.74	4.40,4.59,4.68	12.48,12.89,13.11	4.72,4.81,4.98
$GART_{sin}$	25.47,25.76,25.99	16.82,17.15,17.64	11.57,11.88,12.15	9.92,10.09,10.15	7.52,7.59,7.63	6.57,6.69,6.81	7.40,7.54,7.68	3.32,3.49,3.57	8.21,8.76,9.01	3.94,3.99,4.17
$GART_{mul}$	23.79,24,11,24.54	14.99,15.17,15.68	10.82,11.12,11.54	9.35,9.44,9.76	6.92,7.01,7.14	6.22,6.34,6.49	6.82,7.02,7.21	3.13,3.19,3.27	6.90,7.15,7.56	3.65,3.76,3.81
GMTL	**21.17,22.34,24.42**	**12.31,13.43,14.78**	**9.71,9.92,10.35**	**8.32,8.66,8.89**	**6.15,6.66,7.12**	**5.67,5.89,6.19**	**6.28,6.87,7.01**	**2.68,2.85,2.91**	**5.91,6.16,6.47**	**3.22,3.54,3.81**

Table 3: Temporal link weight prediction performance comparison among different methods.

Algorithm	Transaction	Friendship	Team	Chat	Transaction	Friendship	Team	Chat
	4W / 1D,2D,3D / RMSE_{tlwp}				4W / 1D,2D,3D / MAPE_{tlwp}(%)			
LAST	792,881,922	0.38,0.57,0.74	**2.4,2.8,3.2**	**10.4,11.4,12.7**	15.78,15.91,16.04	9.69,9.89,10.01.	**8.15,8.46,8.64**	**14.17,14.35,14,78**
HA	**801,897,945**	**0.43,0.61,0.82**	2.0,2.4,2.7	9.8,10.9,12.1	**16.11,16.45,16.88**	**10.30,10.41,10.54**	7.40,7.77,7.91	12.98,13.19,13.45
PH	786,859,901	0.31,0.52,0.66	1.8,2.1,2.5	9.4,10.4,11.7	14.93,15.14,15.54	7.91,8.26,8.81	6.66,6.91,7.09	12.45,12.66,12.79
DNE	715,798,864	0.24,0.51,0.59	1.4,1.6,2.1	7.8,8.8,9.5	14.68,14.76,14.81	6.12,6.78,7.01	5.18,5.62,5.86	10.33,10.68,11.45
LSTM	684,753,841	0.21,0.46,0.53	1.1,1.4,1.5	7.1,7.8,8.8	14.44,14.51,14.57	5.35,5.64,5.77	4.07,4.67,4.90	9.40,9.67,9.71
$GART_{!mgat}$	662,737,821	0.18,0.31,0.44	0.9,1.2,1.3	6.9,7.4,8.4	13.51,13.68,13.79	4.59,4.88,5.07	3.33,3.58,3.79	9.14,9.26,9.37
$GART_{!lstm}$	701,776,854	0.23,0.49,0.56	1.2,1.5,1.9	7.4,8.2,9.0	13.90,14.01,14.13	5.87,5.95,6.04	4.44,4.76,4.91	9.80,9.91,9.98
$GART_{sin}$	652,729,809	0.14,0.23,0.36	0.8,1.1,1.1	6.6,7.2,8.1	13.19,13.34,13,56	3.57,3.89,4.65	2.96,3.02,3.11	8.74,8.88,8.95
$GART_{mul}$	616,701,776	0.10,0.15,0.29	0.6,0.8,0.9	6.2,6.6,7.4	11.71,11.98,12.43	2.55,2.87,3.04	2.22,2.45,2.51	8.21,8.44,8.67
GMTL	**581,674,732**	**0.02,0.09,0.18**	**0.4,0.5,0.5**	**5.4,5.9,6.7**	**10.11,10.56,10.93**	**0.51,0.76,0.83**	**1.48,1.82,1.98**	**7.15,7.54,7.81**

where ξ represents the number of samples and Y_i means the real value of the sample i. The loss function used for jointly training our propose GMTL is defined as:

$$L(\theta) = \beta_1 Loss_{stsp}^{A} + \beta_2 Loss_{tlwp}^{(A,B)} + \beta_3 Loss_{stsp}^{B}, \tag{9}$$

where θ refer to all learnable parameters in the GMTL and $\gamma, \beta_{1,2,3}$ are hyper-parameters. We use Tensorflow and Keras to implement our proposed model[3] and use Adam for optimization.

5 EXPERIMENTS

5.1 Experimental Settings

The time unit in online games is set to 1 day. We fill missing values after data cleaning by 0 (like online time, money income, money outcome, chat etc) or the latest value in history (like money balance, friendship etc). In addition, data inputs are normalized by Min-max method. In the experiment, we consider only the 1-order neighbours to construct 1-hop enclosing subgraphs. We execute grid search strategy to locate the best parameters on validations. It is confirmed that in all the tests, data in a historical time window of 4 weeks, i.e., 28 observed data points ($M = 28$) are used to forecast player attributes and relationships in the next 1,2 and 3 days ($H = 1, 2, 3$).

5.2 Evaluation Metrics

Social Time Series Prediction. We use Root Mean Square Error ($RMSE_{stsp}$) and Mean Absolute Percentage Error ($MAPE_{stsp}$) to evaluate the social time series prediction results, which is defined as follows:

$$RMSE_{stsp} = \sqrt{\frac{1}{\xi}\sum_{i=0}^{\xi}(\hat{y}_i^{t+1} - y_i^{t+1})^2} \tag{10}$$

$$MAPE_{stsp} = \frac{1}{\xi}\sum_{i=0}^{\xi}|\frac{\hat{y}_i^{t+1} - y_i^{t+1}}{y_i^{t+1}}| \tag{11}$$

where $\hat{y}_i^{t+1}$ and y_i^{t+1} mean the predicted attribute value and the real attribute value of player i at timestamp $t+1$, and ξ is the total number of samples. The players' attribute values are non-negative, so they are corrected to 0 for all predicted negative numbers.

[3]https://github.com/fuxiAIlab/GMTL

Temporal Link Weight Prediction. We measure the accuracy of temporal link weight prediction with Root Mean Square Error ($RMSE_{tlwp}$) and Mean Absolute Percentage Error ($MAPE_{tlwp}$), which is defined as follows:

$$RMSE_{tlwp} = \sqrt{\frac{1}{\xi}\sum_{i=0}^{\xi}(\hat{w}_{i,j}^{t+1} - w_{i,j}^{t+1})^2} \tag{12}$$

$$MAPE_{tlwp} = \frac{1}{\xi}\sum_{i=0}^{\xi}\left|\frac{\hat{w}_{i,j}^{t+1} - w_{i,j}^{t+1}}{w_{i,j}^{t+1}}\right| \tag{13}$$

where $\hat{w}_{i,j}^{t+1}$ and $w_{i,j}^{t+1}$ mean the predicted link weight and the real link weight between player i and player j at timestamp $t+1$, and ξ is the total number of samples. The players' link weight values are non-negative, so they are corrected to 0 for all predicted negative numbers.

5.3 Model Comparison

Social Time Series Prediction. We compare our model with the following methods: 1). Last Value (**LAST**); 2). Historical Average (**HA**); 3). Autoregressive Integrated Moving Average (**ARIMA**)[1]; 4). Gradient Boosting Regression Tree (**GBRT**)[4]; 5). Long Short-Term Memory (**LSTM**)[6]. We tune the parameters for all methods and then report the best performance.

Temporal Link Weight Prediction. We compare our model with the following methods: 1). Last Value (**LAST**); 2). Historical Average (**HA**); 3). Popular Heuristics (**PH**)[25]; 4). Dynamic node embedding (**DNE**)[28]; 5). Long Short-Term Memory (**LSTM**)[6]. We tune all the parameters and report the best performance.

Model Variant Comparison. To further investigate the effectiveness of each model component, we compare GMTL with its variants as follows: 1). graph attention recurrent network replacing MGAT to GAT (**GART$_{!mgat}$**); 2). graph attention recurrent network replacing LSTM to dense layers (**GART$_{!lstm}$**); 3). graph attention recurrent network for single attribute or single relationship prediction (**GART$_{sin}$**); 4). graph attention recurrent network for multiple attributes or multiple relationships prediction (**GART$_{mul}$**).

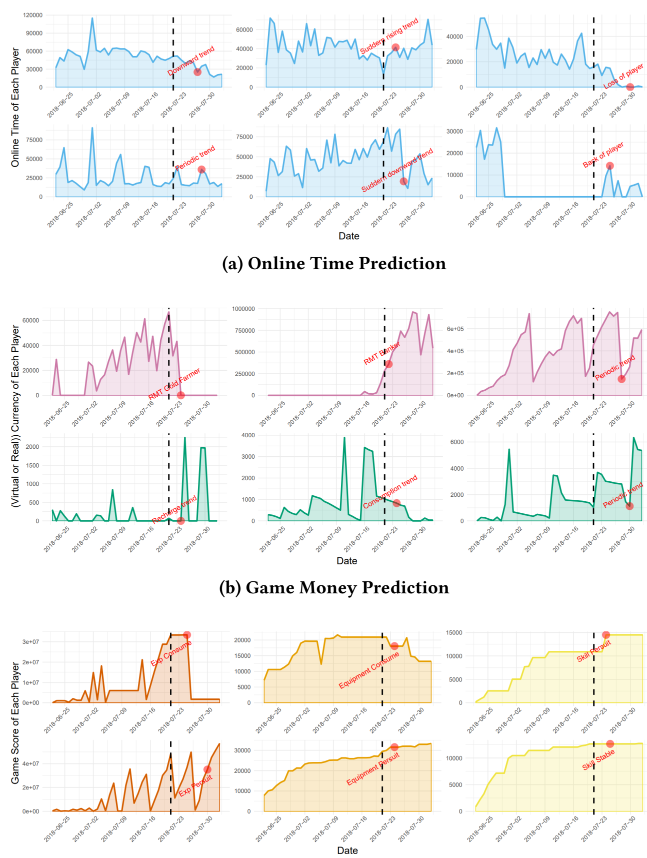
5.4 Performance Comparison

Social Time Series Prediction. Table 2 shows the performance of the proposed method as compared to all other competing methods. GMTL achieves the lowest $RMSE_{stsp}$ and $MAPE_{stsp}$ among all the methods. More specifically, we can see that LAST, HA and ARIMA perform poorly, as they rely purely on players' own historical values for prediction. GBRT further considers grouping patterns and therefore achieves better performance. LSTM generally achieves better prediction results than traditional machine learning models. However, these five baselines do not model the social dependency. Consequently, our proposed GART significantly outperforms those methods. GART$_{sin}$ outperforms GART$_{!mgat}$ and GART$_{!lstm}$ which shows the reasonable design of MGAT and recurrent components in GART. GART$_{mul}$ outperforms GART$_{sin}$ since the former one considers the collaborative correlations between attributes and relationships. Finally, we can find that GMTL additionally considering the task correlation outperforms the methods which only consider social, temporal and collaborative correlations.

Temporal Link Weight Prediction. Table 3 demonstrates that the results of our proposed methods and baselines on the MMORPG dataset. Our proposed GMTL achieves the best $RMSE_{tlwp}$ and $MAPE_{tlwp}$ among all the baselines. We can easily observe that LAST, HA, PH, LSTM and DNE perform worse due to their incapability of handling complex social-temporal data. GART that considering the social temporal correlations show some advantages and the advantage of MGAT and recurrent components are also valified. GART$_{mul}$ outperforms GART$_{sin}$ mainly due to the collaborative correlation included. From the performance comparison, we can conclude that it is harder to make multi-step prediction than single-step prediction both in social time series prediction and temporal link weight prediction.

6 APPLICATIONS

We perform case studies on applications for multi-social-temporal prediction tasks in MMORPGs, to show that the GMTL is capable of making good predictions in the actual business in NetEase Games.



(a) Online Time Prediction

(b) Game Money Prediction

(c) Game Score Prediction

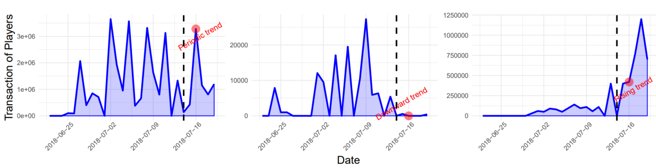
Figure 5: Case study on time series prediction in JusticePC

6.1 Social Time Series Prediction

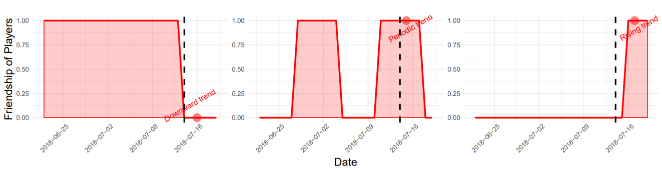
Online Time Prediction. As shown in Figure 5a, the GMLT method grasps the periodicity and long-term trend of the players' activeness to predict the future values of their online time in the coming days, and moreover, it is able to predict the abrupt changes. Operation teams use these predictions as a valuable feedback to improve the game or to take specific corrective actions. For example, when a potential reduction of a player's online time or even a churn is predicted, retention methods will be readily applied, such as phone interviews and retention gifts. Likewise, the return process can be accelerated if an imminent return is predicted, by pushing messages or offering extra discounts.

Game Money Prediction. In Figure 5b, we present the predictions of six game money attributes of a player. The virtual money predictions are usually applied to the smurfing sniffing, which aims at detecting the potential real money trading in the games and diagnose such illegal economic systems. We have tested the effect of the GMTL which outperformed [18] and underperformed [17]. In terms of real money monitoring and predicting, it's mainly used to players charge and purchase estimation.

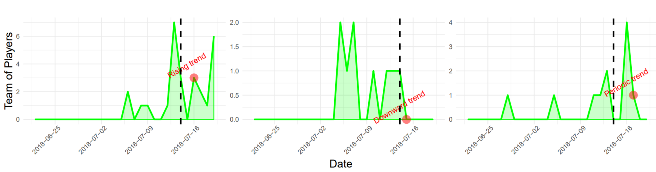
Game Score Prediction. Figure 5c illustrates the predictions of three game scores respectively. The changes of different attributes imply different goals of the players to cultivate their in-game characters. Provided a foresee of the experience score rise, props and game-plays to help quickly gain experience are shown on the front page of the player.



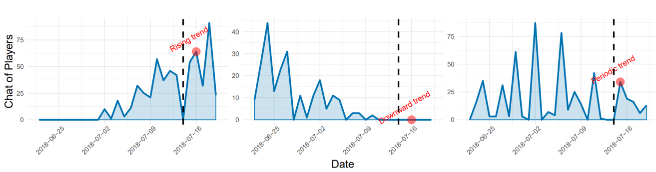
(a) Transaction Prediction



(b) Friendship Prediction



(c) Team Prediction



(d) Chat Prediction

Figure 6: Case study on link weight prediction in JusticePC

6.2 Temporal Link Weight Prediction

Transaction Prediction. As depicted in Figure 6a, the future transaction strength between players is predicted by the GMTL. With a careful surveillance and comparison of the prediction values and actual values, the operation teams get a better chance to catch the anomaly trades and may even prevent those trades in advance.

Friendship Prediction. Figure 6b shows the predictions of the metabolic friendship between players. This facilitates the friend management system to make recommendations in the "people you may know" entry and enable the one-click friend-list maintenance.

Team Prediction. The predictions of team-up relationship are presented in Figure 6c. It enables the teammates matchmaking system suggest suitable teammates proactively when a player starts up a team or when the team leader starts the recruitment procedure.

Chat Prediction. As illustrated in Figure 6d, the GMTL can as well predict the strength of the chat between players. This result is generally adopted by the chatting system to bring up in the chatting page those dialog boxes of frequently communicated friends.

6.3 Application Performance Comparison

We compare the GMTL with the baseline and the state-of-the-art methods in applications. For social time series prediction, we use the precision|recall|f1-score as the evaluation metrics while for temporal link weight prediction, we use the click-through rate|conversion rate as the evaluation metrics.

Table 4: Applications for social time series prediction

Applications	Baseline	GMTL	State-of-the-art
Churn Prediction	85.18%|84.69%|84.93%	**91.27%|89.76%|90.51%**	91.27%|89.76%|90.51%
Return Prediction	44.15%|40.28%|42.13%	**52.89%|47.38%|49.98%**	52.89%|47.38%|49.98%
Farmer Detection	76.75%|64.45%|70.06%	81.69%|72.51%|76.82%	**89.76%|79.67%|84.41%**
Banker Detection	55.13%|41.72%|47.50%	70.53%|61.19%|65.53%	**78.04%|65.67%|71.32%**
Buyer Detection	22.88%|21.20%|22.01%	71.68%|64.37%|67.83%	**78.12%|70.05%|73.87%**
Recharge Prediction	70.12%|62.89%|66.31%	75.71%|67.79%|71.53%	**78.14%|70.27%|74.00%**
Consumption Prediction	71.12%|69.89%|70.50%	79.87%|76.93%|78.37%	**81.27%|78.88%|80.06%**
Experience Pursuit	86.17%|87.28%|86.72%	**91.87%|90.73%|91.30%**	91.87%|90.73%|91.30%
Equipment Pursuit	84.19%|86.21%|85.19%	**89.96%|88.17%|89.06%**	89.96%|88.17%|89.06%
Skill Pursuit	85.89%|82.78%|84.31%	**90.11%|87.27%|88.67%**	90.11%|87.27%|88.67%

Table 5: Applications for temporal link weight prediction

Applications	Baseline	GMTL	State-of-the-art
Abnormal Transaction Detection	51.34%|48.94%	64.18%|58.98%	**71.32%|62.77%**
Friendship Suggestion	53.19%|35.98%	71.87%|58.17%	**78.14%|65.23%**
Friends List Cleanup	54.94%|51.02%	64.56%|59.19%	**69.10%|63.44%**
Team Formation	75.56%|71.87%	**89.17%|81.18%**	89.17%|81.18%
Chat Fast Entry	78.98%|72.66%	**84.71%|79.14%**	84.71%|79.14%

7 CONCLUSION AND FUTURE WORK

We propose a novel graph attention recurrent network based multi-task learning model for multi-social-temporal prediction in online games. It is the first work that introduces the multi-social temporal prediction problem in online games considering the temporal, social, collaborative and task correlations. We evaluate our model on an MMORPG dataset and the experiments show that the performance of social time series prediction and temporal link weight prediction can be significantly improved. In the future, we will evaluate our model with other types of game datasets, such as First Person Shooter (FPS) and Multiplayer Online Battle Arena (MOBA) etc. We will also expand our work to real-life multi-social-temporl data like Facebook, Tweeter, WeChat and Weibo etc.

8 RELATED WORK

8.1 Social Time Series Prediction

Autoregression-based models (e.g., ARIMA[1] and VAR[29]), classical machine learning models (e.g., SVR[15] and GBRT[4]), deep learning models (e.g., LSTM[6] and seq2seq[16]) are widely used in time series prediction. Compared to traditional multivariate time series, social time series have their own characteristics, e.g., social correlation. These works can only capture temporal dependency in time series, which ignore the unique characteristics of MST data. To overcome this problem, cross-domain fusion-based methods [9, 10, 12, 21, 22, 24] show superiority in many spatio-temporal (similar to social-temporal) applications.

8.2 Temporal Link Weight Prediction

Temporal link weight prediction analyzes the evolution patterns of a sequence of networks over time. To tackle this task, [3] provides a method of combining matrix-based and tensor-based techniques; [5] proposes a unified model based on latent matrix factorization and graph regularization; [19] develops graph-based link prediction techniques that incorporate the temporal information contained in evolving social networks; [7] investigates hybrid link prediction methods that combine the power of the time-series model in predicting repeated link occurrences with the ability of static graph link prediction methods to identify new link occurrences; [23] leverages the time-dependent matrix factorization and the network propagation constraint.

8.3 Graph Neural Network

In recent years, many efforts have been made to extend deep neural network models for graph structured data. These neural network models that are applied to graphs are known as graph neural network models[11, 13, 20, 26]. Most of the current graph neural network models are designed for static graphs where nodes and edges are fixed. However, many real-world graphs are evolving. It has been of great interest to study the properties of dynamic graphs. [14] introduces Graph Convolutional Recurrent Network (GCRN), a deep learning model able to predict structured sequences of data. To the best of our knowledge, our work is the first work that is motivated by the graph neural network in online games.

8.4 Multi-task Learning

Multi-task learning (MTL) learns multiple related tasks simultaneously to improve generalization performance. Many MTL approaches have been proposed in the past[2, 8, 27]. The task relatedness can be modeled by constraining multiple tasks to share a common underlying structure, e.g., a common set of features, or a common subspace. MTL approaches have been applied in many domains. Compared to learning separately, jointly learning multiple relative tasks can demonstrate significant performance gains for each task. To our best knowledge, ours is the first work that applies MTL for multi-social-temporal prediction tasks.

REFERENCES

[1] George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association* 65, 332 (1970), 1509–1526.

[2] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[3] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5, 2 (2011), 10.

[4] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[5] Sheng Gao, Ludovic Denoyer, and Patrick Gallinari. 2011. Temporal link prediction by integrating content and structure information. In *Proceedings of the 20th ACM international conference on Information and knowledge management.* ACM, 1169–1174.

[6] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[7] Zan Huang and Dennis KJ Lin. 2009. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing* 21, 2 (2009), 286–303.

[8] Abhishek Kumar and Hal Daume III. 2012. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417* (2012).

[9] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[10] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction.. In *IJCAI.* 3428–3434.

[11] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International conference on machine learning.* 2014–2023.

[12] Zheyi Pan, Yuxuan Liang, Junbo Zhang, Xiuwen Yi, Yong Yu, and Yu Zheng. 2018. HyperST-Net: Hypernetworks for Spatio-Temporal Forecasting. *arXiv preprint arXiv:1809.10889* (2018).

[13] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.

[14] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing.* Springer.

[15] Alex J Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and computing* 14, 3 (2004), 199–222.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems.*

[17] Jianrong Tao, Jianshi Lin, Shize Zhang, Sha Zhao, Runze Wu, Changjie Fan, and Peng Cui. 2019. MVAN: Multi-view Attention Networks for Real Money Trading Detection in Online Games. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM.

[18] Jianrong Tao, Jiarong Xu, Linxia Gong, Yifu Li, Changjie Fan, and Zhou Zhao. 2018. NGUARD: A Game Bot Detection Framework for NetEase MMORPGs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM, 811–820.

[19] Tomasz Tylenda, Ralitsa Angelova, and Srikanta Bedathur. 2009. Towards time-aware link prediction in evolving social networks. In *Proceedings of the 3rd workshop on social network mining and analysis.* ACM, 9.

[20] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv:1710.10903* (2017).

[21] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

[22] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[23] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Haifeng Chen, and Wei Wang. 2017. Link Prediction with Spatial and Temporal Consistency in Dynamic Networks.. In *IJCAI.* 3343–3349.

[24] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence.*

[25] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems.* 5171–5181.

[26] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

[27] Renjie Zheng, Junkun Chen, and Xipeng Qiu. 2018. Same representation, different attentions: shareable sentence representation learning from multiple tasks. *arXiv preprint arXiv:1804.08139* (2018).

[28] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. 2018. Dynamic network embedding by modeling triadic closure process. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

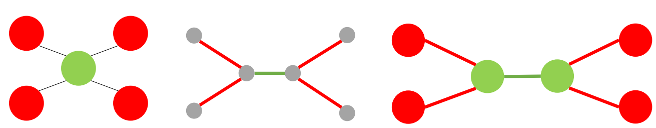
[29] Eric Zivot and Jiahui Wang. 2006. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-PLUS®* (2006), 385–429.

A NETEASE GAMES

Living up to the company's motto of "good games have no borders", NetEase Games has made great popularity and traction in domestic and overseas markets. In 2018, revenues of NetEase's online gaming segment reached $5.845 billion, up 10.8% year over year. NetEase Games continues to maintain high growth rate and deliver new hit games. To commit to the pursuit of the highest quality games and player experience, NetEase Games has developed and published dozens of popular games especially MMORPGs on pc and mobile, including JusticePC, Ghost II Online, Ghost Mobile, Fantasy Westward Journey Mobile, Fantasy Westward Journey Online etc. Other genes of online games produced by NetEase Games are also famous worldwide such as Onmyoji, Identity V, Knives Out Mobile, Knives Out Plus, Rules of Survival, LifeAfter, Fever Basketball etc.

B ENCLOSING SUBGRAPH

In Figure 7a, we consider the red nodes as the neighbours of the green node. Similarly, we consider the red edges as the neighbours of the green edge shown in Figure 7b. In multi-social temporal prediction tasks, we construct the enclosing subgraph shown in Figure 7c to extract both node neighbours and edge neighbours (the red ones) when predicting attributes of nodes or relations between nodes (the green ones).



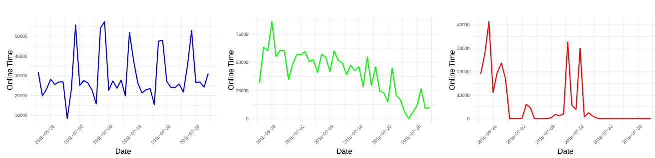
(a) Node Neighbours **(b) Edge Neighbours** **(c) Enclosing Subgraph**

Figure 7: Illustration of enclosing subgraph.

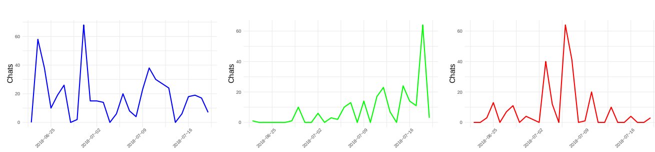
C CORRELATION ANALYSIS

C.1 Temporal Correlation

Figure 8 shows the curves of online time attribute and chat relation. The curves all show an empirical temporal correlation in time series, namely, period, trend and sudden of a specific attribute or relation.



(a) Period of online time **(b) Trend of online time** **(c) Sudden of online time**



(d) Period of chats **(e) Trend of chats** **(f) Sudden of chats**

Figure 8: Temporal correlations of attributes and relations

C.2 Social Correlation

We visualize the three attributes (online time, virtual money and experience) between selected player and his nearest trader, friend, teammate and chatter in Figure 9. Additionally, two relations (transaction and chat) between selected players and his nearest traders, friends, teammates and chatters are visualized in Figure 10. We apply the Min-max normalization and compute the cosine similarity between players. Similarity matrix plots show the obvious social correlations over players' attributes and relations.

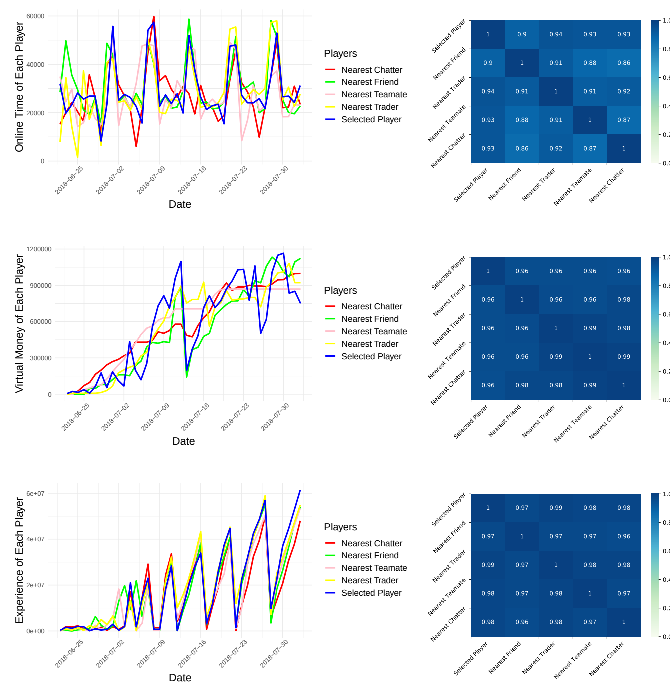


Figure 9: Social correlations of attributes

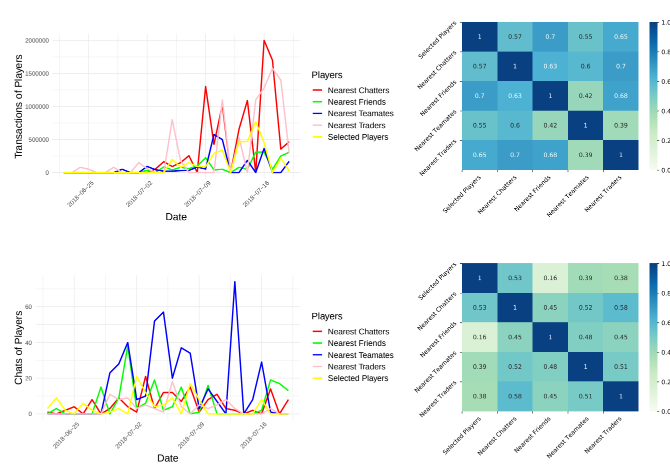


Figure 10: Social correlations of relations

C.3 Collaborative Correlation

We extract the ten attributes from two players and the four relations from two pairs of players randomly selected from the MMORPG dataset. We firstly normalize the data with Min-max method and then visualize them in Figure 11. We also compute the cosine similarity between attributes or relations and figure out that players' attributes and relations react great collaborative correlations.

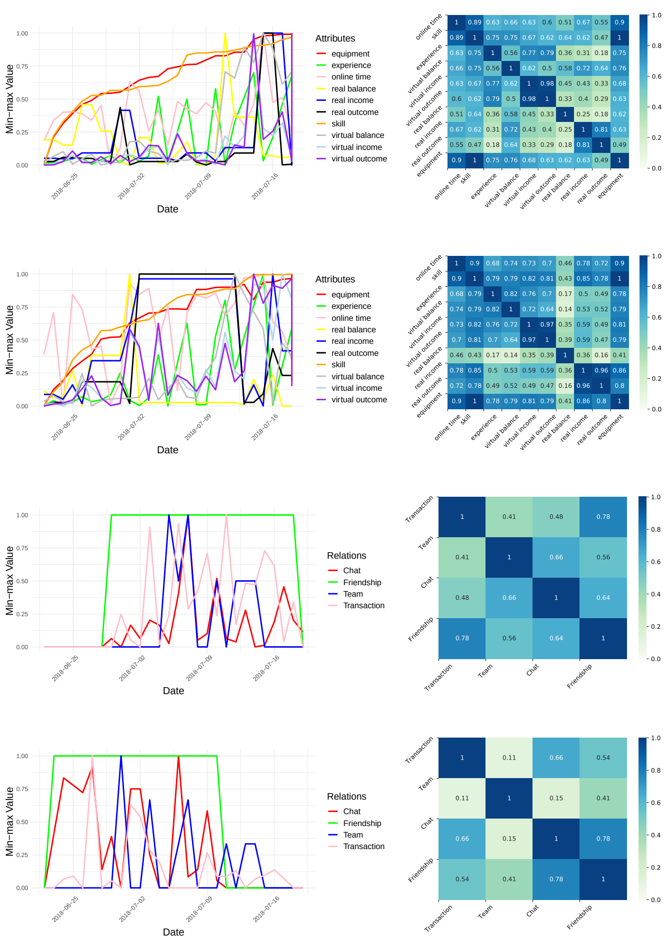


Figure 11: Collaborative correlations illustration

C.4 Task Correlation

In Figure 12, we randomly select two players from the MMORPG dataset and visualize their attributes' and relations' curves. We compute the cosine similarity separately on each player's attributes and their relations over the Min-max normalized values. We conclude that it shows surprising correlations between social time series prediction tasks and temporal link weight prediction tasks.

D DATASET DETAILS

In the experiment, the data from 2018/06/22 to 2018/09/30 is used for training (101 days), the data from 2018/10/01 to 2018/10/31 is used for validation (31 days), and the data from 2018/11/01 to 2018/12/10 is used for testing (40 days). We filter the samples with active days less than 7. This is a common practice used in industry because we do not care about such low-active scenarios. Table 6 shows the volume and details of dataset segmentation in our experiments.

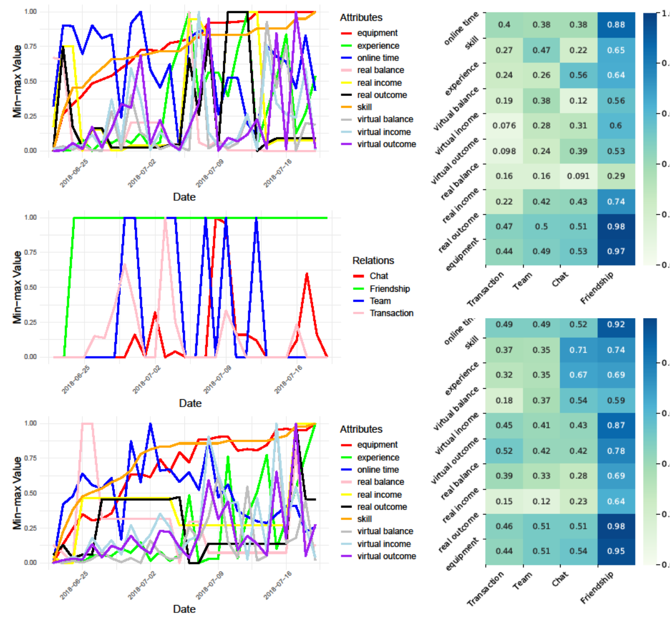


Figure 12: Task correlations of attributes and relations

Table 6: Statistics of the dataset segmentation.

Segmentation	Period	Transaction		Friendship	
		‖V‖	‖E‖	‖V‖	‖E‖
Training	2018.06.22-2018.09.30	101,478	8,414,804	154,682	150,782,690
Validation	2018.10.01-2018.10.31	89,297	2,524,441	104,781	45,234,807
Test	2018.11.01-2018.12.10	91,471	3,534,219	135,398	63,328,730
Segmentation	**Period**	**Team**		**Chat**	
		‖V‖	‖E‖	‖V‖	‖E‖
Training	2018.06.22-2018.09.30	135,479	6,386,661	124,891	8,156,429
Validation	2018.10.01-2018.10.31	119,890	1,915,998	91,365	2,446,928
Test	2018.11.01-2018.12.10	104,135	2,682,399	105,887	3,425,702

E DETAILED EXPERIMENTAL SETTINGS

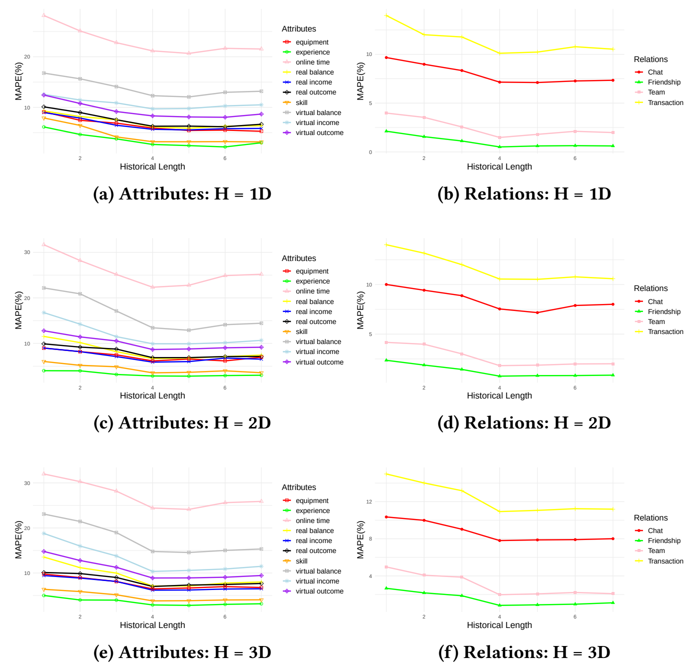
In this section, we show the detailed experimental settings:

- **LAST**: We use an obvious baseline of just predicting the last known values, $\mathbf{A}^{t+H} = \mathbf{A}^{t}$, $\mathbf{W}^{t+H} = \mathbf{W}^{t}$.
- **HA**: We use weighted average of previous seasons as the prediction. The period used is 1 week, and the prediction is based on aggregated data from previous weeks. For example, the prediction for this Sunday is the average value from last four Sundays.
- **ARIMA**: We use ARIMA with Kalman filter which is implemented using the statsmodel python package. The orders are (3,0,1).
- **GBRT**: We feed the 28 dimensional attribute vector into GBRT which is an ensemble method for the regression tasks and widely used in practice.
- **PH**: Graph structure features locate inside the observed node and edge structures of the network. We compute eight popular

heuristics: common neighbours, Jaccard, preferential attachment, Adamic-Adar, resource allocation, Katz, PageRank and SimRank. We then feed them into a linear regression model to fit the values.

- **DNE**: 64 dimensional node embedding vector is learned on single evolving relation networks. We then feed the latent properties of nodes into a feed forward neural network to fit the values.
- **LSTM**: We feed the 28 dimensional attribute vector into LSTM with one recurrent layer which contains 128 units.
- **GART$_{!\mathbf{mgat}}$**: We separately apply GAT to four different networks and then concat them for sequence learning.
- **GART$_{!\mathbf{lstm}}$**: We repalce the LSTM with a totally dense layer after applying MGAT. The first dense layer has 64 units and the second dense layer has 32 units.
- **GART$_{\mathbf{sin}}$**: We only consider one attribute or one relation for prediction. We may use the four relation networks considering the social correlations but ignore the collaborative correlations.
- **GART$_{\mathbf{mul}}$**: Ten attributes or four relations are trained and predicted jointly which considering the collaborative correlation.
- **GMTL**: It is the proposed model which considering the temporal, social, collabotative and task correlations. We set $\gamma = 0.5$, $\beta_1 = 1$, $\beta_2 = 2$ and $\beta_3 = 1$.

For all models, dropout with ratio 0.2 is used. The models are trained with batch size 64 and early stop is performed by monitoring the validation error.



(a) Attributes: H = 1D (b) Relations: H = 1D
(c) Attributes: H = 2D (d) Relations: H = 2D
(e) Attributes: H = 3D (f) Relations: H = 3D

Figure 13: Influence of historical length.

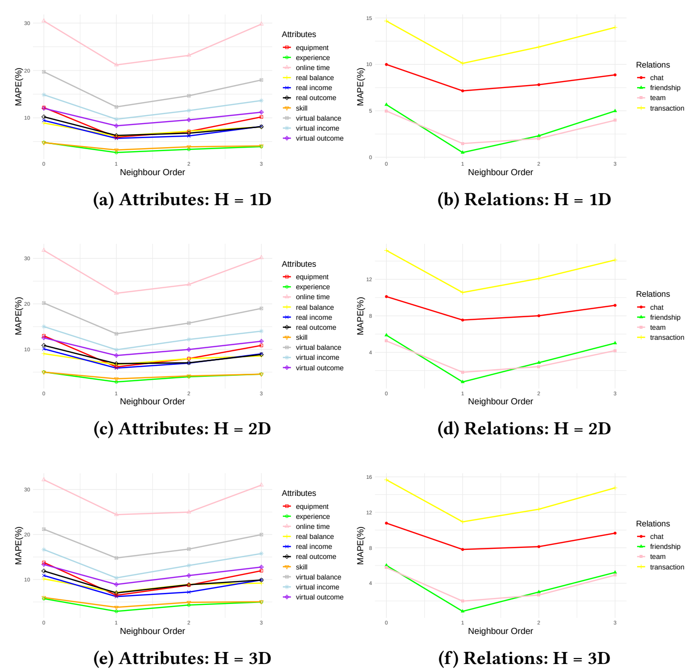
F ADDITIONAL EXPERIMENTAL RESULTS

F.1 Influence of Historical Length

In this section, we study how the historical length affects the performance. Figure 13 shows the prediction error of MAPE with respect to the historical length. We can see that when the length is 4 weeks, our method achieves the best performance. The decreasing trend in MAPE as the length increases shows the importance of considering the temporal correlation. Furthermore, as the length increases to more than 4 weeks, the performance slightly degrades but mainly remains stable. One potential reason is that when considering longer temporal correlation, more parameters need to be learned. As a result, the training becomes harder.

F.2 Influence of Neighbour Order

Our intuition is that applying MGAT to deal with social correlations in multi-social-temporal prediction tasks. We verified the intuition by varying the neighbour order for MGAT. As the 0-order means without considering social correlations. In Figure 14, we show the performance of our method with respect to the size of the surrounding neighbour order. We can see that when the size of neighbour order is 1 (1-hop enclosing subgraph), the method achieves the best performance. We also figure out that the method performs poorly when the size of neighbour order is 0, which shows the great importance of social correlations in mult-social-temporal prediction tasks. When we increase the neighbour order, we find that out method performs decreasingly, this may cause by model complexity as well as big graph memories.



(a) Attributes: H = 1D (b) Relations: H = 1D
(c) Attributes: H = 2D (d) Relations: H = 2D
(e) Attributes: H = 3D (f) Relations: H = 3D

Figure 14: Influence of neighbour order.

F.3 Application Details

We compare the GMTL with the baseline and the state-of-the-art methods in applications for social time series prediction and temporal link weight prediction. For social time series prediction, we use the precision|recall|f1-score as the evaluation metrics while for temporal link weight prediction, we use the click-through rate|conversion rate as the evaluation metrics.

Table 7: Applications for social time series prediction

Applications	Baseline	GMTL	State-of-the-art
Churn Prediction	85.18%|84.69%|84.93%	**91.27%|89.76%|90.51%**	91.27%|89.76%|90.51%
Return Prediction	44.15%|40.28%|42.13%	**52.89%|47.38%|49.98%**	52.89%|47.38%|49.98%
Farmer Detection	76.75%|64.45%|70.06%	81.69%|72.51%|76.82%	**89.76%|79.67%|84.41%**
Banker Detection	55.13%|41.72%|47.50%	70.53%|61.19%|65.53%	**78.04%|65.67%|71.32%**
Buyer Detection	22.88%|21.20%|22.01%	71.68%|64.37%|67.83%	**78.12%|70.05%|73.87%**
Recharge Prediction	70.12%|62.89%|66.31%	75.71%|67.79%|71.53%	**78.14%|70.27%|74.00%**
Consumption Prediction	71.12%|69.89%|70.50%	79.87%|76.93%|78.37%	**81.27%|78.88%|80.06%**
Experience Pursuit	86.17%|87.28%|86.72%	**91.87%|90.73%|91.30%**	91.87%|90.73%|91.30%
Equipment Pursuit	84.19%|86.21%|85.19%	**89.96%|88.17%|89.06%**	89.96%|88.17%|89.06%
Skill Pursuit	85.89%|82.78%|84.31%	**90.11%|87.27%|88.67%**	90.11%|87.27%|88.67%

Table 8: Applications for temporal link weight prediction

Applications	Baseline	GMTL	State-of-the-art
Abnormal Transaction Detection	51.34%|48.94%	64.18%|58.98%	**71.32%|62.77%**
Friendship Suggestion	53.19%|35.98%	71.87%|58.17%	**78.14%|65.23%**
Friends List Cleanup	54.94%|51.02%	64.56%|59.19%	**69.10%|63.44%**
Team Formation	75.56%|71.87%	**89.17%|81.18%**	89.17%|81.18%
Chat Fast Entry	78.98%|72.66%	**84.71%|79.14%**	84.71%|79.14%