Technical Appendix

1 Ablation Study

In order to understand the importance of each component in TDHGPN, we conduct a series of ablation experiments on different components of the model. We omit the Memetracker dataset, because this dataset does not have social network. In particular, we generate several variants of our model as follows:

- **TDHGPN-S** removes the social graph.
- **TDHGPN-D** removes the information diffusion graph.
- **TDHGPN-H** removes the heterogeneous graph.
- **TDHGPN-GPN** uses graph convolutional network instead of graph perception network.
- **TDHGPN-MB** removes the time-based mini-batch.
- **TDHGPN-T** removes the users' temporal features, the Query set in Transformer uses the users' structural features instead.
- **TDHGPN-TRM** removes Transformer and replaced it with RNN.
- **TDHGPN-F** removes the proposed residual fusion method in Transformer, and replaces it with the original residual connection method.

According to Table 1, we can find that when we remove the social graph, there will be a small drop in performance compared to TDHGPN. When the diffusion graph is removed, a similar phenomenon can also be discovered. The results show that both the social graph and the diffusion graph in TDHGPN are essential for information diffusion prediction.

When the heterogeneous graph is removed from the model, the performance further decays a lot compared with removing the social graph or diffusion graph. The phenomenon shows that both relations contain complementary information and combining them does help to improve the performance.

When the graph convolutional network is used to replace the graph perception network, there will be a small drop in performance compared with TDHGPN. This shows that our proposed graph perception network can learn more effective cascaded graph-level representations.

When the Time-Based Mini-Batch module is removed from the model, the performance of TDHGPN-MB in Douban dataset is slightly better than that of TDHGPN. It is because that removing the time-based mini-batch module will make the message forwarding time more accurate, resulting in slightly better performance. Although the use of time-based mini-batch module will reduce the performance by a small margin, it is still within an acceptable range. But it significantly speeds up the training speed of the model, which is shown in the next subsection.

When the users' temporal features are removed, the experimental effect is greatly reduced. The experimental results show that the temporal features play a crucial role in information diffusion prediction. Combining temporal features and structural features effectively, obtaining the spatial-temporal features of users has a positive effect on information diffusion prediction.

When the Transformer model is removed and replaced with a recurrent neural network (RNN), the experimental effect is greatly reduced, especially on the Twitter dataset. The experimental results show that the Transformer model is very effective in the field of information diffusion prediction, which shows that the multi-head attention inside the Transformer model can greatly improve the experimental effect.

When the novel proposed residual fusion method in the Transformer model is removed and replaced by the original residual connection method, the experimental effect is greatly reduced. The experimental results show that the new residual fusion method proposed by us is more effective than the original residual connection method, because the new residual fusion method adopts the fusion gate mechanism, which can selectively integrate the users' temporal features and structural features.

2 Parameter Analysis

In this section, we study how different values of hyperparameters affect the performance of TDHGPN on dataset Twitter.

2.1 Dimension size of user embedding *d*

The dimension size of user embedding d is selected from $\{16, 32, 64, 128, 256\}$, and the performance of its different values on the Twitter dataset is shown in Table 2. It can be seen from the data in the table that with the increase of dimension d, the experimental effect gradually becomes better, but when the dimension d exceeds 64, the experimental effect begins to decline. This is because as the dimension d of the users' features increase, the information contained in the users' features

also increase, and when it increases to a certain value, the redundant information contained in the users' features will lead to decrease in the experimental effect. Therefore, we set the dimension d of users' features to 64 in this paper.

2.2 Number of *step_len*

As shown in Fig. 1, we evaluate the effect of the parameter $step\_len$ in the time-based mini-batch module on DHGP-NTM. As $step\_len$ increases, the time of each epoch for training TDHGPN will gradually decrease. But when $step\_len$ is large than 5, the training efficiency is not further improved. As shown in Fig. 1, when the value of $step\_len$ is 5, TDHGPN have a good balance between efficiency and performance. Hence, we set the value of $step\_len$ to 5 by default in this paper. In this case, the time-based mini-batch module can improve the training efficiency of TDHGPN by nearly 3 times. When $step\_len$ is 2, the training efficiency of TDHGPN is almost same as that of DyHGCN, but TDHGPN is obviously better than TDHGPN in terms of hits@k and map@k, as shown in Table 1.

2.3 Time Interval *n*

The time interval n is selected from {1, 2, 4, 8, 16}, and the performance of its different values on the Twitter dataset is shown in Table 3. As the time interval n increases, the effect of the TDHGPN model gradually increases, but when the time interval n is greater than 8, Map@k and Hits@k do not further improve but decline. This is because as the time interval n increases, the users' temporal features are more accurate, and when it increases to a certain value, too many learned temporal features will easily lead to overfitting, but will reduce the experimental effect. Therefore, we set the time interval n to 8 in this paper.

The number of heads of multi-head attention *Head* The number of heads of multi-head attention is selected from {1, 2, 4, 8, 16}, and the performance of the selection of different values on the Twitter dataset is shown in Table 4. From the data in the table, it can be seen that with the increase of the number of heads of multi-head attention, the experimental effect is gradually getting better. When the $Head$ is equal to 8, the experimental effect is optimal, and when the $Head$ continues to increase, the effect is decreased. This is because as the number of heads of multi-head attention increases, more information of different spatial dimensions is learned, and when it increases to a certain value, too much redundant information learned will lead to a decrease in the experimental effect. Therefore, we set the number of heads of multi-head attention $Head$ to 8 in this paper.

2.4 The number of layers of GPNConv *L*

The number of layers of GPNConv is selected from {1, 2, 3}, and the performance of TDHGPN on Twitter dataset is shown in Table 5. As the GPNConv layers increase, the performance of TDHGPN gradually decreases on hits@k and map@k. Hence, we set the number of layers of GPNConv to 1 by default in this paper.

2.5 The number of decoder layers *N*

The number of layers N is selected from {1, 2, 3, 4, 5}, and the performance of the selection of different values on the Twitter dataset is shown in Table 6. It can be seen from the data in the table that when the number of layers N is equal to 2, the effect of the experiment is optimal, and with the increase of the number of layers N, the effect of the experiment decreases gradually. This is because when the number of layers of the neural network increases to a certain value, too much information will be learned, which will lead to overfitting, but will reduce the experimental effect. Therefore, we set the number of layers N of the decoder in Transformer to 2 in this paper.

Datasets	Model	hits@10	hits@50	hits@100	map@10	map@50	map@100
Twitter	TDHGPN	**29.25**	**48.37**	**59.70**	**17.71**	**18.59**	**18.75**
	TDHGPN-S	29.20	48.29	59.45	17.51	18.49	18.64
	TDHGPN-D	29.16	48.33	59.64	17.54	18.40	18.55
	TDHGPN-H	29.06	48.12	59.15	17.19	18.03	18.38
	TDHGPN-GPN	29.15	48.19	59.57	17.62	18.51	18.67
	TDHGPN-MB	29.06	48.09	59.37	17.47	18.34	18.50
	TDHGPN-T	26.23	44.42	55.64	16.20	17.02	17.17
	TDHGPN-TRM	24.79	42.89	54.17	14.31	15.11	15.27
	TDHGPN-F	26.10	44.31	55.43	14.99	15.82	15.98
Douban	TDHGPN	17.77	31.19	38.69	10.39	11.00	**11.11**
	TDHGPN-S	17.67	30.97	38.53	10.27	10.97	11.08
	TDHGPN-D	17.73	31.09	38.66	10.28	10.89	11.00
	TDHGPN-H	17.53	30.84	38.43	10.10	10.72	10.83
	TDHGPN-GPN	17.74	31.07	38.58	10.29	10.90	11.01
	TDHGPN-MB	**17.81**	**31.28**	**38.74**	**10.42**	**11.01**	11.09
	TDHGPN-T	14.98	27.54	34.80	7.93	8.49	8.60
	TDHGPN-TRM	16.88	29.22	36.58	9.98	10.54	10.64
	TDHGPN-F	15.97	28.18	35.88	9.46	10.01	10.12

Table 1: Ablation study on Twitter and Douban datasets.

d	hits@10	hits@50	hits@100	map@10	map@50	map@100
16	24.73	43.09	55.16	13.32	14.12	14.30
32	27.74	47.06	58.88	16.14	16.99	17.16
64	**29.25**	**48.37**	**59.70**	**17.71**	**18.59**	**18.75**
128	29.20	48.24	59.24	17.69	18.46	18.60
256	29.12	48.07	59.11	17.53	18.38	18.54

Table 2: The effect of dimension size of user embedding on Twitter dataset.

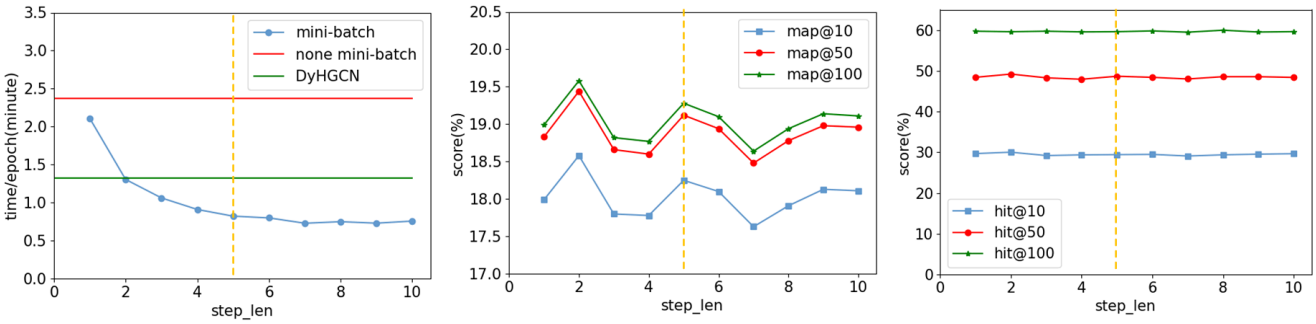

Figure 1: The effect of $step\_len$ on Twitter dataset.

n	hits@10	hits@50	hits@100	map@10	map@50	map@100
1	29.22	48.12	59.46	17.64	18.49	18.65
2	28.97	48.17	59.40	17.66	18.52	18.68
4	29.18	**48.37**	59.68	**17.71**	18.55	18.71
8	**29.25**	**48.37**	**59.70**	**17.71**	**18.59**	**18.75**
16	29.19	48.28	59.64	17.66	18.51	18.67

Table 3: The effect of time interval on Twitter dataset.

$Head$	hits@10	hits@50	hits@100	map@10	map@50	map@100
1	28.85	48.07	59.06	17.61	18.45	18.60
2	28.95	48.20	59.32	17.56	18.44	18.60
4	29.21	47.88	59.54	17.53	18.36	18.52
8	**29.25**	**48.37**	**59.70**	**17.71**	**18.59**	**18.75**
16	29.17	48.20	59.50	17.69	18.55	18.70

Table 4: The effect of heads of multi-head attention on Twitter dataset.

L	hits@10	hits@50	hits@100	map@10	map@50	map@100
1	**29.25**	**48.37**	**59.70**	**17.71**	**18.59**	**18.75**
2	29.05	48.00	59.33	17.21	18.16	18.23
3	28.51	46.35	57.68	16.53	17.39	17.70

Table 5: The effect of GPNConv layers on Twitter dataset.

N	hits@10	hits@50	hits@100	map@10	map@50	map@100
1	28.74	47.97	59.35	17.34	18.19	18.35
2	**29.25**	**48.37**	**59.70**	**17.71**	**18.59**	**18.75**
3	28.58	47.36	59.02	17.24	18.08	18.25
4	28.39	47.66	58.87	16.95	17.82	17.98
5	28.67	47.34	58.52	17.06	17.89	18.04

Table 6: The effect of the number of decoder layers on Twitter dataset.