A Dynamic Heterogeneous Network with Time-Based Mini-Batch for Information Diffusion Prediction

Wei Fan, Meng Liu, and Yong Liu(✉)

Heilongjiang University, Harbin, China
{2201792,2191438}@s.hlju.edu.cn, liuyong123456@hlju.edu.cn

Github:https://github.com/DHGPNTM/DHGPNTM
DASFAA 2022

Reported by Nengqiang Xiang

Introduction

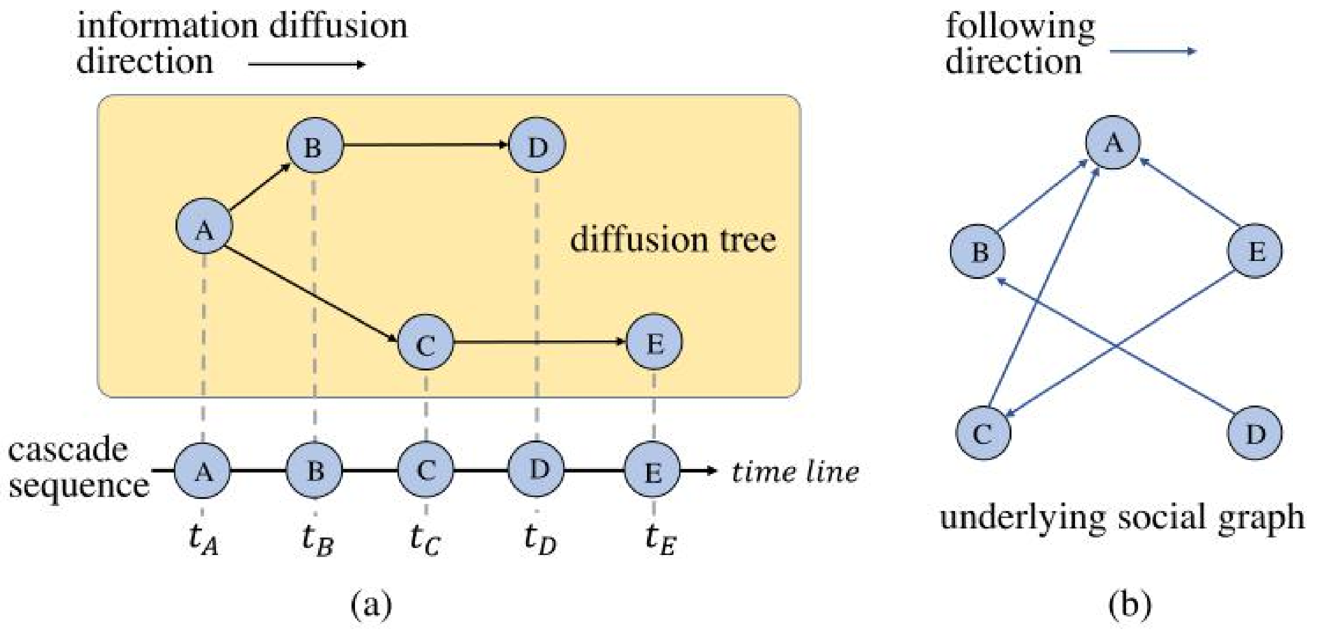
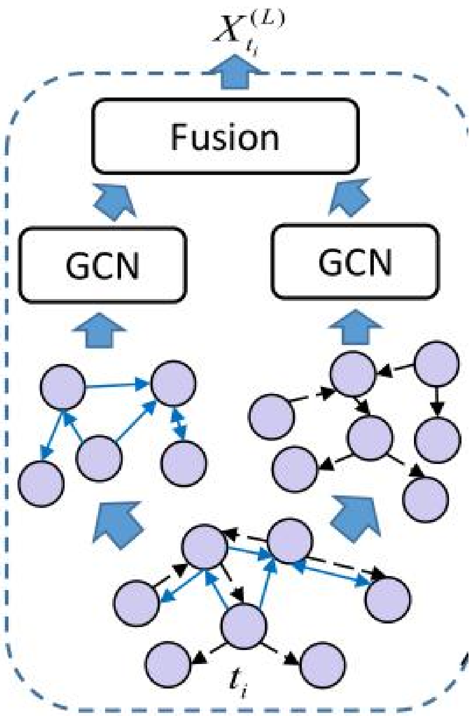


Fig. 1. An example of an information cascade sequence, diffusion tree and social graph.

Existing approaches either focus only on the use of diffusion paths or use social networks, without considering both factors together. And the existing GNN still suffers from excessive smoothness.

In this paper, we propose the Dynamic Heterogeneous Graph Perception Network with Time-Based Mini-Batch (DHGPNTM) for Information Diffusion Prediction

Method



Dynamic Graph Encoding

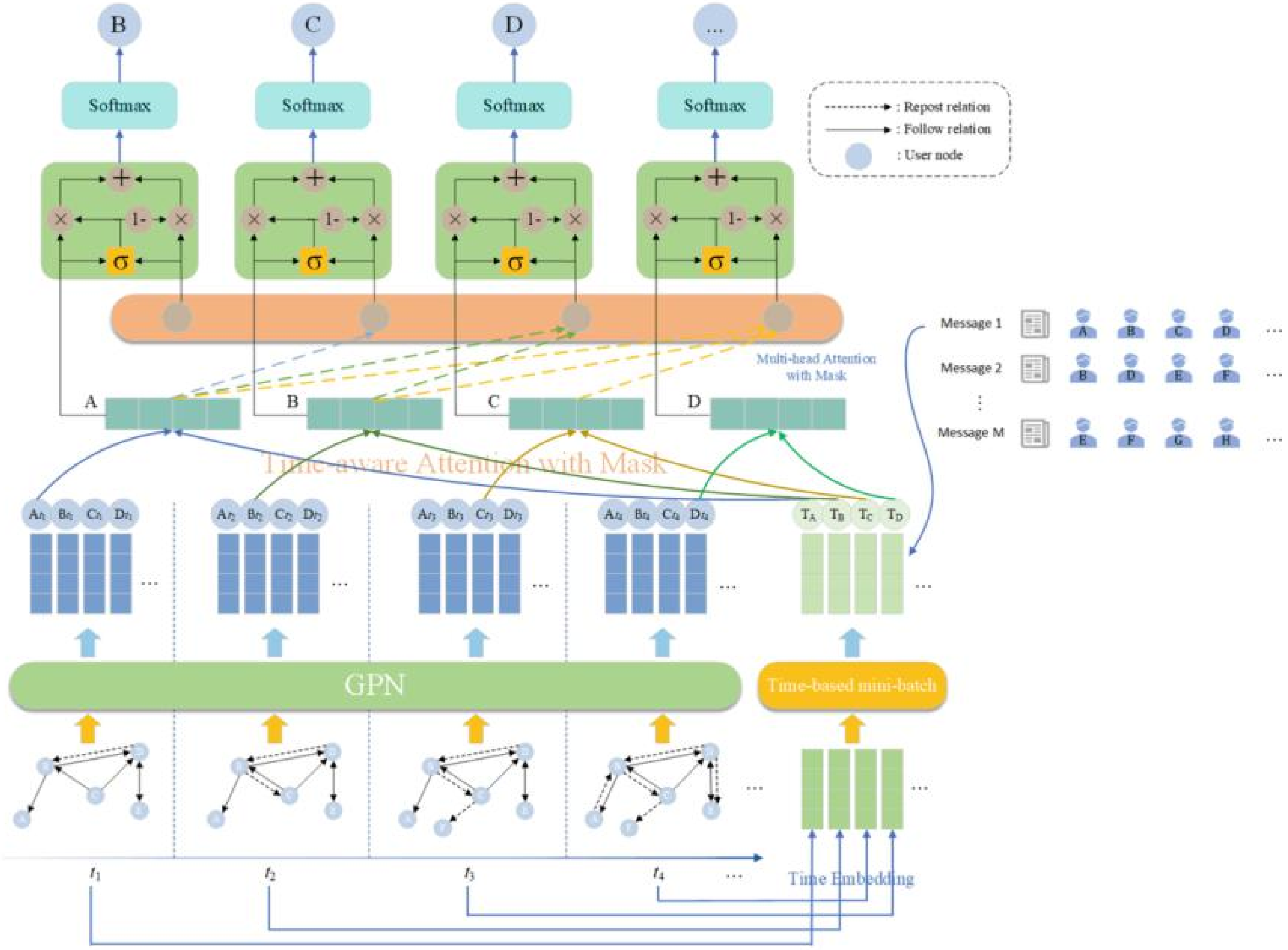


Fig. 1. The architecture of dynamic heterogeneous graph perception network with time-based mini-batch.

Method

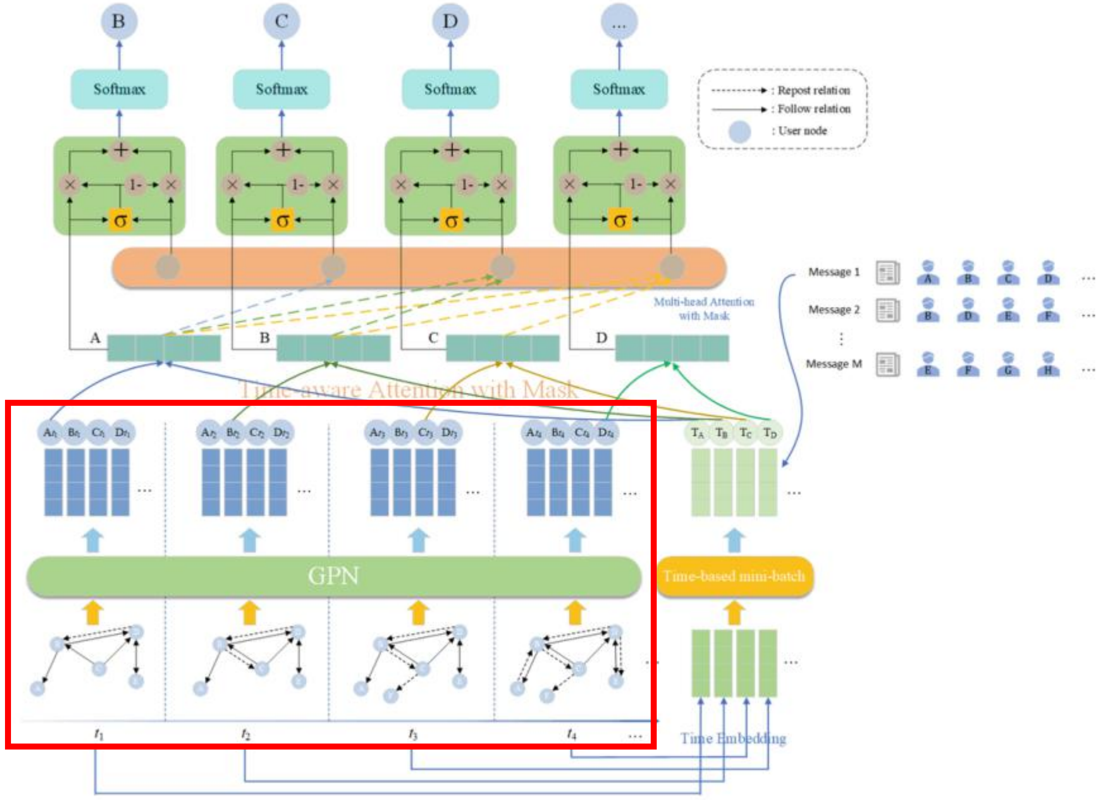


Fig. 1. The architecture of dynamic heterogeneous graph perception network with time-based mini-batch.

Heterogeneous Graph Construction

$$G_H^i = (V, E_H^i)\ E_H^i = E \cup E_M^i$$

$$G_H = \{G_H^1, G_H^2, ..., G_H^n\}$$

Graph Perception Network

$$e_{uv} = ReLU((f_{MLP}^l(h_v^{l-1}))^{\mathrm{T}} W_a^l f_{MLP}^l(h_u^{l-1})), \tag{1}$$

$$\alpha_{uv} = Softmax(e_{uv}) = \frac{exp(e_{uv})}{\sum_{k \in N(v) \cup \{v\}} exp(e_{kv})}, \tag{2}$$

$$h_v^l = ReLU(\sum_{u \in N(v) \cup \{v\}} \alpha_{uv} \cdot f_{MLP}^l(h_u^{l-1})), \tag{3}$$

$$h_v = ReLU(f_P(h_v^0) + f_P(h_v^L)), \tag{4}$$

Method

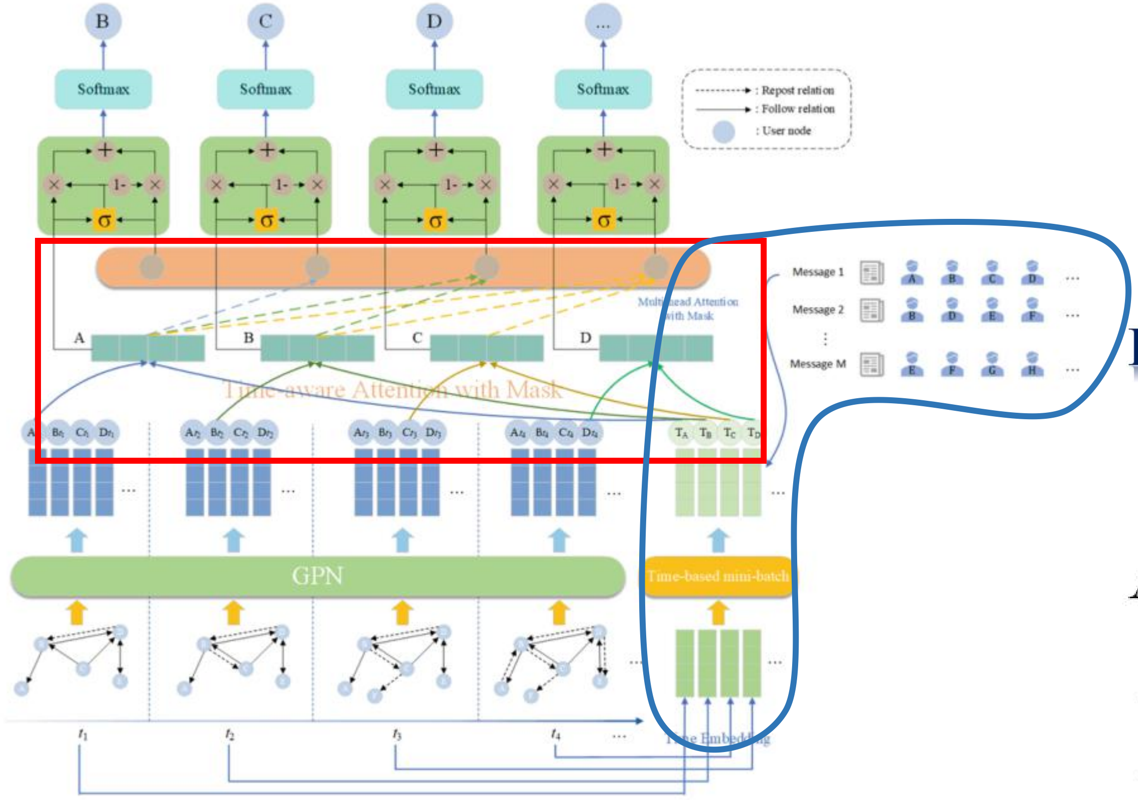


Fig. 1. The architecture of dynamic heterogeneous graph perception network with time-based mini-batch.

User Dynamic Preferences Based on Mini-Batch

$$\widetilde{v}_{ij} = Softmax(\frac{h_j^{\mathrm{T}} t'}{\sqrt{d}}) \cdot h_j, \quad t' = Lookup(t_{ij}), \tag{5}$$

Dependency-Aware User Embedding

$$\begin{aligned} \widetilde{V} &= \{\widetilde{v}_{11}, \widetilde{v}_{12}, ..., \widetilde{v}_{MN_c}\} \in R^{M \times N_c \times d} \\ Attention(Q, K, S) &= Softmax(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})S, \\ h_i &= Attention(\widetilde{V}W_i^Q, \widetilde{V}W_i^K, \widetilde{V}W_i^S), \\ Z &= [h_1 : h_2 : ... : h_{Head}]W^O, \end{aligned} \tag{6}$$

Method

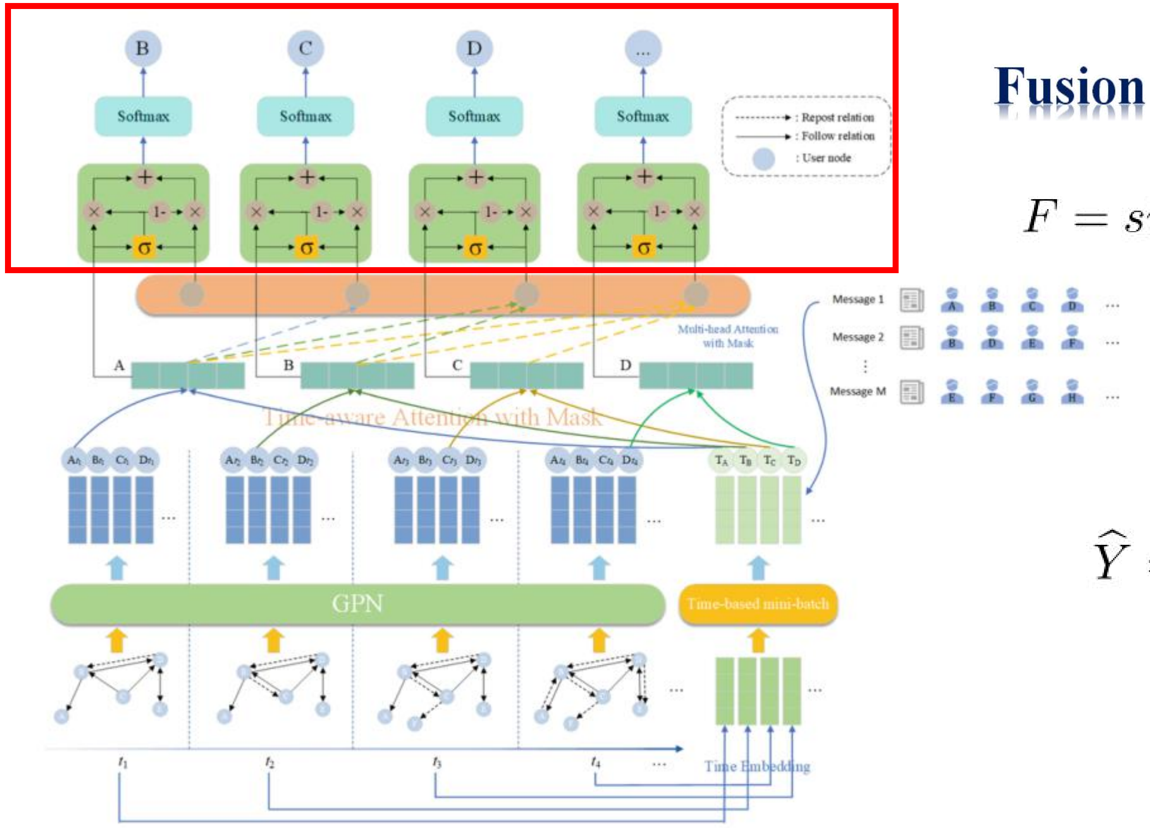


Fig. 1. The architecture of dynamic heterogeneous graph perception network with time-based mini-batch.

Fusion Gate

$$F = sigmoid(\widetilde{V}W_f^1 + ZW_f^2 + b_f), A = F \odot \widetilde{V} + (1 - F) \odot Z, \quad (7)$$

Predicting & Loss

$$\widehat{Y} = Softmax(W_2(Relu(W_1 A^{\mathrm{T}} + b_1))^{\mathrm{T}} + b_2), \quad (8)$$

$$\mathcal{L}(\theta) = -\sum_{i=1}^{M}\sum_{j=2}^{N_c}\sum_{k=1}^{|V|} y_{ijk}\log(\widehat{y}_{ijk}), \quad (9)$$

Experiments

Table 1. Statistics of the Twitter, Douban, and Memetracker datasets.

Datasets	Twitter	Douban	Memetracker
# Users	12,627	23,123	4,709
# Links	309,631	348,280	-
# Cascades	3,442	10,602	12,661
Avg. Length	32.60	27.14	16.24

Experiments

Datasets	Model	hits@10	hits@50	hits@100	map@10	map@50	map@100
Twitter	DeepDiffuse	4.57	8.80	13.39	3.62	3.79	3.85
	TopoLSTM	6.51	15.48	23.68	4.31	4.67	4.79
	NDM	21.52	32.23	38.31	14.30	14.80	14.89
	SNIDSA	23.37	35.46	43.49	14.84	15.40	15.51
	FOREST	26.18	40.95	50.39	17.21	17.88	18.02
	DyHGCN	28.10	47.17	58.16	16.86	17.73	17.89
	DHGPNTM	**29.68**	**48.65**	**59.86**	**18.13**	**18.99**	**19.15**
Douban	DeepDiffuse	9.02	14.93	19.13	4.80	5.07	5.13
	TopoLSTM	9.16	14.94	18.93	5.00	5.26	5.32
	NDM	10.31	18.87	24.02	5.54	5.93	6.00
	SNIDSA	11.81	21.91	28.37	6.36	6.81	6.91
	FOREST	14.16	24.79	31.25	7.89	8.38	8.47
	DyHGCN	15.92	28.53	36.05	8.56	9.12	9.23
	DHGPNTM	**17.86**	**31.32**	**38.87**	**10.37**	**10.98**	**11.09**
Memetracker	DeepDiffuse	13.93	26.50	34.77	8.14	8.69	8.80
	NDM	25.44	42.19	51.14	13.57	14.33	14.46
	FOREST	29.43	47.41	56.77	16.37	17.21	17.34
	DyHGCN	29.74	48.45	58.39	16.48	17.33	17.48
	DHGPNTM	**30.70**	**50.48**	**60.63**	**18.01**	**18.92**	**19.06**

Thanks