Improving Information Cascade Modeling by Social Topology and Dual Role User Dependency

Baichuan Liu[§] [0000-0002-6216-3011], Deqing Yang[§⋆] [0000-0002-1390-3861], Yuchen Shi[‡], and Yueyi Wang[‡]

School of Data Science, Fudan University, Shanghai 200433, China.
[§] {bcliu20,yangdeqing}@fudan.edu.cn, [‡] {ycshi21,yueyiwang21}@m.fudan.edu.cn

Github:https://github.com/JUNJIN0126/TAN-DRUD
DASFAA 2022

Reported by Nengqiang Xiang

Introduction

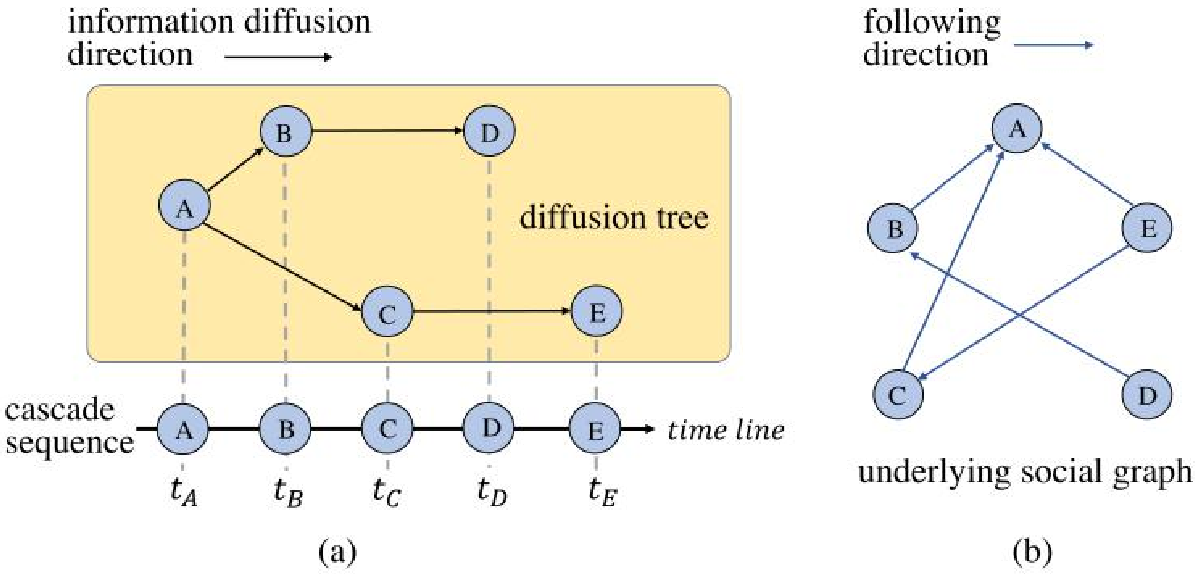
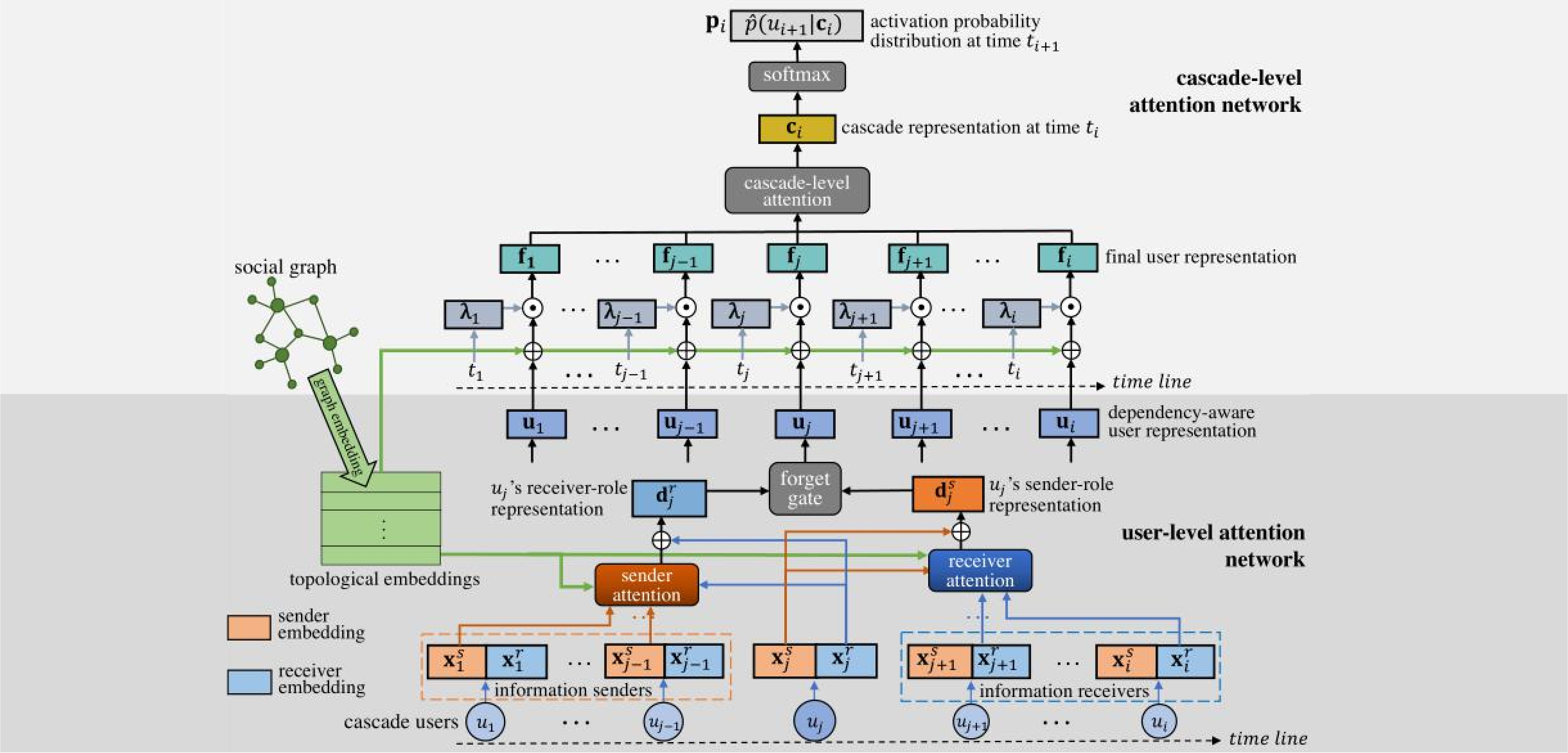


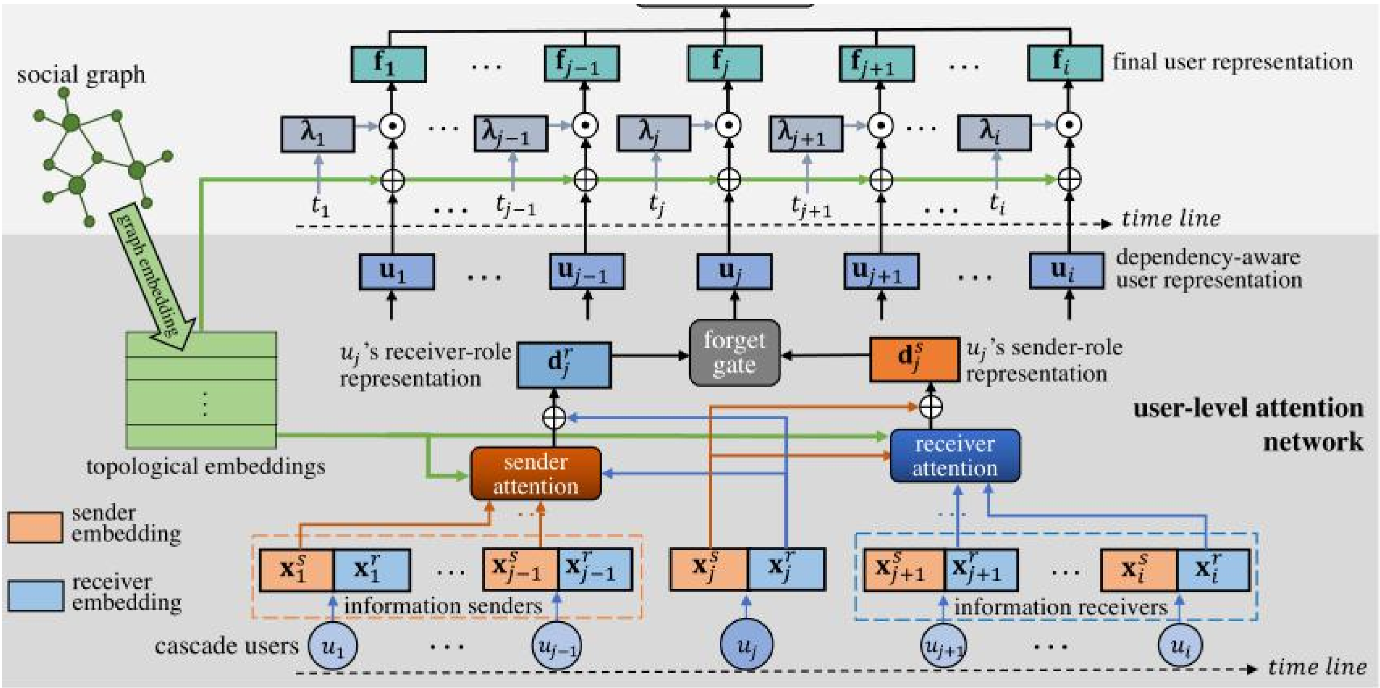
Fig. 1. An example of an information cascade sequence, diffusion tree and social graph.

The user dependencies captured by most of previous diffusion models are only unidirectional. They suppose a successor is only influenced by a predecessor during information diffusion, whereas the opposite dependency is rarely considered

Method


$\mathbf{p}_i$ $\hat{p}(u_{i+1}|\mathbf{c}_i)$ activation probability distribution at time t_{i+1}
softmax
cascade-level attention network
$\mathbf{c}_i$ cascade representation at time t_i
cascade-level attention
$\mathbf{f}_1$ ... $\mathbf{f}_{j-1}$ $\mathbf{f}_j$ $\mathbf{f}_{j+1}$ ... $\mathbf{f}_i$ final user representation
social graph
λ_1 ... λ_{j-1} λ_j λ_{j+1} ... λ_i
t_1 ... t_{j-1} t_j t_{j+1} ... t_i time line
graph embedding
$\mathbf{u}_1$ ... $\mathbf{u}_{j-1}$ $\mathbf{u}_j$ $\mathbf{u}_{j+1}$ ... $\mathbf{u}_i$ dependency-aware user representation
u_j's receiver-role representation $\mathbf{d}_j^r$
forget gate
$\mathbf{d}_j^s$ u_j's sender-role representation
user-level attention network
topological embeddings
sender attention
receiver attention
sender embedding
receiver embedding
$\mathbf{x}_1^s$ $\mathbf{x}_1^r$ ... $\mathbf{x}_{j-1}^s$ $\mathbf{x}_{j-1}^r$ $\mathbf{x}_j^s$ $\mathbf{x}_j^r$ $\mathbf{x}_{j+1}^s$ $\mathbf{x}_{j+1}^r$ ... $\mathbf{x}_i^s$ $\mathbf{x}_i^r$
information senders
information receivers
cascade users u_1 ... u_{j-1} u_j u_{j+1} ... u_i time line

Method



User-level Attention Network

$$\alpha'^{s}_{kj} = \frac{\exp\left(\langle \mathbf{W}^{o}_{s}\mathbf{x}^{s}_{k}, \mathbf{W}^{c}_{r}\mathbf{x}^{r}_{j}\rangle\right)}{\sum_{l=1}^{j-1}\exp\left(\langle \mathbf{W}^{o}_{s}\mathbf{x}^{s}_{l}, \mathbf{W}^{c}_{r}\mathbf{x}^{r}_{j}\rangle\right)}, \quad (1)$$

$$\mathbf{E} = \frac{\mathbf{G}_c\mathbf{G}_c^{\top}}{||\mathbf{G}_c|| \times ||\mathbf{G}_c^{\top}||}, \quad (2)$$

$$\alpha^{s}_{kj} = \frac{\exp\left(\alpha'^{s}_{kj}e_{kj}\right)}{\sum_{l=1}^{j-1}\exp\left(\alpha'^{s}_{lj}e_{lj}\right)}. \quad (3)$$

$$\mathbf{d}'^{r}_{j} = \sum_{k=1}^{j-1}\alpha^{s}_{kj}\mathbf{x}^{s}_{k}, \quad \mathbf{d}^{r}_{j} = \mathbf{d}'^{r}_{j} + \mathbf{x}^{r}_{j}. \quad (4)$$

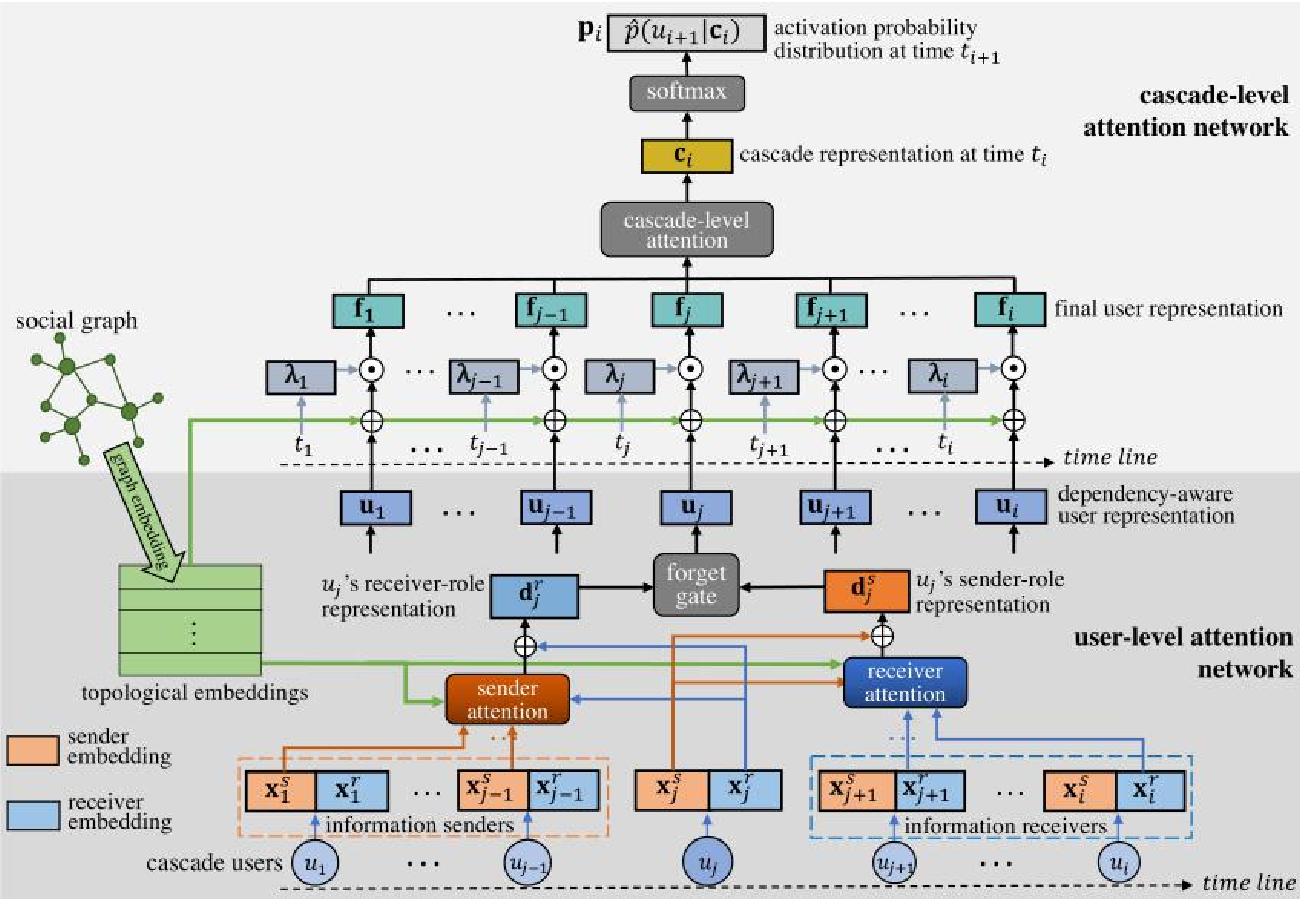
Sender and Receiver Embeddings

$$\mathbf{X}^{s}, \mathbf{X}^{r} \in \mathbf{R}^{N\times d}$$

Social Topological Embeddings

$$\mathbf{G}_c \in \mathbf{R}^{i\times d_g}$$

Method



$$\alpha'^{r}_{kj} = \frac{\exp\left(\langle \mathbf{W}^{o}_{r}\mathbf{x}^{r}_{k}, \mathbf{W}^{c}_{s}\mathbf{x}^{s}_{j}\rangle\right)}{\sum_{l=j+1}^{i} \exp\left(\langle \mathbf{W}^{o}_{r}\mathbf{x}^{r}_{l}, \mathbf{W}^{c}_{s}\mathbf{x}^{s}_{j}\rangle\right)}, \tag{5}$$

$$\alpha^{r}_{kj} = \frac{\exp\left(\alpha'^{r}_{kj} e_{kj}\right)}{\sum_{l=j+1}^{i} \exp\left(\alpha'^{r}_{lj} e_{lj}\right)}, \tag{6}$$

$$\mathbf{d}'^{s}_{j} = \sum_{k=j+1}^{i} \alpha^{r}_{kj}\mathbf{x}^{r}_{k}, \quad \mathbf{d}^{s}_{j} = \mathbf{d}'^{s}_{j} + \mathbf{x}^{s}_{j}. \tag{7}$$

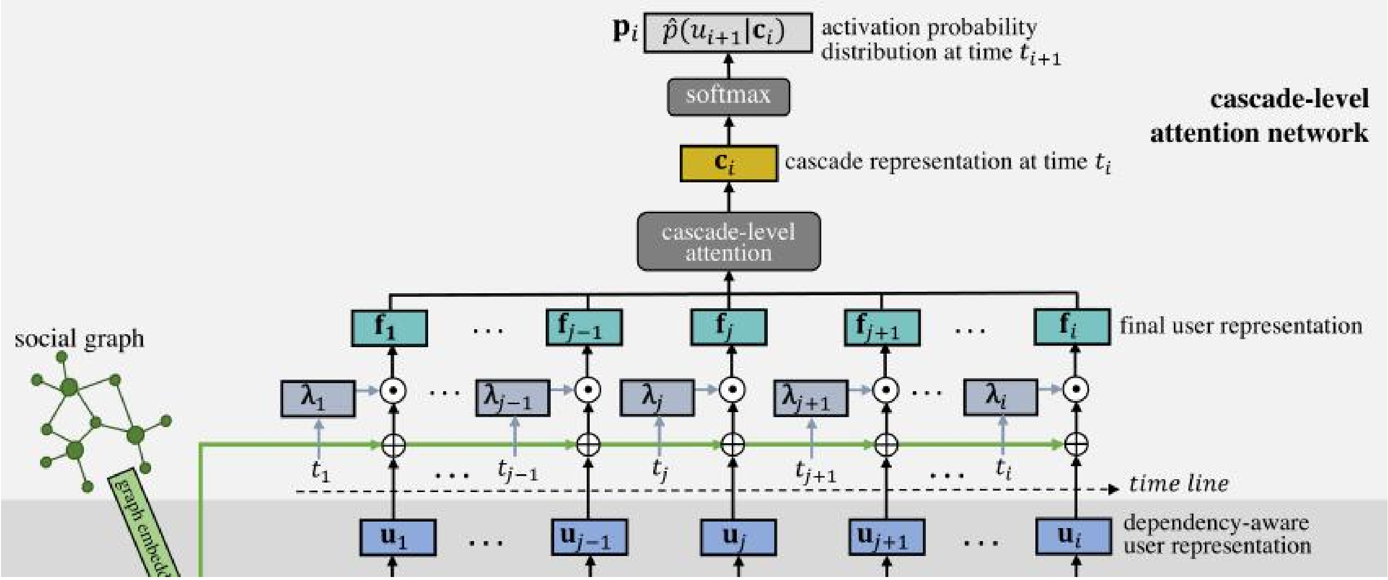
$$\mathbf{m} = \sigma(\mathbf{W}^{s}_{m}\mathbf{d}^{s}_{j} + \mathbf{W}^{r}_{m}\mathbf{d}^{r}_{j} + \mathbf{b}_{m}), \tag{8}$$

$$\mathbf{n} = \sigma(\mathbf{W}^{s}_{n}\mathbf{d}^{s}_{j} + \mathbf{W}^{r}_{n}\mathbf{d}^{r}_{j} + \mathbf{b}_{n}), \tag{9}$$

$$\mathbf{u}_{j} = (\mathbf{1} - \mathbf{m}) \odot \mathbf{d}^{s}_{j} + (\mathbf{1} - \mathbf{n}) \odot \mathbf{d}^{r}_{j} \tag{10}$$

Method

Cascade-level Attention Network



$$\mathbf{G}_{new} = \tanh\left(\mathbf{W}_g \mathbf{G}_c + \mathbf{b}_g\right) \tag{11}$$

$$\Delta^t = T_{\max}/T \quad \Delta t_j = t_i - t_j$$

$$\text{if } \; n = int(\Delta t_j / \Delta^t) \quad \mathrm{t}_j^i = 1 \qquad n \in [0, T)$$

$$\boldsymbol{\lambda}_j = \sigma(\mathbf{W}_t \mathbf{t}_j + \mathbf{b}_t) \quad \mathbf{t}_j \in \mathbb{R}^T \tag{12}$$

$$\mathbf{f}_j = \boldsymbol{\lambda}_j \odot (\mathbf{g}_j + \mathbf{u}_j) \tag{13}$$

$$\beta_j = \frac{\exp\left(\langle \mathbf{w}, \mathbf{f}_j \rangle\right)}{\sum_{k=1}^{i} \exp\left(\langle \mathbf{w}, \mathbf{f}_k \rangle\right)} \tag{14}$$

$$\mathbf{c}_i = \sum_{j=1}^{i} \beta_j \mathbf{f}_j. \tag{15}$$

Prediction and Optimization

$$\mathbf{p}_i = \operatorname{softmax}(\mathbf{W}_c \mathbf{c}_i + \mathbf{b}_c) \tag{16}$$

$$\mathcal{L} = -\frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{n_m - 1} \log \hat{p}(u_{i+1} \mid \mathbf{c}_i^m) + \lambda L_2 \tag{17}$$

Experiments

Table 1. Statistics of the three used datasets.

Dataset	Twitter	Douban	Meme
User number	12,627	23,123	4,709
Cascade number	3,442	10,602	12,661
Average cascade length	32.60	27.14	16.24
Social link number	309,631	348,280	–

Experiments

Table 2. TAN-DRUD's prediction performance (score %) with dual role embedding sizes (d).

d	RR	P@10	P@50	P@100
16	14.31	24.33	43.56	53.50
32	15.53	26.30	45.12	54.58
64	**16.62**	**28.13**	**45.61**	**55.43**
128	15.82	27.53	45.17	54.62

Table 3. TAN-DRUD's prediction performance (score %) with different time interval numbers (T).

T	RR	P@10	P@50	P@100
1	15.95	27.12	45.53	**55.91**
10	15.84	27.26	45.06	55.04
50	**16.62**	**28.13**	45.61	55.43
100	16.32	27.52	**46.04**	55.36

Experiments

Table 4. Prediction performance (score %) of all compared models for the three datasets. The best performance scores among all compared models are indicated in bold. The performance scores of leading baseline are underlined.

Model \ Dataset	Twitter				Douban				Meme			
	RR	P@10	P50	P100	RR	P@10	P@50	P@100	RR	P@10	P@50	P@100
DeepDiffuse	2.21	4.45	14.35	21.61	3.23	9.02	14.93	19.13	6.48	13.45	30.10	41.31
Bi-LSTM	7.12	13.41	26.71	36.06	7.95	15.97	29.89	37.41	12.32	24.73	46.27	56.33
Topo-LSTM	4.56	10.17	21.37	29.29	3.87	8.24	16.61	23.09	–	–	–	–
SNIDSA	–	23.37	35.46	43.39	–	11.81	21.91	28.37	–	–	–	–
FOREST	**17.49**	24.63	37.73	46.20	8.19	13.58	23.47	29.69	**16.76**	28.49	45.85	55.19
HiDAN	12.99	22.45	35.51	43.01	8.78	17.40	32.37	40.49	15.31	29.03	50.01	60.07
AN-DRUD	13.54	23.28	36.90	45.28	8.91	17.72	32.73	41.01	16.32	**29.48**	**51.09**	**61.33**
TAN-DRUD	16.62	28.13	45.61	55.43	**9.41**	**18.21**	**34.26**	**42.02**	–	–	–	–
improv. rate%	-4.97	14.21	20.89	19.98	7.18	4.66	5.84	3.78	-2.63	1.56	2.16	2.10

Experiments

Table 5. TAN-DRUD's prediction performance (score %) upon Twitter with different graph embedding models.

Model	RR	P@10	P@50	P@100
SDNE	16.27	26.68	41.96	51.58
LINE	15.50	26.71	43.66	53.12
DeepWalk	16.55	27.84	**46.06**	55.25
Node2Vec	**16.62**	**28.13**	45.60	**55.43**
SCE	13.59	24.20	40.54	50.75

Experiments

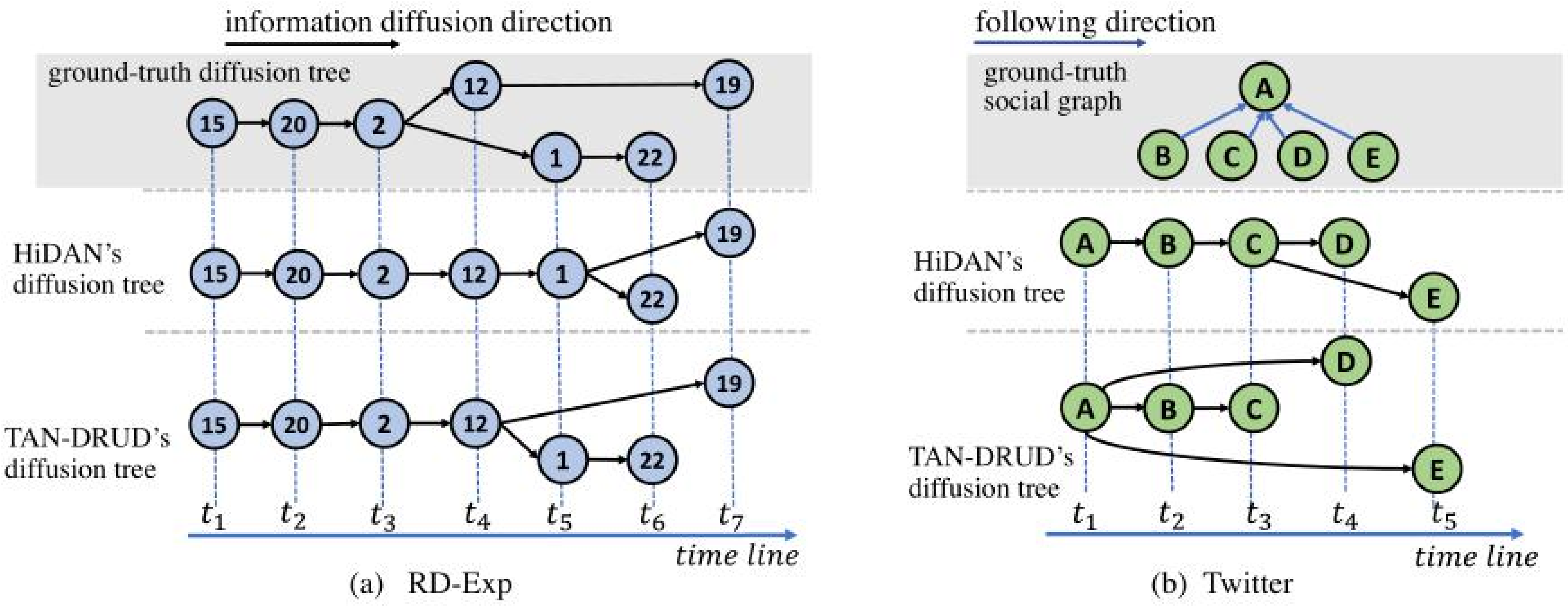


Fig. 3. Case study of diffusion tree inference.

Thanks