Continuous-Time Graph Learning for Cascade Popularity Prediction

Xiaodong Lu , Shuo Ji , Le Yu , Leilei Sun , Bowen Du , Tongyu Zhu*
SKLSDE Lab, Beihang University, Beijing 100191, China
{xiaodonglu, jishuo, yule, leileisun, dubowen, zhutongyu}@buaa.edu.cn

https://github.com/lxd99/CTCP

—— *IJCAI 2023*

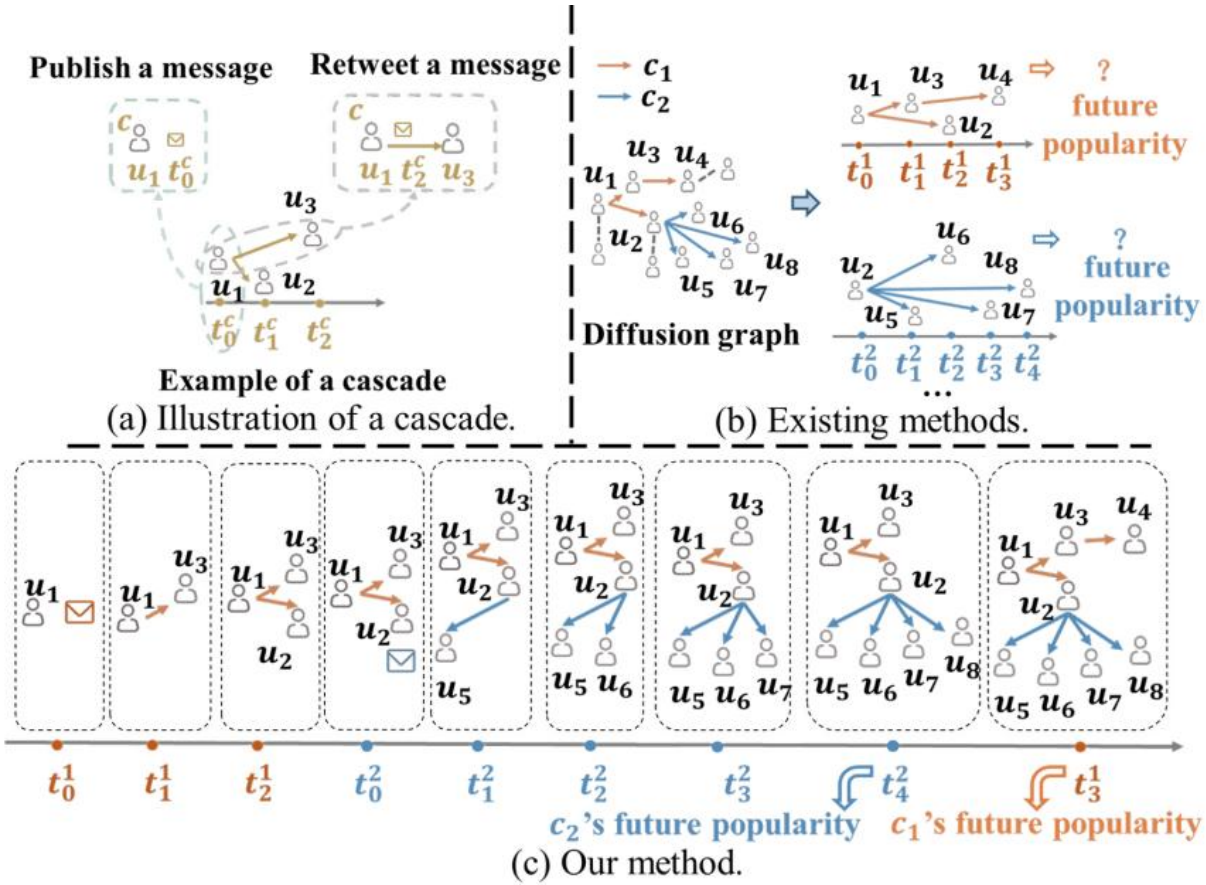
Reported by Yuyang Lai

1.Introduction

2.Method

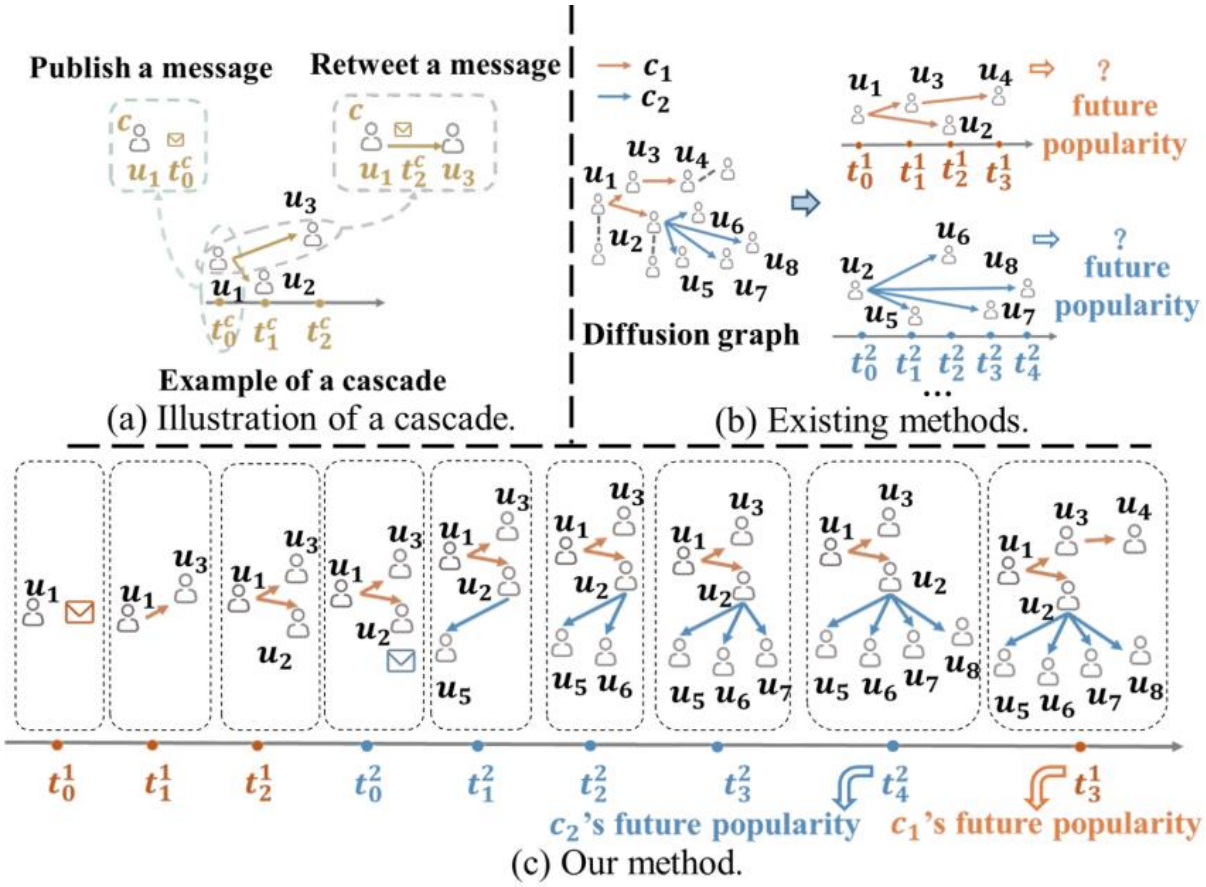
3.Experiments

Introduction



- Firstly, different cascades can be correlated with each other because of **the shared users or similar semantics**.
- Secondly, **the states of users are often evolving in a continuous manner** (e.g., a user is likely to gradually change his interests according to the information he/she received from the social network at different times), which cannot be captured by existing methods either.

Introduction



- we first combine all cascades into a **dynamic diffusion graph** as shown in Figure 1 (c), which can be considered as a universal sequence of diffusion behaviors (i.e, user-cascade and user-user interactions)

$$\mathcal{G}_d^t = \{(u_i, v_i, c_i, t_i) | t_i < t\}$$

- Then, we propose **an evolution learning module** to chronologically learn on each diffusion behavior by maintaining a dynamic representation for each user and cascade that evolves continuously as the diffusion behavior happens.

- Next,**a cascade representation learning model** is proposed to generate the static and dynamic cascade embeddings by aggregating user representations from both temporal and structural perspectives.

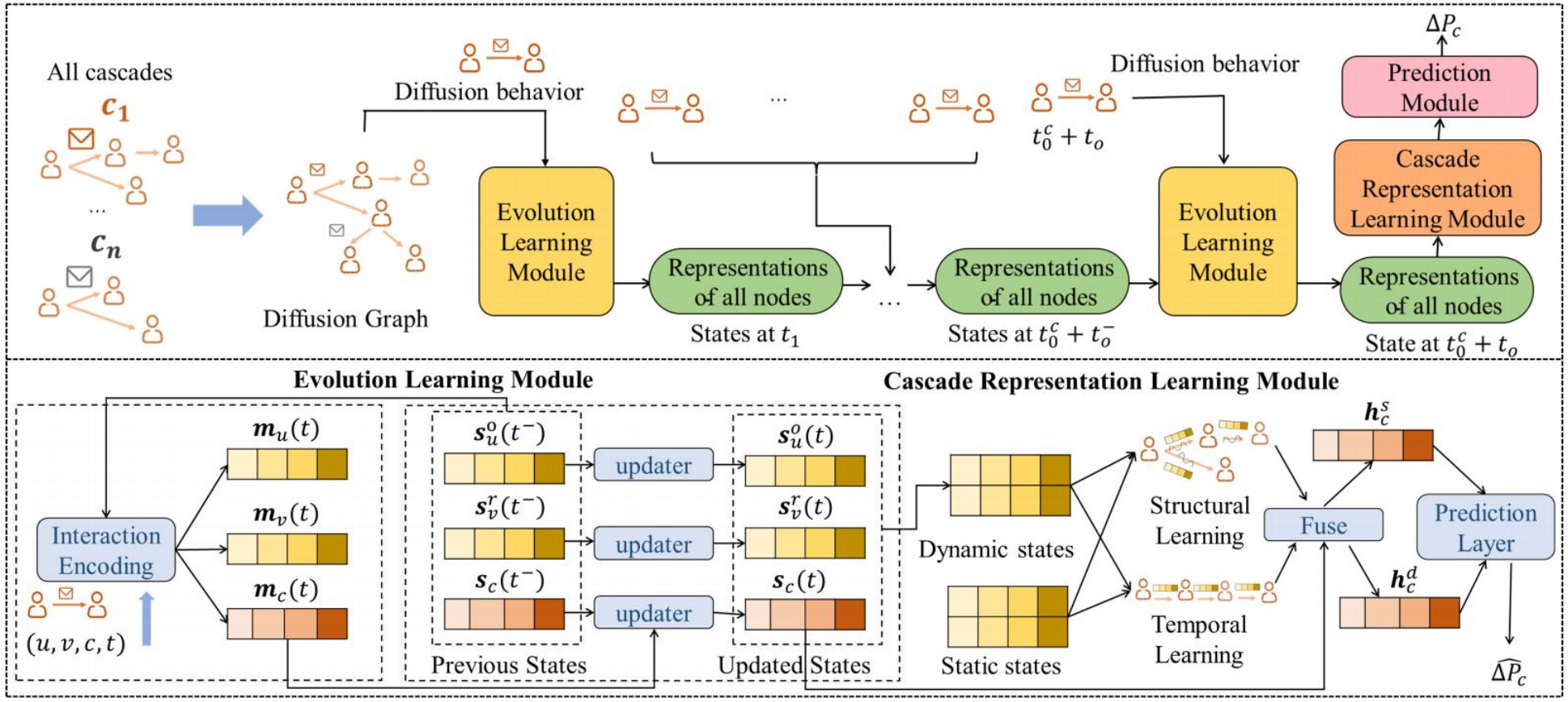
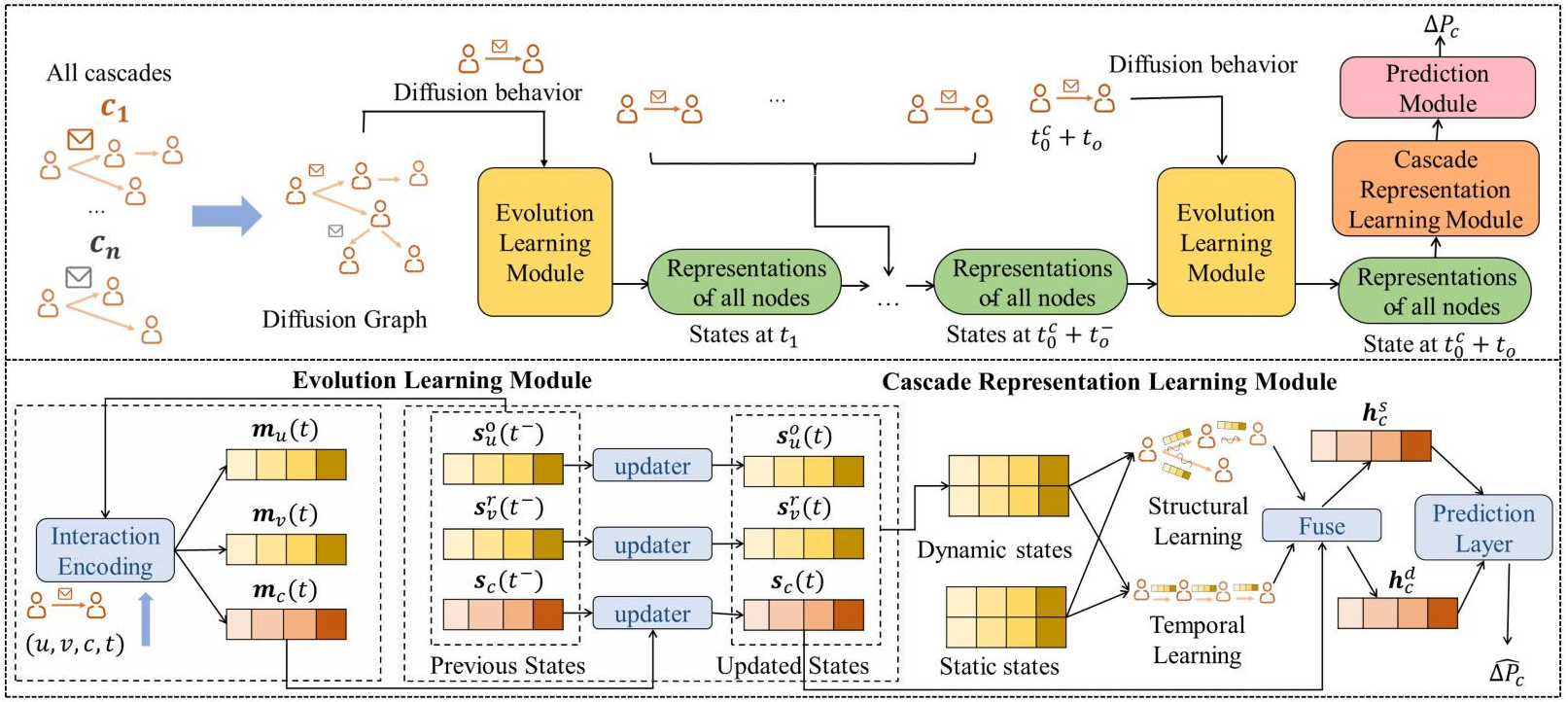


Figure 2: Framework of the proposed method. It consists of an evolution learning module to model the dynamics of user preferences and the correlation between cascades, a cascade representation learning module to capture the temporal and structural information within a cascade, and a prediction module to give future popularity. The popularity of cascade is predicted according to the most recent representations.

Method



Cascade

$$\mathcal{U} \quad c$$

$$g^c(t) = \{(u_i^c, v_i^c, t_i^c)\}_{i=1,\ldots,|g^c(t)|}$$

$$(u_i^c, v_i^c, t_i^c) \qquad (u_0^c, t_0^c)$$

Diffusion Graph

$$\mathcal{G}_d^t = \{(u_i, v_i, c_i, t_i) | t_i < t\}$$

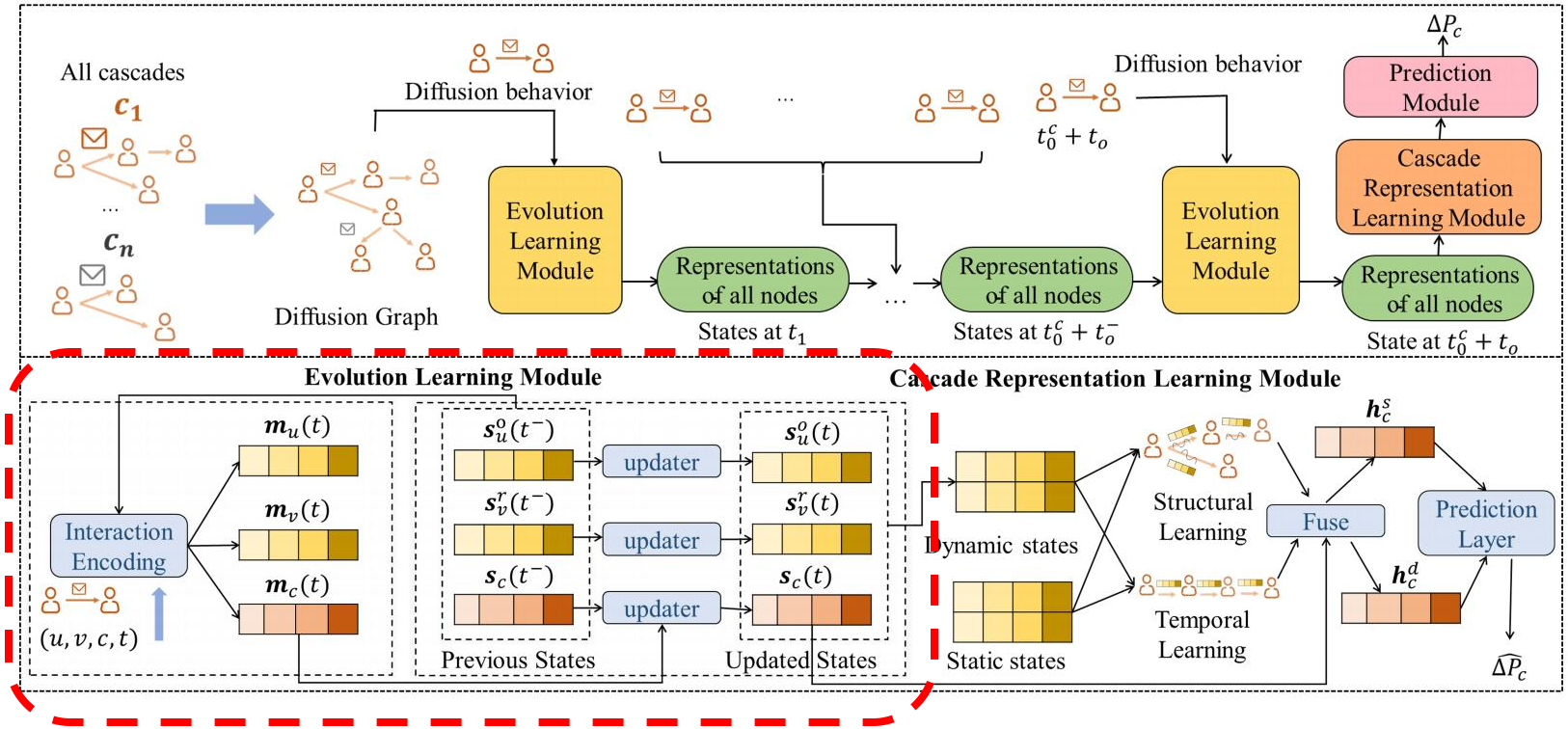
Cascade Prediction

$$\Delta P_c = |g^c(t_0^c + t_p)| - |g^c(t_0^c + t_o)| \qquad t_p >> t_o$$

$$f : g^c(t_0^c + t_o) \rightarrow \Delta P_c$$

$$f : g^c(t_0^c + t_o) \times G_d^{t_0^c + t_o} \rightarrow \Delta P_c$$

Method



Dynamic States

$$(u, v, c, t)$$

$$s_u^o(t) \qquad s_u^r(t) \qquad s_c(t)$$

Dynamic State Learning

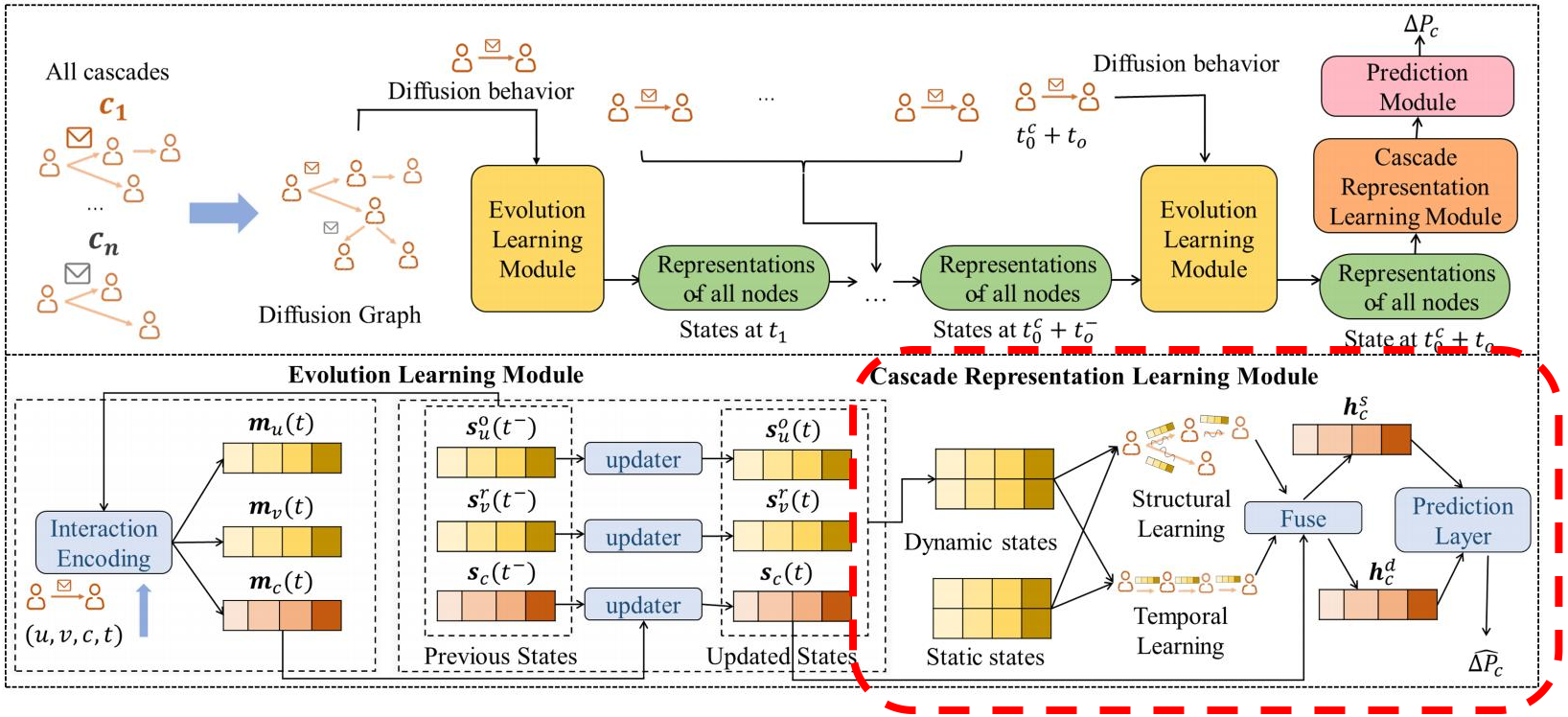
$$\boldsymbol{f}_u^t = [\cos w_1^r \Delta t_u, \cos w_2^r \Delta t_u, ..., \cos w_n^r \Delta t_u], \tag{1}$$

$$\Delta t_u = t - t_u^-$$

$$\boldsymbol{m}_u(t) = \sigma(\boldsymbol{W}^r[\boldsymbol{s}_u^o(t^-)||\boldsymbol{s}_v^r(t^-)||\boldsymbol{s}_c(t^-)||\boldsymbol{f}_u^t] + \boldsymbol{b}^r), \tag{2}$$

$$\begin{aligned}
\boldsymbol{g}_i &= \sigma(\boldsymbol{W}_{i,s}\boldsymbol{s}_u^o(t^-) + \boldsymbol{W}_{i,m}\boldsymbol{m}_u(t) + \boldsymbol{b}_i), \\
\boldsymbol{g}_f &= \sigma(\boldsymbol{W}_{f,s}\boldsymbol{s}_u^o(t^-) + \boldsymbol{W}_{f,m}\boldsymbol{m}_u(t) + \boldsymbol{b}_f), \\
\hat{\boldsymbol{s}}_u^o(t) &= \tanh(\boldsymbol{W}_m\boldsymbol{m}_u(t) + \boldsymbol{g}_i \odot (\boldsymbol{W}_s\boldsymbol{s}_u^o(t^-) + \boldsymbol{b}_s) + \boldsymbol{b}), \\
\boldsymbol{s}_u^o(t) &= \boldsymbol{g}_f \odot \boldsymbol{s}_u^o(t) + (1 - \boldsymbol{g}_f) \odot \hat{\boldsymbol{s}}_u^o(t^-),
\end{aligned} \tag{3}$$

Method



Temporal Learning

$$[0, \tfrac{t_0}{n_t}), [\tfrac{t_0}{n_t}, 2\tfrac{t_0}{n_t})..., [(n_t - 1)\tfrac{t_0}{n_t}, t_o)$$

$[i\frac{t_0}{n_t}, (i+1)\frac{t_0}{n_t})$ $\boldsymbol{e}_i^t$ $\boldsymbol{e}_i^p$

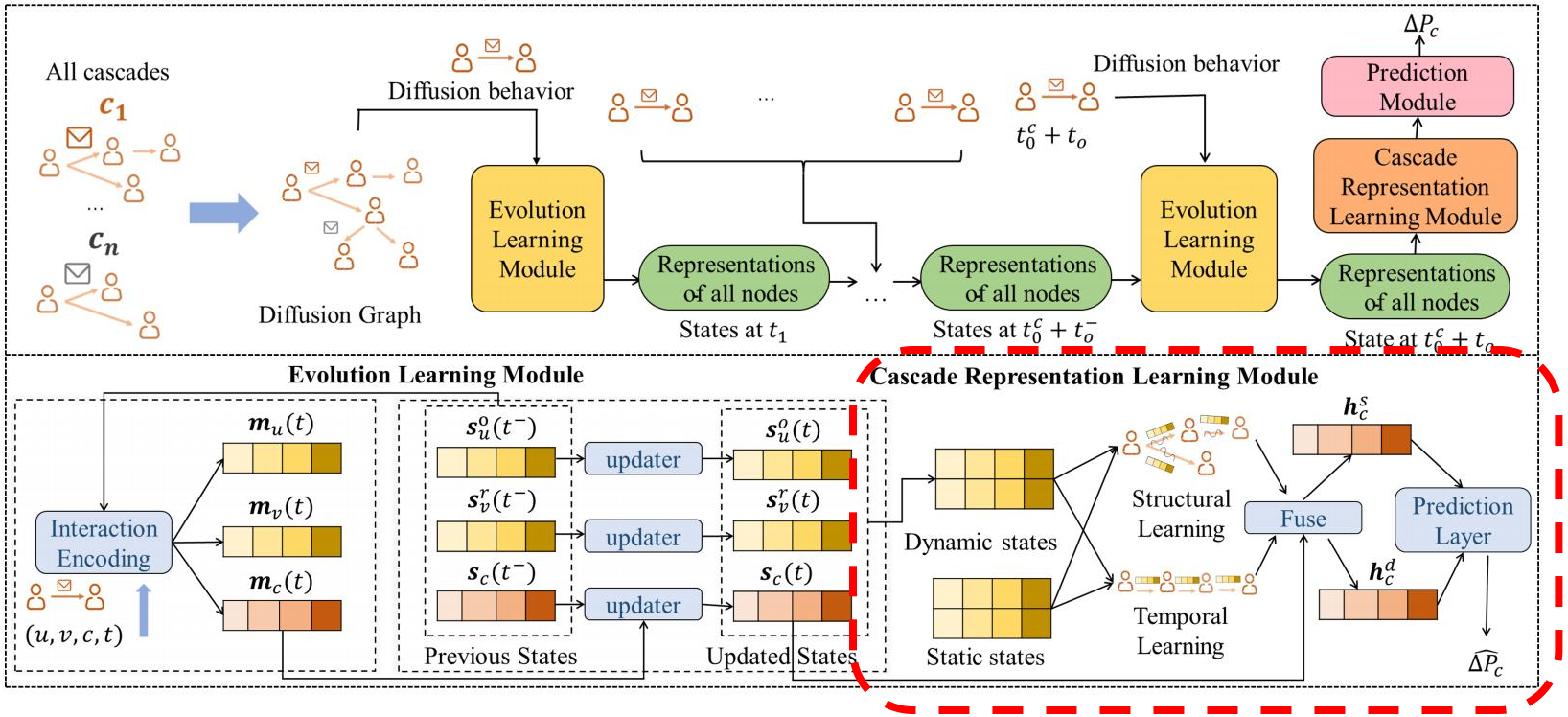
$$\boldsymbol{h}_{c,t}^s = LSTM^s([\boldsymbol{z}_{u_1}^s, \boldsymbol{z}_{u_2}^s, ..., \boldsymbol{z}_{u_n}^s]), \quad (4)$$

$$\boldsymbol{z}_{u_i}^s = \boldsymbol{s}_{u_i} + \boldsymbol{e}_{[t_i]}^t + \boldsymbol{e}_i^p, \quad (5)$$

$$[t_i] * \tfrac{t_0}{n_t} \leq t_i < ([t_i] + 1) * \tfrac{t_0}{n_t}$$

$\boldsymbol{h}_{c,t}^d$

Method



Structural Learning

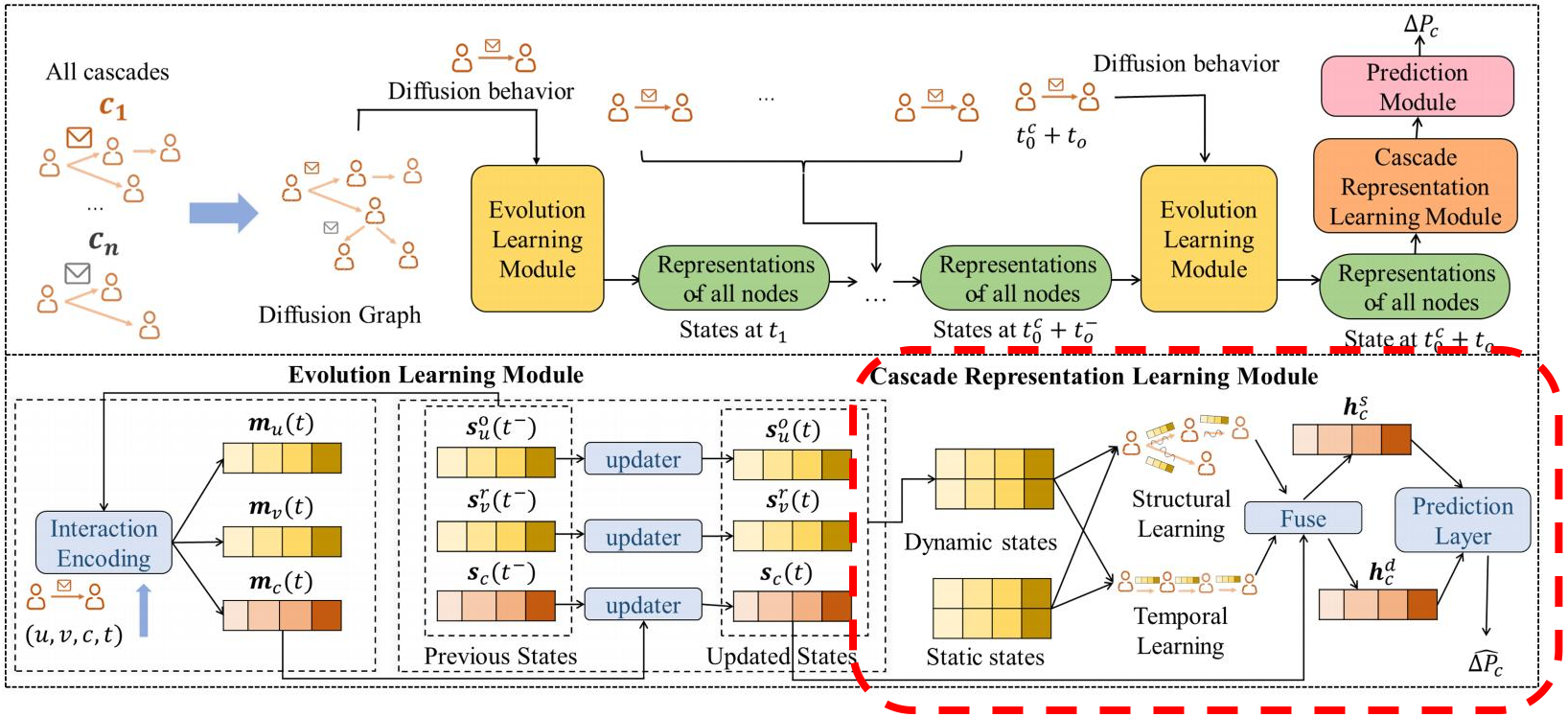
$$\tilde{\boldsymbol{h}}^s_{u,\uparrow} = \sum_{v \in S(u)} \boldsymbol{h}^s_{v,\uparrow},$$

$$\begin{aligned}
\boldsymbol{i}^s_{u,\uparrow} &= \sigma(\boldsymbol{W}^s_{i,\uparrow}[\boldsymbol{s}_u || \tilde{\boldsymbol{h}}^s_{u,\uparrow}] + \boldsymbol{b}^s_{i,\uparrow}), \\
\boldsymbol{f}^s_{uv,\uparrow} &= \sigma(\boldsymbol{W}^s_{f,\uparrow}[\boldsymbol{s}_u || \boldsymbol{h}^s_{v,\uparrow}] + \boldsymbol{b}^s_{f,\uparrow}), \\
\boldsymbol{o}^s_{u,\uparrow} &= \sigma(\boldsymbol{W}^s_{o}[\boldsymbol{s}_u || \tilde{\boldsymbol{h}}^s_{u,\uparrow}] + \boldsymbol{b}^s_{o,\uparrow}), \\
\boldsymbol{g}^s_{u,\uparrow} &= \tanh(\boldsymbol{W}^s_{g,\uparrow}[\boldsymbol{s}_u || \tilde{\boldsymbol{h}}^s_{u,\uparrow}] + \boldsymbol{b}^s_{g,\uparrow}), \\
\boldsymbol{c}^s_{u,\uparrow} &= \boldsymbol{i}^s_{u,\uparrow} \odot \boldsymbol{g}^s_{u,\uparrow} + \sum_{v \in S(u)} \boldsymbol{f}^s_{uv,\uparrow} \odot \boldsymbol{c}^s_{v,\uparrow} \\
\boldsymbol{h}^s_{u,\uparrow} &= \boldsymbol{o}^s_{u,\uparrow} \odot \tanh(\boldsymbol{c}^s_{u,\uparrow}),
\end{aligned} \tag{6}$$

$$\boldsymbol{h}^s_{c,\uparrow} = \sum \boldsymbol{h}^s_{leaf,\uparrow} \qquad \boldsymbol{h}^s_{c,\downarrow}$$

$$\boldsymbol{h}^s_{c,s} \qquad \boldsymbol{h}^d_{c,s}$$

Method



Embedding Fusion

$h_{c,t}^s$ $h_{c,s}^s$ h_c^s

$$h_c^d = \sigma(W_a[h_{c,t}^d || h_{c,s}^d || \tilde{h}_c^d]), \quad (7)$$

$$\tilde{h}^d = \sigma(W_b[\tilde{s}_c(t) || s_u^o(t) || s_v^r(t)]), \quad (8)$$

$$\tilde{s}_c(t) = s_c(t) + e_{[t_0^c]}^g, \quad (9)$$

Prediction Module

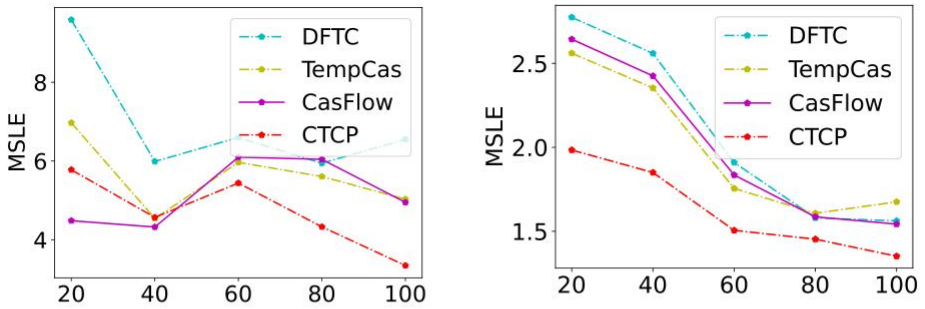
$$\widehat{\Delta P}_c = \lambda f_{\text{static}}(h_c^s) + (1-\lambda) f_{\text{dynamic}}(h_c^d), \quad (10)$$

$$\mathcal{J}(\theta) = \frac{1}{n}\sum_c (log(\Delta P_c) - log(\widehat{\Delta P_c}))^2, \quad (11)$$

Experiments

Datasets	#Users	#Cascades	#Retweets
Twitter	199,005	19,718	602,253
Weibo	918,852	39,076	1,572,287
APS	218,323	48,575	939,686

Table 1: Statistics of datasets.



(a) Twitter (b) APS

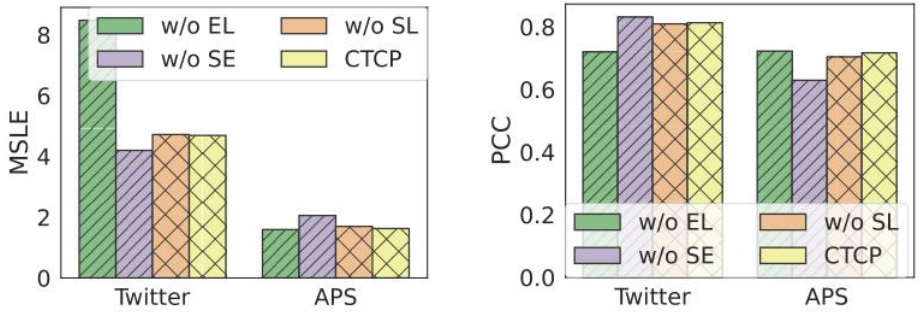
Figure 3: Performance on cascades with different publication times. The horizontal axis indicates the percentile range of the publication time of cascades and the vertical axis is the average prediction MSLE of cascade in that range.

Experiments

Model	Twitter				Weibo				APS			
	MSLE	MALE	MAPE	PCC	MSLE	MALE	MAPE	PCC	MSLE	MALE	MAPE	PCC
XGBoost	11.5330	2.9871	0.8571	0.3792	3.6253	1.3736	0.3571	0.6493	2.5808	1.2559	0.3437	0.4762
MLP	11.9105	2.9712	0.9324	0.3733	3.9370	1.4409	0.3812	0.6098	2.6075	1.2577	0.3516	0.4787
DeepHawkes	7.7795	2.1553	0.5547	0.6500	4.2520	1.4658	0.3998	0.5670	2.3356	1.2001	0.3158	0.5524
DFTC	5.9173	1.8426	0.4851	0.7495	2.9370	1.2046	0.2959	0.7296	2.0357	1.1159	0.2943	0.6247
CasCN	7.1021	2.0567	0.5231	0.6940	3.7714	1.4040	0.3612	0.6707	2.1248	1.1358	0.3035	0.6062
MS-HGAT	5.9992	1.9006	0.4741	0.7507	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
TempCas	5.5870	1.7584	0.4574	0.7651	2.7453	1.1702	0.2786	0.7500	2.0043	1.1022	0.2957	0.6346
CasFlow	5.2549	1.5775	0.4031	0.7847	2.6336	**1.1230**	**0.2687**	0.7619	2.0064	1.1053	0.2936	0.6320
CTCP	**4.6916**	**1.5668**	**0.3562**	**0.8136**	**2.5929**	1.1414	0.2723	**0.7667**	**1.6289**	**0.9906**	**0.2611**	**0.7176**

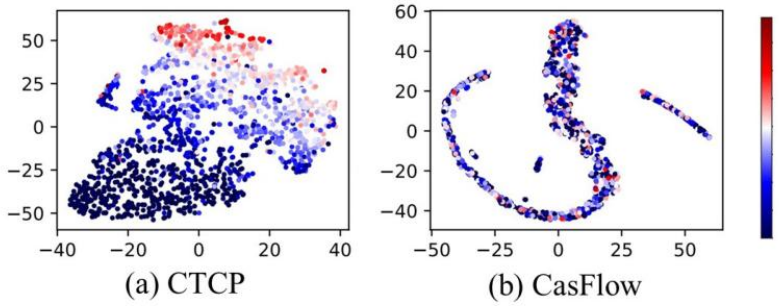
Table 2: Performance of all methods in three datasets, where the methods can be divided into three categories: feature-based, sequence-based, and graph-based methods from top to bottom in the table. The best results appear in bold and OOM indicates the out-of-memory error.

Experiments



(a) Performance on MSLE (b) Performance on PCC

Figure 4: Ablation study on Twitter and APS.



(a) CTCP (b) CasFlow

Figure 5: Projection of learned cascade representations on Twitter, where each point represents a cascade representation and the color represents its incremental popularity. (Dark blue means low popularity and dark red means high popularity).

Thank you!