Transformer-Style Relational Reasoning with Dynamic Memory Updating for Temporal Network Modeling

Dongkuan Xu[1], Junjie Liang[1], Wei Cheng[2], Hua Wei[1], Haifeng Chen[2], Xiang Zhang[1]

[1]The Pennsylvania State University, [2]NEC Laboratories America, Inc.

PennState

NEC Laboratories America

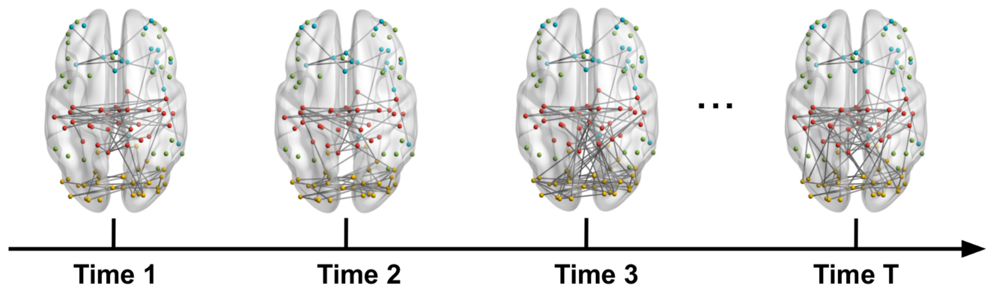
Relentless passion for innovation

Agenda

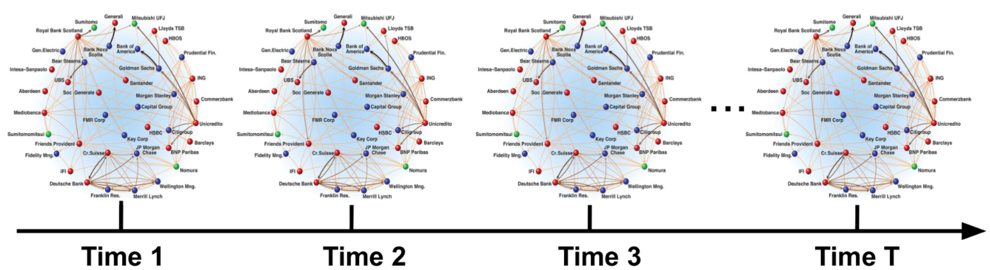
- Motivations
- Challenges
- Problem Definition
- Proposed Model: TRRN
- Experiments

Motivations

- Networks are often dynamic and evolve over time



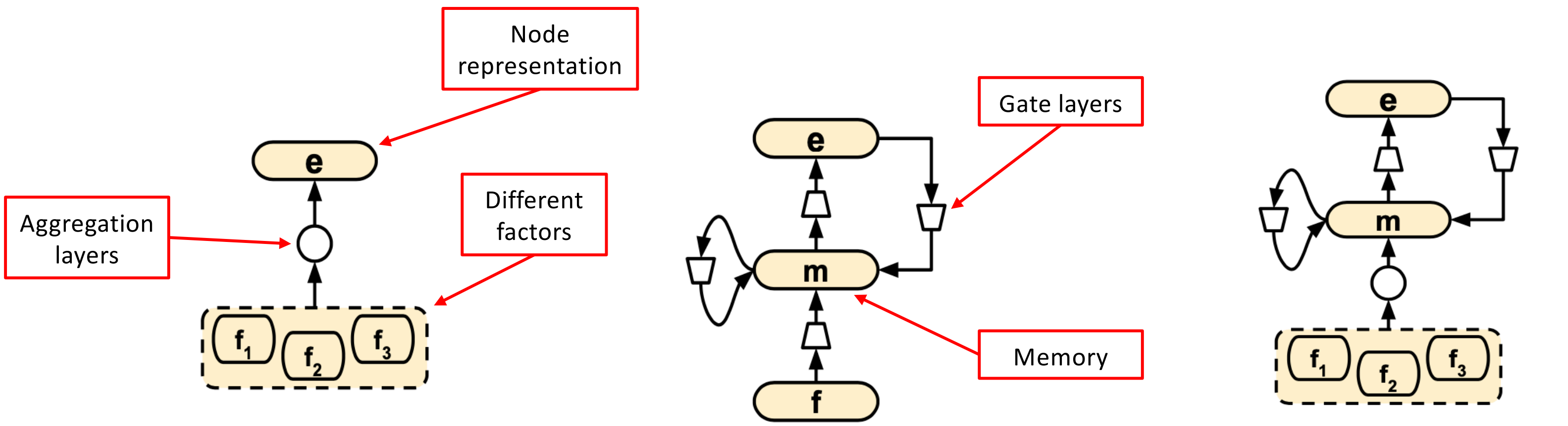
Temporal Brain Network



Temporal Financial Network

Existing Approaches

- Three architectures for temporal network modeling



(a) Graph NNs (b) Gated RNNs (c) Spatio-temporal NNs

- However, their capacity for learning complicated temporal dependency is limited

Challenges

- Complicated temporal dependency
  - How to relate the information of different time steps

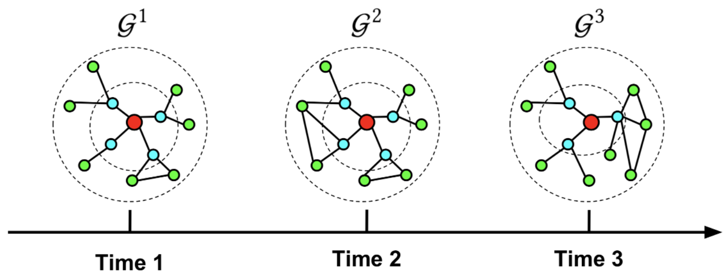
- Contextualized representations of nodes
  - Node behaviors are determined by various factors
  - How to embed them into node representations and differentiate their influence

- Sparse dynamics in the evolution of temporal networks
  - Dynamic systems are characterized by independent and sparsely interacting dynamic processes [1]
  - How to model the hidden dynamic processes

[1] Goyal, Anirudh, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. "Recurrent independent mechanisms." *arXiv preprint arXiv:1909.10893* (2019).

Problem Definition

- Temporal network



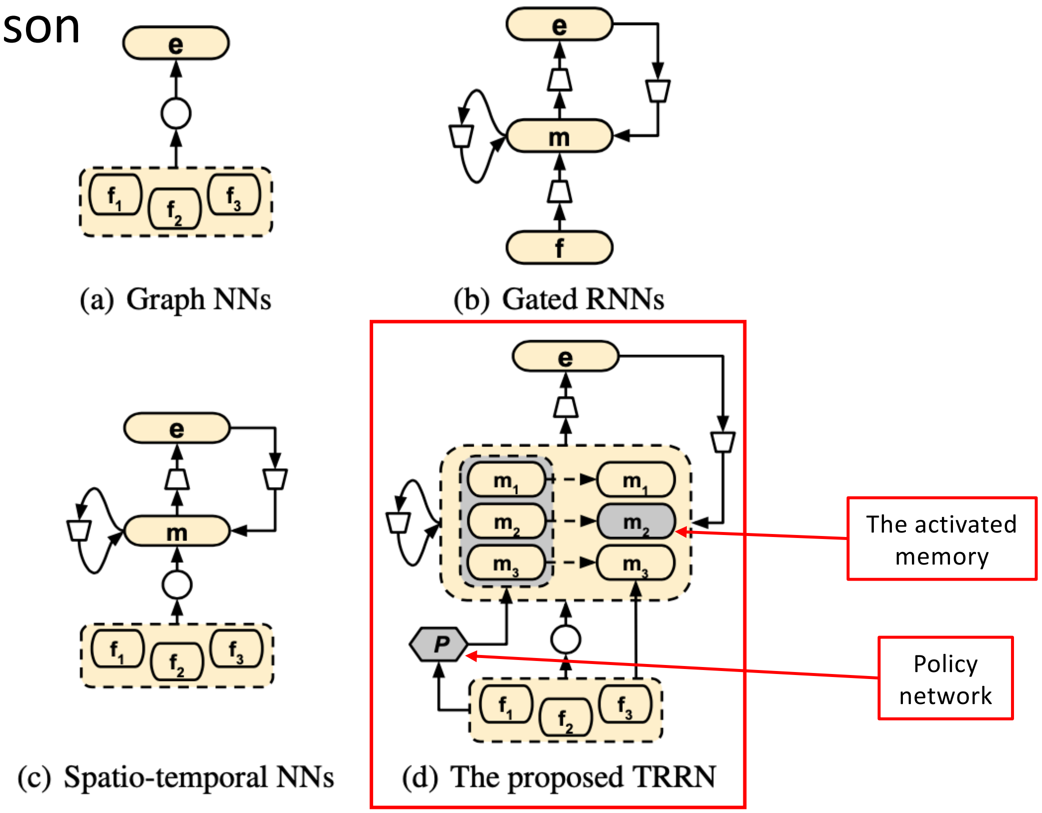
$\mathbb{G} = (\mathcal{G}^1, \mathcal{G}^2, \cdots, \mathcal{G}^T)$, where $\mathcal{G}^t = (\mathcal{V}, \mathbf{A}^t, \mathbf{X}^t)$

Node set $\mathcal{V}$, Adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{N \times N}$, Node attributes $\mathbf{X}^t \in \mathbb{R}^{N \times d}$

- Temporal network modeling
  - The goal is to learn the latent representation for every node at different time steps
  - The latent representations should preserve various factors related to nodes

Proposed Model: TRRN

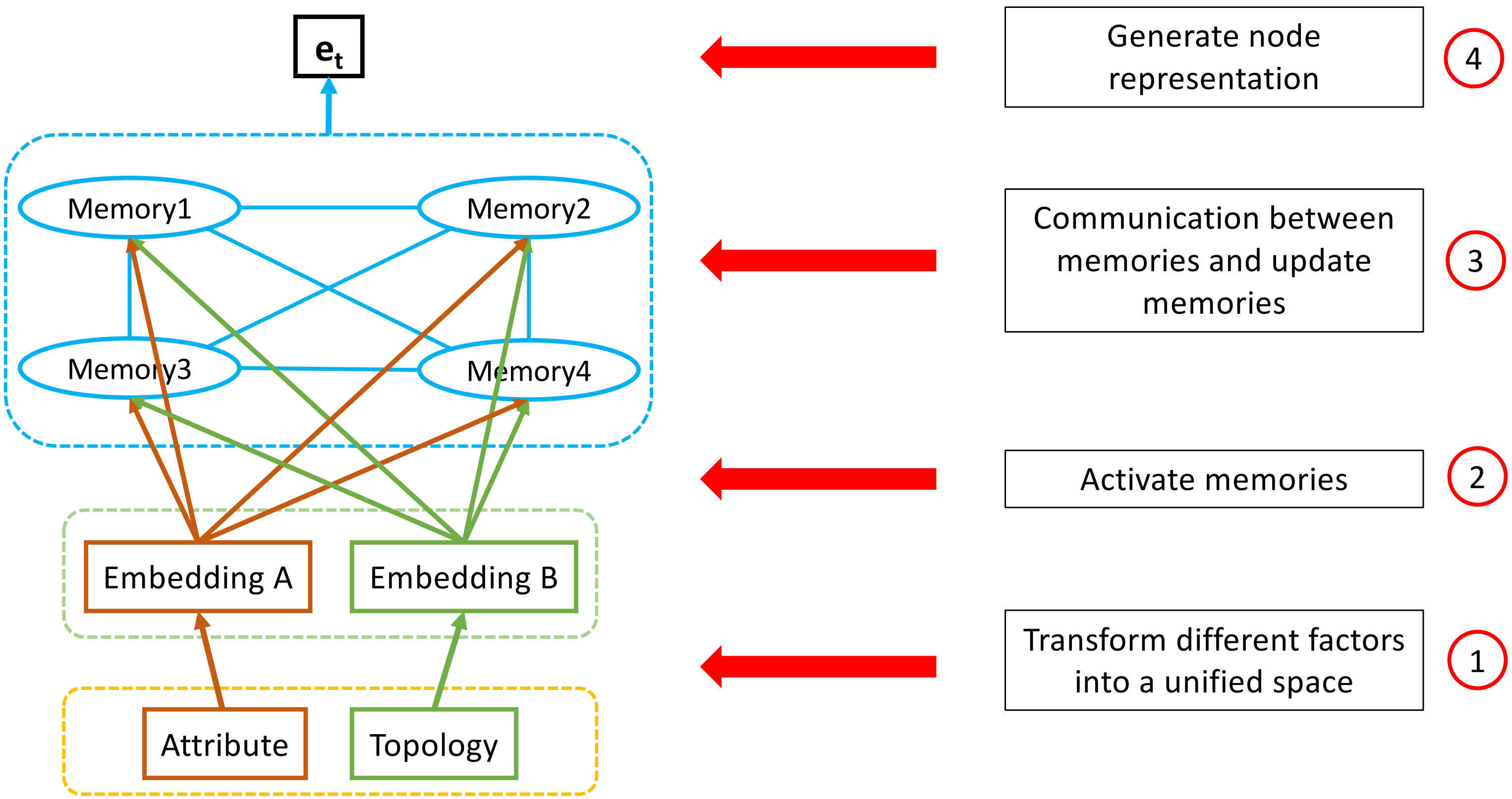
- Architecture Comparison



Four architectures for temporal network modeling

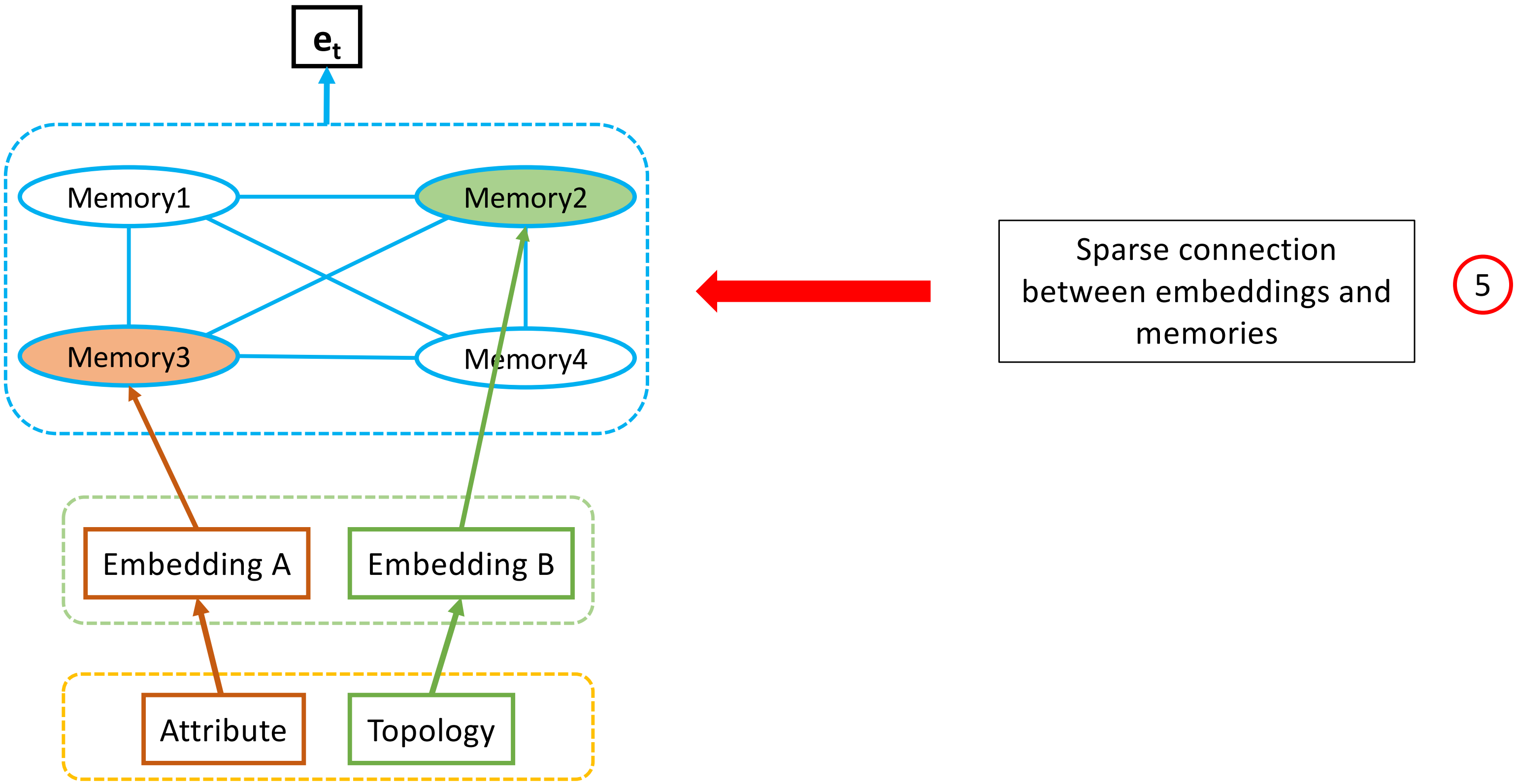
Step-by-step Analysis of TRRN

- Step 1. Activate memories separately



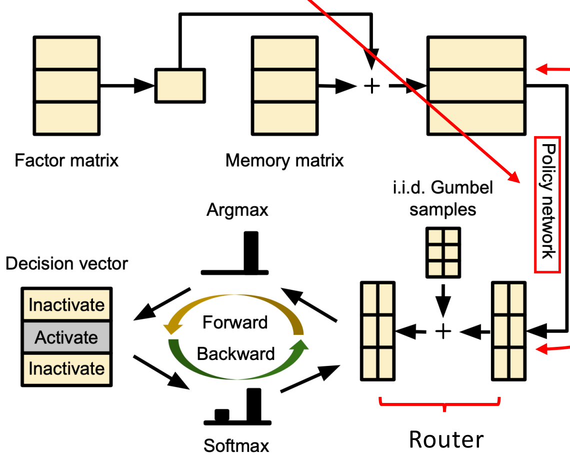
Step-by-step Analysis of TRRN

- Step 1. Activate memories separately; Step 2. Sparsity constrains



Activate Memories

- Policy network estimates the probability of each memory being activated
- Output of policy network, i.e., B
  - Fed into Routers to produce the binary activation decisions
  - Rows correspond to different memories
  - Columns represent two categories (activate and inactivate)



$$\mathbf{B} = PolicyNetwork(\begin{bmatrix} \mathbf{m}_1^\top \oplus \tilde{\mathbf{f}}^\top \\ \mathbf{m}_2^\top \oplus \tilde{\mathbf{f}}^\top \\ \cdots \\ \mathbf{m}_{n_M}^\top \oplus \tilde{\mathbf{f}}^\top \end{bmatrix}) \in \mathbb{R}^{n_M \times 2}$$

Illustration of activating memories selectively

Update Memories

- Self-Attention

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}$$

- Mask inactivated memories
- Concatenate memories and factors

z is the mask

$$\mathbf{Q} = \boxed{(\mathbf{M} * \mathbf{z})}\,\boldsymbol{W}^q,$$
$$\mathbf{K} = \left[\mathbf{M}; \boxed{\mathbf{f}_1^\top; \mathbf{f}_2^\top; \cdots; \mathbf{f}_{n_F}^\top}\right]\boldsymbol{W}^s,$$
$$\mathbf{V} = \left[\mathbf{M}; \mathbf{f}_1^\top; \mathbf{f}_2^\top; \cdots; \mathbf{f}_{n_F}^\top\right]\boldsymbol{W}^v,$$
$$\boxed{\tilde{\mathbf{M}}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_s}}\right)\mathbf{V} + \boxed{\mathbf{M} * (\mathbf{1} - \mathbf{z})} + \boxed{\mathbf{M}},$$

Updated memories Inactivated memories A residual connection

Experiments

- Data sets

Statistics	Task-fMRI	DBLP5	Epinions	Reddit
# Nodes	5000	6606	16025	8291
# Edges	1955488	42815	1144258	264050
# Attributes	20	100	20	20
# Time Steps	12	10	11	10
# Node Categories	10	5	10	4

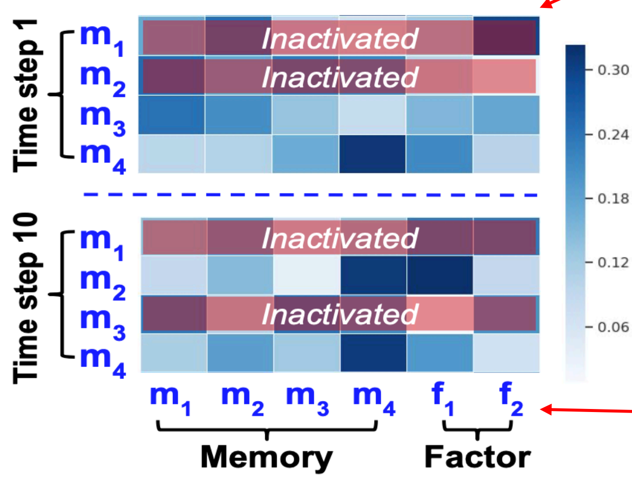
- Node classification comparison[1]

Table 3: Node classification results (%).

Method	Task-fMRI		DBLP5		Epinions		Reddit	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
DeepWalk	71.4±1.3	97.2±1.0	35.4±1.2	61.0±1.8	30.1±1.6	68.4±1.8	47.5±1.7	71.9±2.4
GAT	43.8±2.5	86.2±3.4	32.5±2.4	48.6±2.9	22.5±1.5	63.1±1.8	29.6±1.9	52.4±2.6
GCN	65.0±1.4	86.7±0.9	33.7±1.3	50.0±1.2	20.9±0.7	62.4±1.4	27.7±0.8	54.0±1.0
GraphSAGE	69.4±2.6	96.7±2.1	71.0±1.1	90.7±1.9	24.5±2.9	63.9±2.0	42.5±2.1	66.8±2.5
node2vec	71.0±1.5	96.8±1.8	36.9±1.1	64.2±1.1	32.8±1.5	70.2±1.6	48.0±1.3	72.2±1.1
LSTM	83.6±1.8	98.6±1.5	74.1±0.6	91.4±0.8	17.9±1.0	61.5±1.2	40.2±1.4	66.5±1.6
GRU	80.4±1.7	98.2±1.7	75.6±1.0	91.5±1.1	17.3±1.1	61.7±1.6	42.1±1.4	67.2±1.7
DynGEM	71.0±2.7	97.2±2.7	52.3±3.2	59.0±3.4	31.6±2.4	54.6±2.5	39.9±3.5	66.2±2.8
DANE	85.2±1.3	94.8±2.9	82.5±1.7	92.3±1.0	31.8±1.8	67.1±1.8	45.7±1.9	70.0±1.6
STAR	89.2±1.2	99.2±0.8	80.3±1.5	95.5±0.7	32.6±1.6	67.4±1.5	50.8±1.3	75.0±1.7
TRRN	**91.5±1.0**	**99.8±0.8**	**88.9±1.5**	**97.6±1.8**	**34.6±1.3**	**72.8±1.2**	**52.0±1.7**	**79.2±1.8**

1. Link prediction comparison is summarized in paper

Analysis of memory activating & updating



Visualize the self-attention weight matrix, i.e., softmax $(\mathbf{QK}^\top/\sqrt{d_s})$

f_1 is attribute factor
f_2 is topology factor

The visualization of self-attention matrix of a node from DBLP5 at two time steps.

Thanks!

Q & A