Meta-Learned Metrics over Multi-Evolution Temporal Graphs





Dongqi Fu* (UIUC)

Liri Fang* (UIUC)

Ross Maciejewski (ASU)

Vetle I. Torvik (UIUC)

Jingrui He (UIUC)

Presenter: Dongqi Fu

Email: dongqif2@illinois.edu

Homeland Security

NIH National Institutes of Health

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

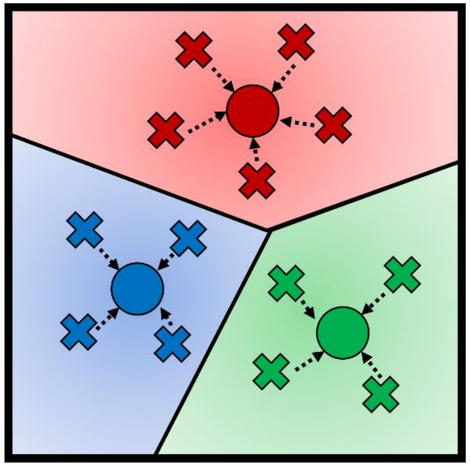
Roadmap

- Motivation
- Proposed Temp-GFSM Framework
- Experiments
- Conclusion

Graph Metric Learning

- Graph metric learning aims to learn a proper distance metric among graphs from the input space, which reflects their underlying relationship

- Good graph metrics could contribute to many real-world applications
  - Protein and drug discovery [1]
  - Molecular property prediction [2]
  - Epidemic infectious pattern analysis [3]
  - and many more ……

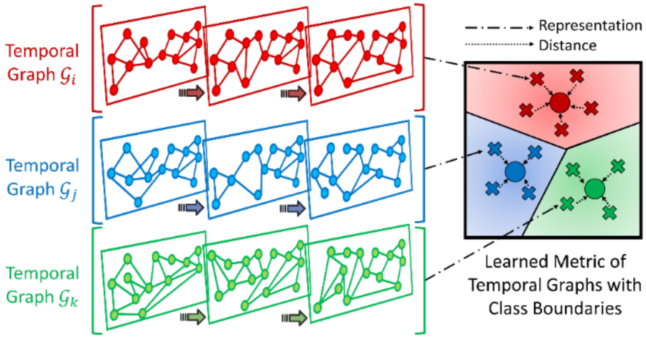


X: Graphs O: Classes

- Currently, the dynamics of graphs are largely overlooked in the graph metric learning process

[1] Dai et al., Discriminative embeddings of latent variable models for structured data. ICML 2016
[2] Gilmer et al., Neural message passing for quantum chemistry. ICML 2017
[3] Derr et al., Epidemic graph convolutional network. WSDM 2020

Problem Definition

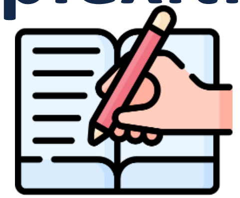


- Learning metrics over temporal graphs
  - Input: a set of n temporal graphs $D = \{G_1, \ldots, G_n\}$
  - Output: the metric space $\mathbb{M}$ parameterized by θ, where similar (e.g., same class) temporal graphs are closer and dissimilar (e.g., different class) graphs are farther apart

- Preliminary
  - Learning a metric over a bunch of data items ≈ the problem of extracting their hidden representation vectors [4,5,6,7]
    - Our objective is learning a metric $\mathbb{M}$ (i.e., mapping or transformation function f_θ) controlled by parameters θ

[4] Globerson et al., Metric learning by collapsing classes. NeurIPS 2005
[5] Salakhutdinov et al., Learning a nonlinear embedding by preserving class neighbourhood structure. AISTATS 2007
[6] Snell et al., Prototypical networks for few-shot learning. NeurIPS 2017.
[7] Wang et al., Survey on distance metric learning and dimensionality reduction in data mining. Data Min. Knowl. Discov. 2015

Temporal Information Brings Complexities



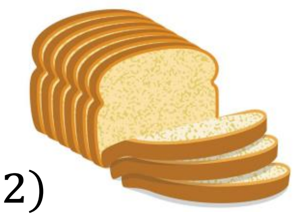
- Suppose you are writing an excellent article ……
  - When you **select a word**, you may want to make its nearby context words share the similar meaning
  - After you **compose a sentence**, you may want the current sentence support (or reject) previous (or following) sentences
  - Then you **finish paragraphs**, and you may want to make the whole article catchy and fluent

- When the similar logic applies to graphs physical evolutions
  - We need the **multi-dynamics** to **model temporal graphs**
  - Could contribute to identifying dissemination processes like rumors and diseases [8]

[8] Oettershagen et al., Temporal Graph Kernels for Classifying Dissemination Processes. SDM 2020

Multi-Dynamics of Temporal Graphs

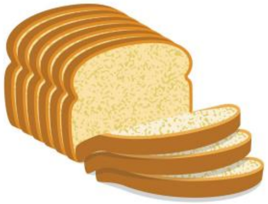
- What are evolving in temporal graphs?
  - **Streaming** [9] (or **continuous-time** [10])
    - An initial state G with a set of timestamped events O, each event can be node/edge addition/deletion
    - Rapid node/edge-level evolution, i.e., microscopic evolution [11] such as protein interactions in a cell [13]



  - **Snapshot** [9] (or **discrete-time** [10])
    - A sequence of time-respecting snapshots $G^{(1)}, G^{(2)}, \ldots\ldots$
    - Episodic and slowly-changing evolution patterns, i.e., macroscopic evolution [12] such as periodical metabolic cycles in a cell [13]

[9] Aggarwal et al., Evolutionary network analysis: A survey. ACM Comput. Surv., 2014
[10] Kazemi et al., Representation learning for dynamic graphs: A survey. JMLR 2020
[11] Leskovec et al., Microscopic evolution of social networks. KDD 2008
[12] Leskovec et al., Graphs over time: densification laws, shrinking diameters and possible explanations. KDD 2005
[13] Fu et al., DPPIN: A biological repository of dynamic protein-protein interaction network data. CoRR, 2021

Challenges



- From **multi-dynamic** evolution patterns
  - C.1 - How to **integrate** streaming and snapshot patterns?
  - C.2 – How to find **dominating** evolution patterns for the similarity (e.g., class labels) of input temporal graphs?

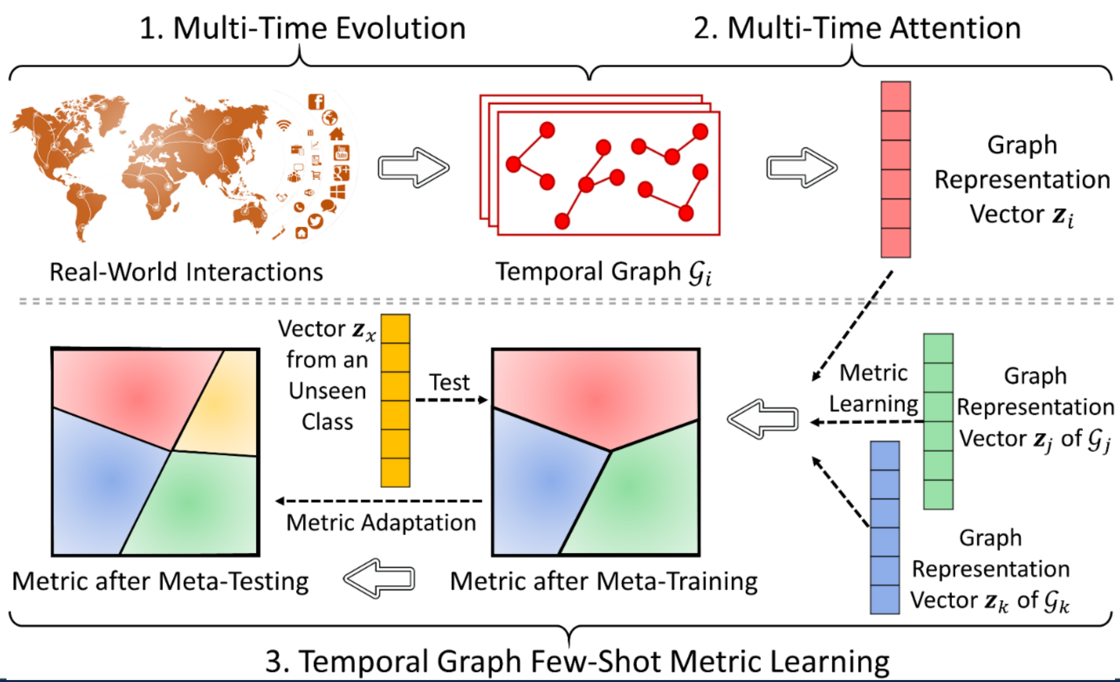
- From the **label scarcity** of temporal graphs
  - C.3 - How to ensure the **accuracy** of $\mathbb{M}$ when the learning process could not leverage large amount of labeled data?
  - C.4 - How to **adapt** the learned $\mathbb{M}$ to a new subspace for newly arrived classes while maintaining the existing classes?

Roadmap

- Motivation
- Proposed Temp-GFSM Framework
- Experiments
- Conclusion

Overview of Temp-GFSM

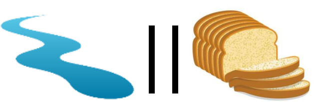
- An end-to-end trainable framework
  - Multi-Time Evolution
  - Multi-Time Attention
  - Temporal Graph Few-Shot Metric Learning

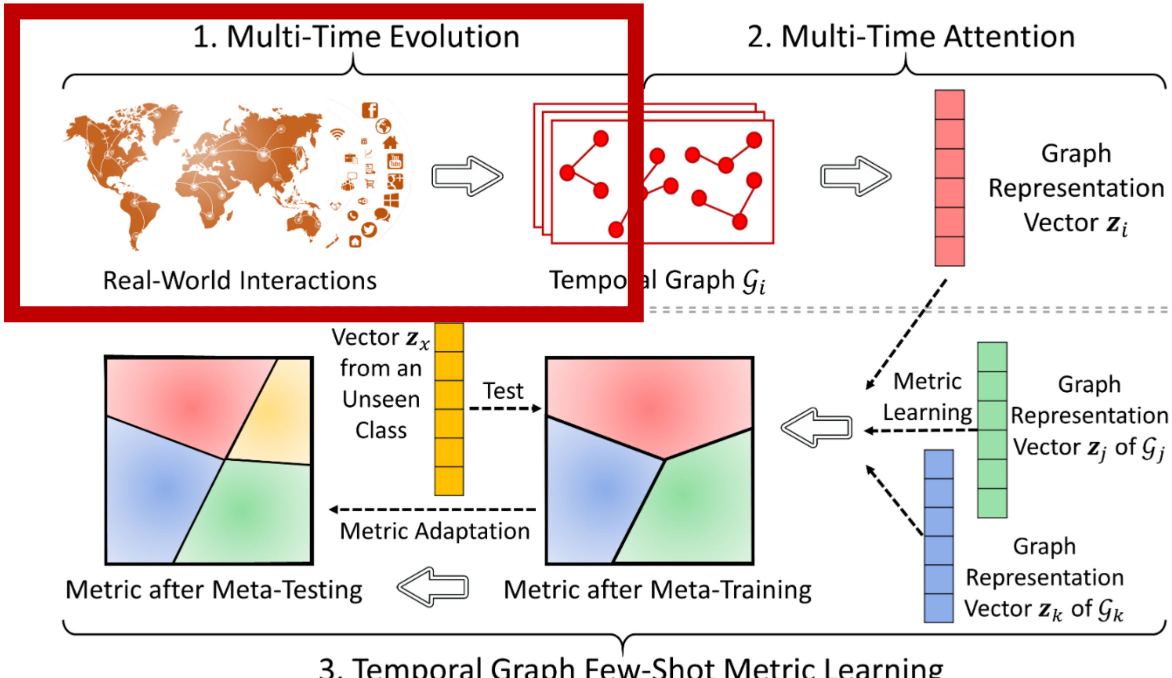


Overview of Temp-GFSM

- An end-to-end trainable framework
  - Multi-Time Evolution (Carrying Multiple Dynamics)
  - Multi-Time Attention
  - Temporal Graph Few-Shot Metric Learning

Challenge C.1

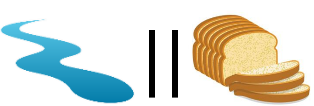


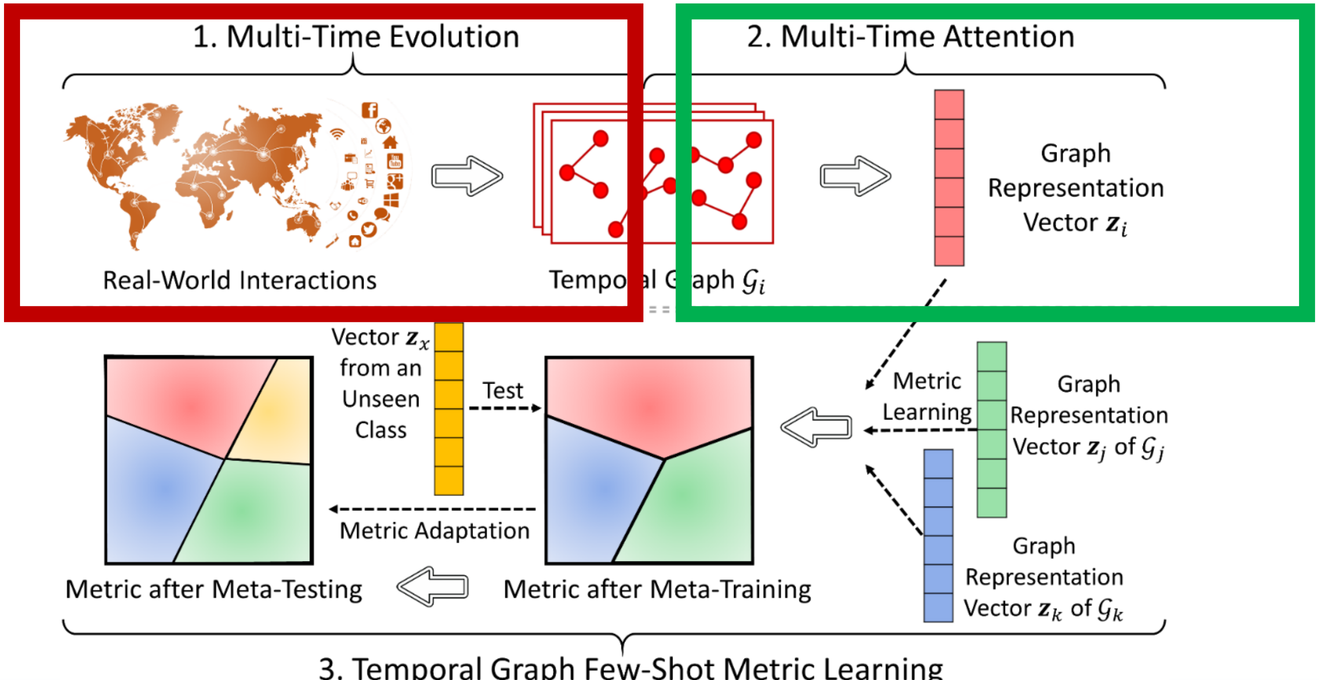


Overview of Temp-GFSM

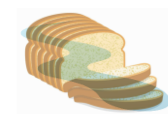
- An end-to-end trainable framework
  - Multi-Time Evolution (Carrying Multiple Dynamics)
  - Multi-Time Attention (Weighting Multiple Dynamics)
  - Temporal Graph Few-Shot Metric Learning

Challenge C.1





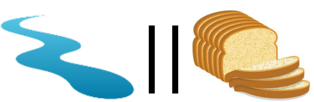
Challenge C.2

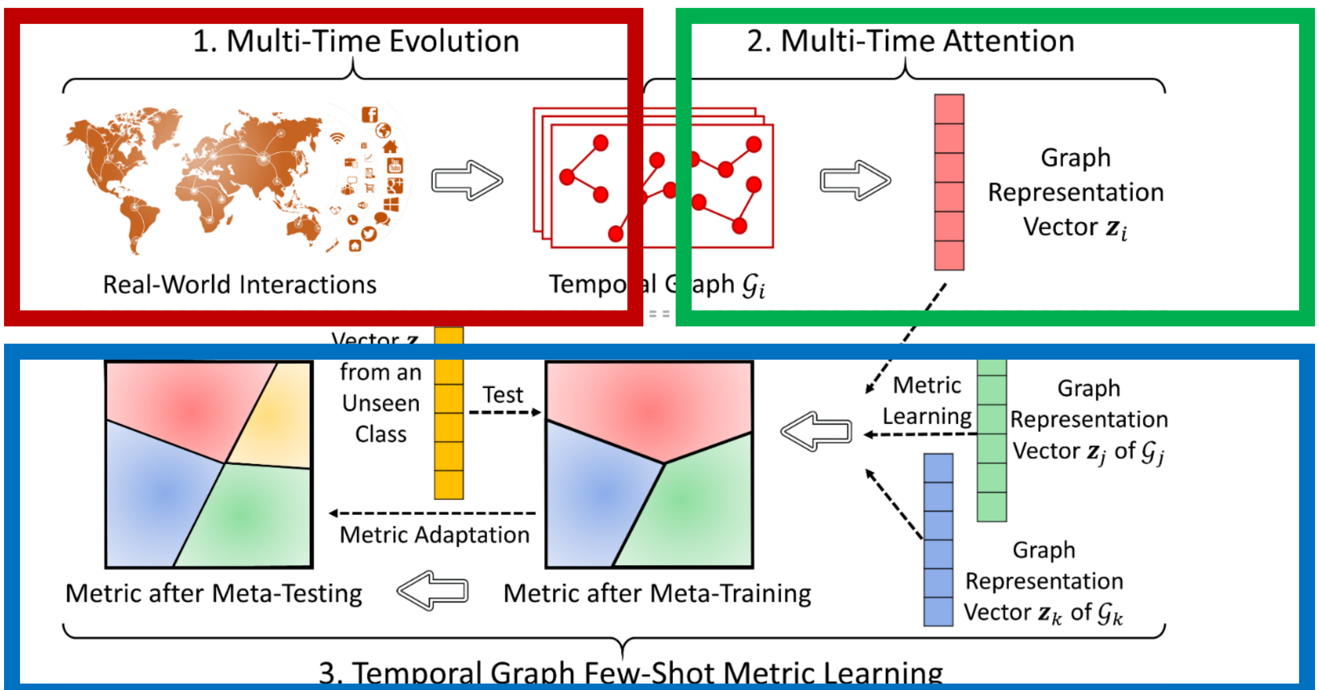


Overview of Temp-GFSM

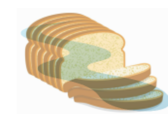
- An end-to-end trainable framework
  - Multi-Time Evolution (Carrying Multiple Dynamics)
  - Multi-Time Attention (Weighting Multiple Dynamics)
  - Temporal Graph Few-Shot Metric Learning (New Class Adaption)

Challenge C.1

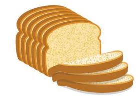


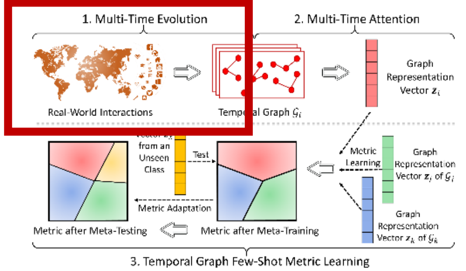


Challenge C.2



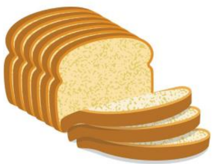
Challenge C.3 & C.4

Multi-Time Evolution  || 

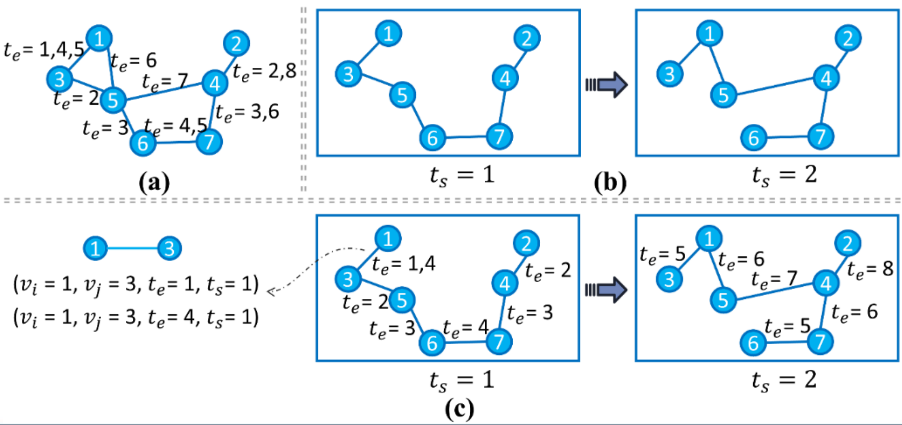


- A fundamental container model carrying multi-dynamic evolution patterns for the next learning process

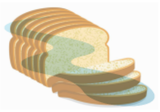
edge timestamp (continuous [10])   snapshot timestamp (discrete [10])

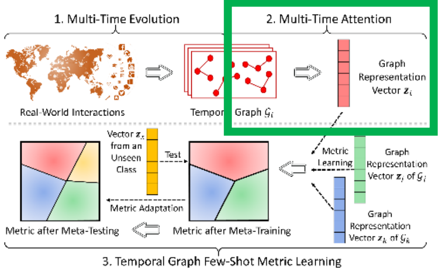


- An edge is marked as (v_i, v_j, t_e, t_s), where
  - (v_i, v_j, t_e) means the connection between v_i and v_j exists at time t_e
  - (v_i, v_j, t_e, t_s) means the event (v_i, v_j, t_e) happens in the window of snapshot $S^{(t_s)}$

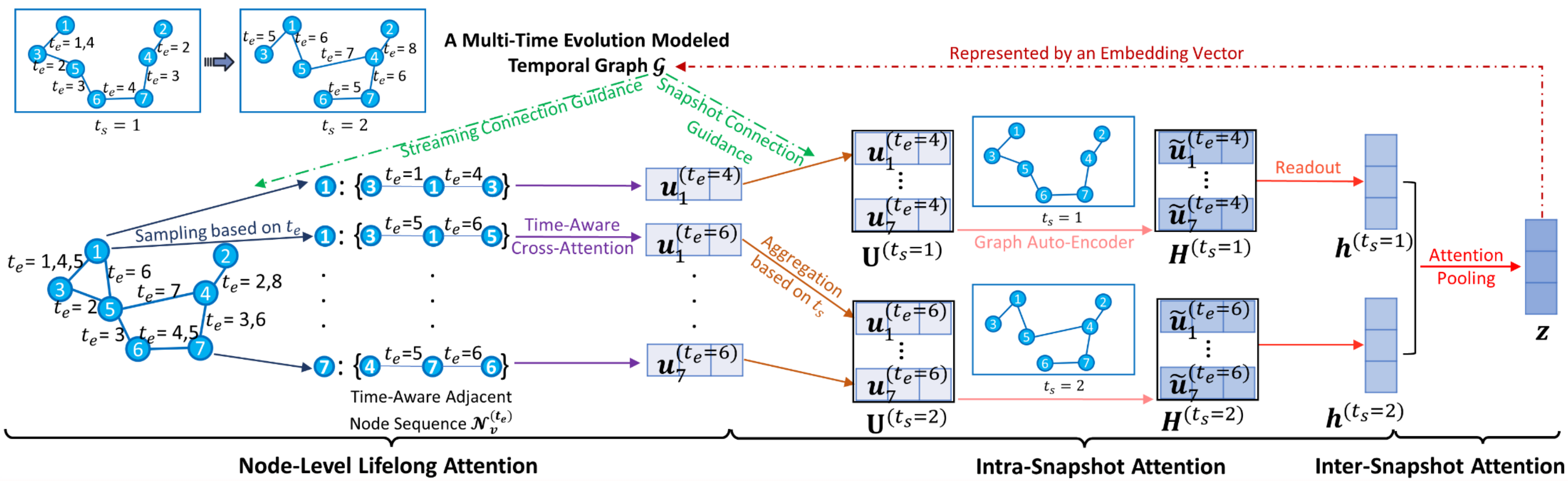


Multi-Time Attention

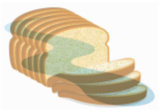


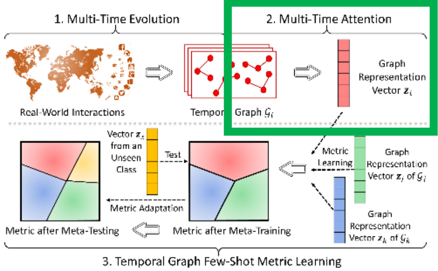


- To assign different weights to different evolution patterns, such that the learned metric could reflect different class properties
- Temporal graph G -> representation vector Z
  - Node-Level Lifelong Attention
  - Intra-Snapshot Attention
  - Inter-Snapshot Attention

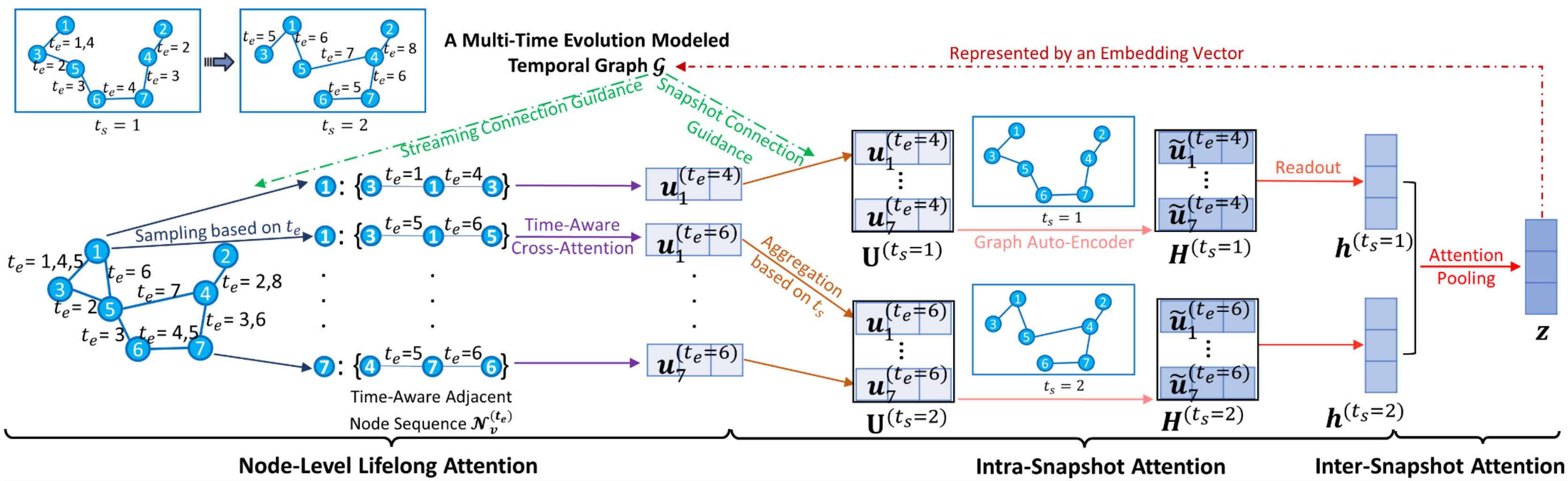


Multi-Time Attention

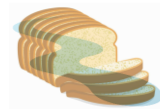


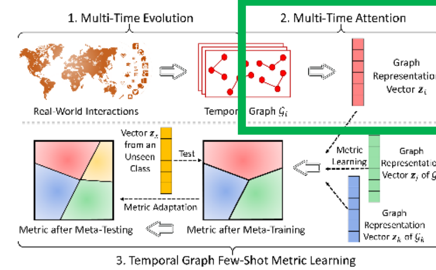


- To assign different weights to different evolution patterns, such that the learned metric could reflect different class properties
- Temporal graph G -> representation vector Z
  - Node-Level Lifelong Attention (Select Meaningful Words)
  - Intra-Snapshot Attention (Compose Supportive Sentences)
  - Inter-Snapshot Attention (Finish a Fluent Article with Paragraphs)

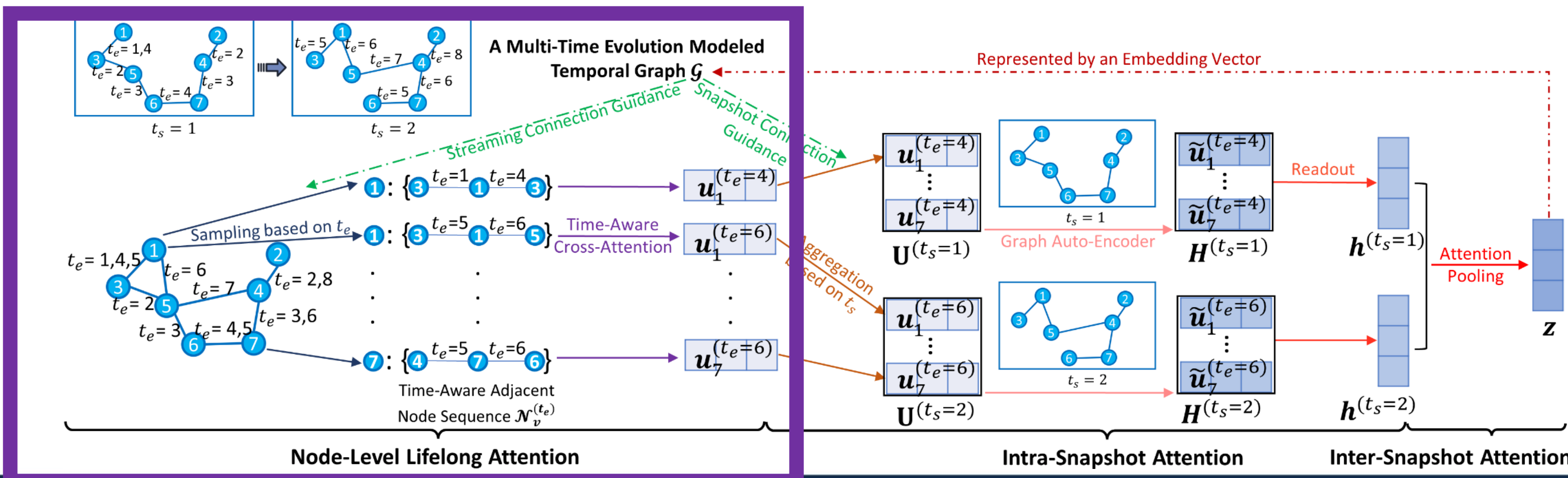


Multi-Time Attention

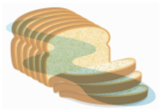


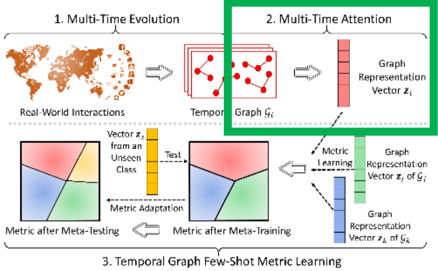


- To assign different weights to different evolution patterns, such that the learned metric could reflect different class properties
- Temporal graph G -> representation vector Z
  - Node-Level Lifelong Attention (Streaming Evolution)
  - Intra-Snapshot Attention
  - Inter-Snapshot Attention

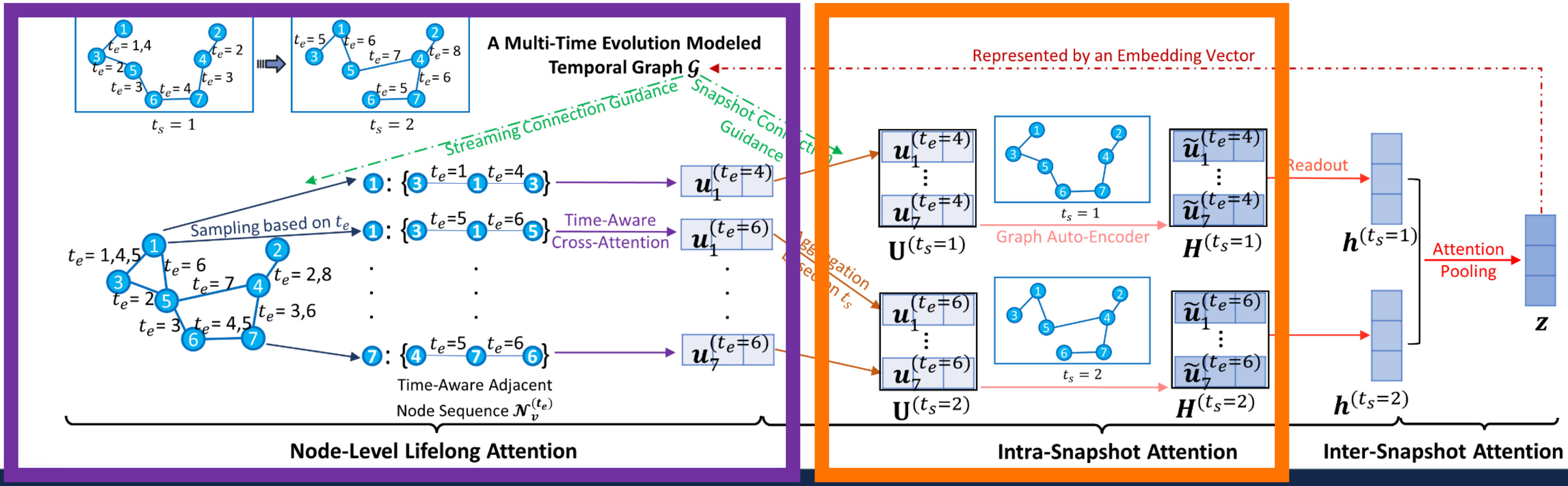


Multi-Time Attention

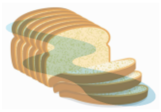


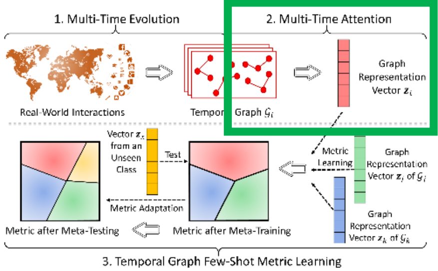


- To assign different weights to different evolution patterns, such that the learned metric could reflect different class properties
- Temporal graph G -> representation vector Z
  - Node-Level Lifelong Attention (Streaming Evolution)
  - Intra-Snapshot Attention (Streaming in Episodic Evolution)
  - Inter-Snapshot Attention

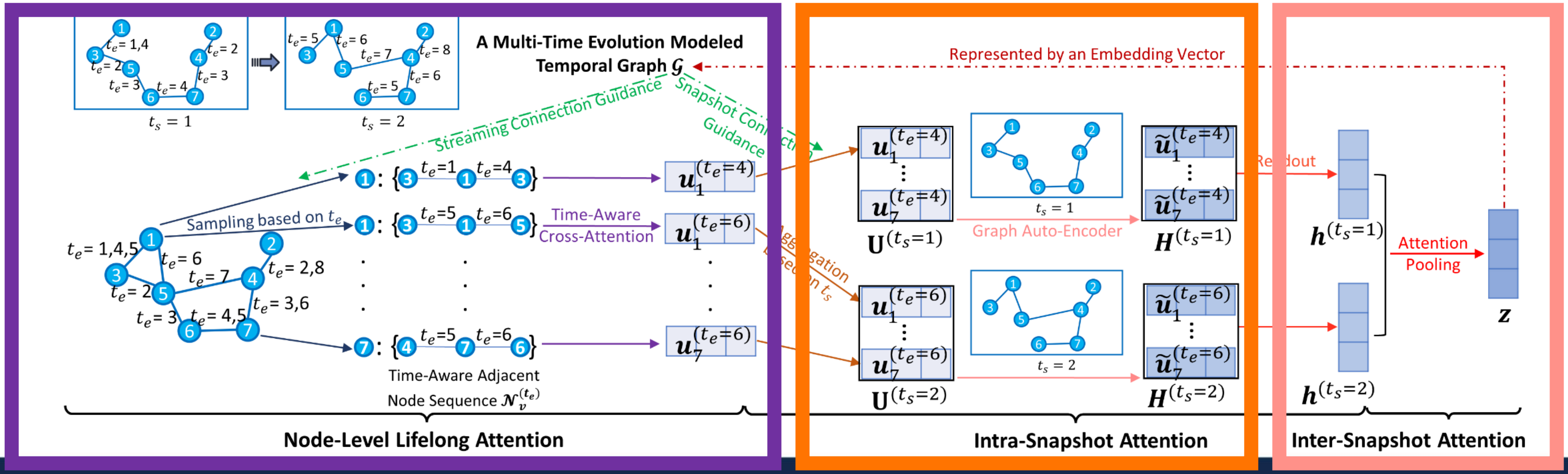


Multi-Time Attention

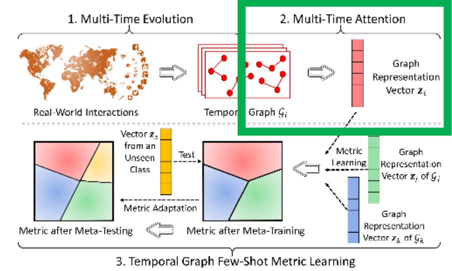




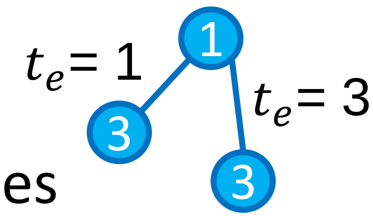
- To assign different weights to different evolution patterns, such that the learned metric could reflect different class properties
- Temporal graph G -> representation vector Z
  - Node-Level Lifelong Attention (Streaming Evolution)
  - Intra-Snapshot Attention (Streaming in Episodic Evolution)
  - Inter-Snapshot Attention (Episodic Encoding)



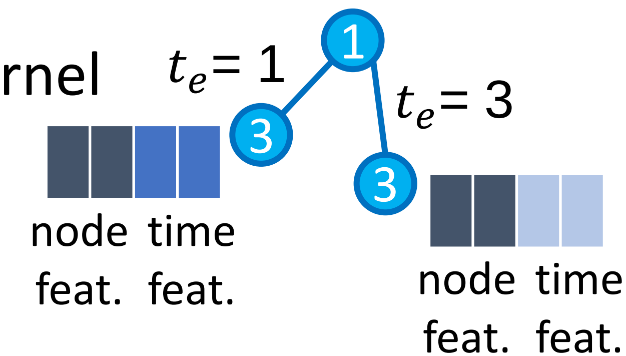
Node-Level Lifelong Attention



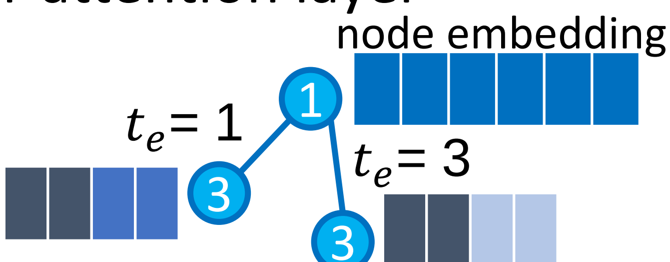
- Capturing node streaming behaviors
- Time-aware 1-hop neighbor sampling
  - A star subgraph of ever connected nodes



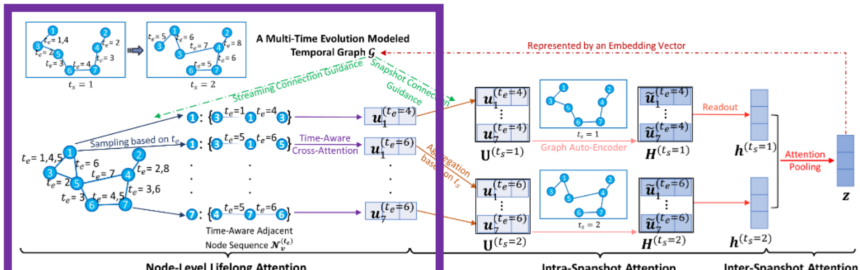
- Node feature concatenates time kernel



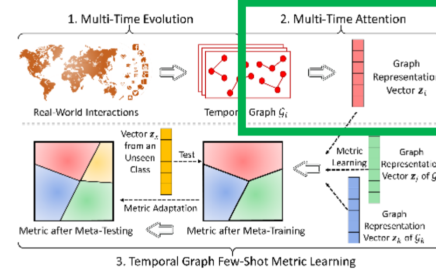
- Self-attention layer



  - Stacking cross-attention layers for attending multi-hop neighbors



Intra-Snapshot Attention



- Adding episodic constraints on node streaming behaviors
  - Snapshot reconstruction loss

topology of snapshot $S^{(t_s)}$

node streaming embedding matrix **reformatted from t_e to t_s** according to our Multi-Time Evolution model

$$\mathcal{L}_{rec}(\mathbf{A}^{(t_s)}, \mathbf{U}^{(t_s)}) = \|\mathbf{A}^{(t_s)} - \hat{\mathbf{A}}^{(t_s)}\|_F$$

Frobenius norm

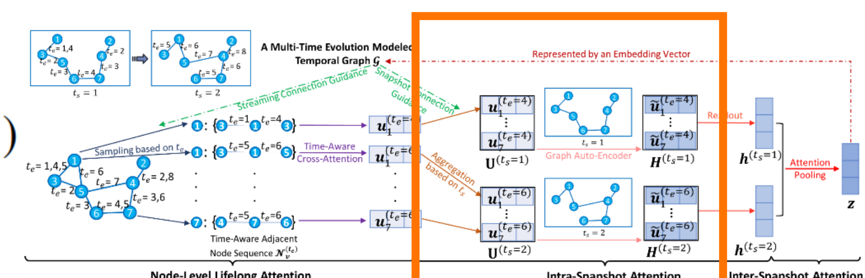
reconstructed topology

$$\hat{\mathbf{A}}^{(t_s)} = \sigma(\mathbf{H}^{(t_s)}\mathbf{H}^{(t_s)\top})$$

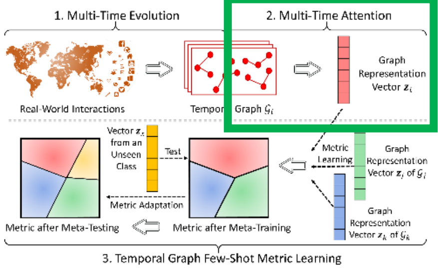
$$\mathbf{H}^{(t_s)} = \mathrm{GNN}_{enc}(\mathbf{A}^{(t_s)}, \mathbf{U}^{(t_s)})$$

  - Snapshot representation

$$\mathbf{h}^{(t_s)} = \mathrm{Readout}(\mathbf{H}^{(t_s)}(v,:) \mid v \in \{1,\ldots,|V^{(t_s)}|\})$$



Inter-Snapshot Attention



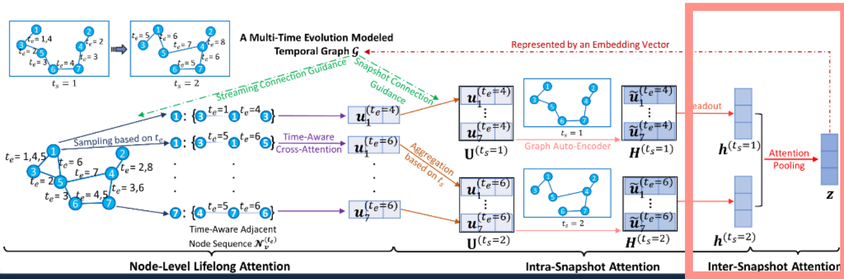
- Discovering class-distinctive episodes
  - If a snapshot is shared by different class temporal graphs, then decrease its weight for graph-level representation

$$\alpha^{(t_s)} = \mathrm{softmax}(\mathbf{w}^{(t_s)\top}\mathbf{h}^{(t_s)})$$

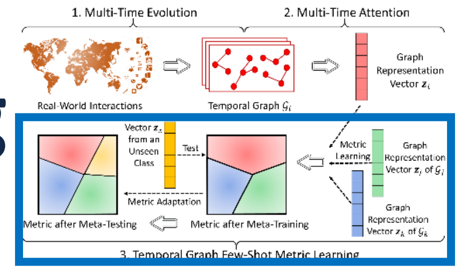
attention pooling weight   snapshot representation

  - Temporal graph representation

$$\mathbf{z} = \sum_{t_s=1}^{T_s}(\alpha^{(t_s)}\mathbf{h}^{(t_s)})$$



Temporal Graph Few-Shot Learning



- Meta-training process procedure
  - Split $D = \{(G_1, y_1), \ldots, (G_n, y_n)\}$ into D^{train} for meta-training and D^{test} for meta-testing
  - Shuffle D^{train} into graph metric learning tasks $\mathcal{T}_i$ (N-way K-shot temporal graph classification task).
  - In each $\mathcal{T}_i$, train θ_i on $D^{train}_{support}$ to classify among D^{train}_{query}
  - Tailor each θ_i to combine the meta-learner Θ

- Meta-training loss function: $\mathcal{L}_{cls} + \gamma \mathcal{L}_{rec}$

reconstruction loss

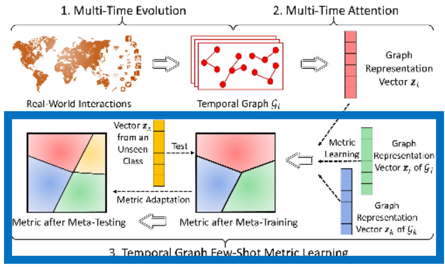
classification loss

$$\mathcal{L}_{cls} = -\sum_{j}^{C_k} \log \frac{\exp(-dist(\mathbf{z}_j, \mathbf{p}_k))}{\sum_{\bar{k}} \exp(-dist(\mathbf{z}_j, \mathbf{p}_{\bar{k}}))}$$

class prototype constructed in $D^{train}_{support}$

$\mathcal{G}_j \in \mathcal{D}^{train}_{query}$ and $y_j = k$

Temporal Graph Few-Shot Learning



- Meta-training process procedure
  - Split $D = \{(G_1, y_1), \ldots, (G_n, y_n)\}$ into D^{train} for meta-training and D^{test} for meta-testing
  - Shuffle D^{train} into graph metric learning tasks $\mathcal{T}_i$ (N-way K-shot temporal graph classification task).
  - In each $\mathcal{T}_i$, train θ_i on $D^{train}_{support}$ to classify among D^{train}_{query}
  - Tailor each θ_i to combine the meta-learner Θ

- Meta-testing process procedure
  - Similar with meta-training, but just fine-tune Θ on $D^{test}_{support}$, then directly report performance on D^{test}_{query}

Roadmap

- Motivation
- Proposed Temp-GFSM Framework
- Experiments
- Conclusion

Effectiveness: Temporal Graph Classification

New Data

- Dataset #1: Dynamic protein-interaction network [13]
  - 12 different classes from 132 graphs
  - Task: use majority classes (e.g., 8) to accurately classify new arrival classes (e.g., 4)

- Performance

Method \ Few-shot Setting		3 way - 5 shot	3 way - 3 shot	3 way - 2 shot	3 way - 1 shot
Graph Kernel	Weisfeiler-Lehman Opt + kNN	0.6833 ± 0.1486	0.5722 ± 0.1554	0.4958 ± 0.1843	0.3417 ± 0.1371
	Neighborhood Hash + kNN	0.6833 ± 0.1858	0.5972 ± 0.1222	0.5833 ± 0.2050	0.5500 ± 0.2082
	Shortest Path + kNN	0.5433 ± 0.1988	0.5306 ± 0.0728	0.5292 ± 0.0946	0.2333 ± 0.1217
Graph Metric Learning	GL2Vec + kNN	0.0717 ± 0.0900	0.0917 ± 0.0793	0.0333 ± 0.0471	0.0333 ± 0.0667
	Graph2Vec + kNN [2]	–	–	–	–
	TGAT + kNN	0.1200 ± 0.0960	0.1250 ± 0.0793	0.1208 ± 0.0865	0.0917 ± 0.0319
	CAW + kNN	0.0400 ± 0.0362	0.0435 ± 0.0441	0.0569 ± 0.0370	0.0528 ± 0.0210
	tdGraphEmbed + kNN	0.2167 ± 0.1736	0.1056 ± 0.0814	0.1500 ± 0.1800	0.0750 ± 0.0877
	GL2Vec + Protonet	0.7100 ± 0.0361	0.6625 ± 0.0407	0.6075 ± 0.0496	0.5750 ± 0.0537
	Graph2Vec + ProtoNet	0.3792 ± 0.0459	0.3958 ± 0.0731	0.3958 ± 0.0241	0.3958 ± 0.0798
	TGAT + ProtoNet	0.2417 ± 0.0500	0.3083 ± 0.0739	0.2917 ± 0.1167	0.2417 ± 0.0319
	CAW + ProtoNet	0.1496 ± 0.0104	0.2113 ± 0.0110	0.2404 ± 0.0117	0.2842 ± 0.0044
	tdGraphEmbed + ProtoNet	0.6562 ± 0.1882	0.6791 ± 0.1141	0.6271 ± 0.1159	0.4229 ± 0.0463
	Temp-GFSM (Ours)	**0.7292 ± 0.0682**	**0.7917 ± 0.1278**	**0.7062 ± 0.0762**	**0.6833 ± 0.0589**

[13] Fu et al., DPPIN: A biological repository of dynamic protein-protein interaction network data. CoRR, 2021

Effectiveness: Temporal Graph Classification

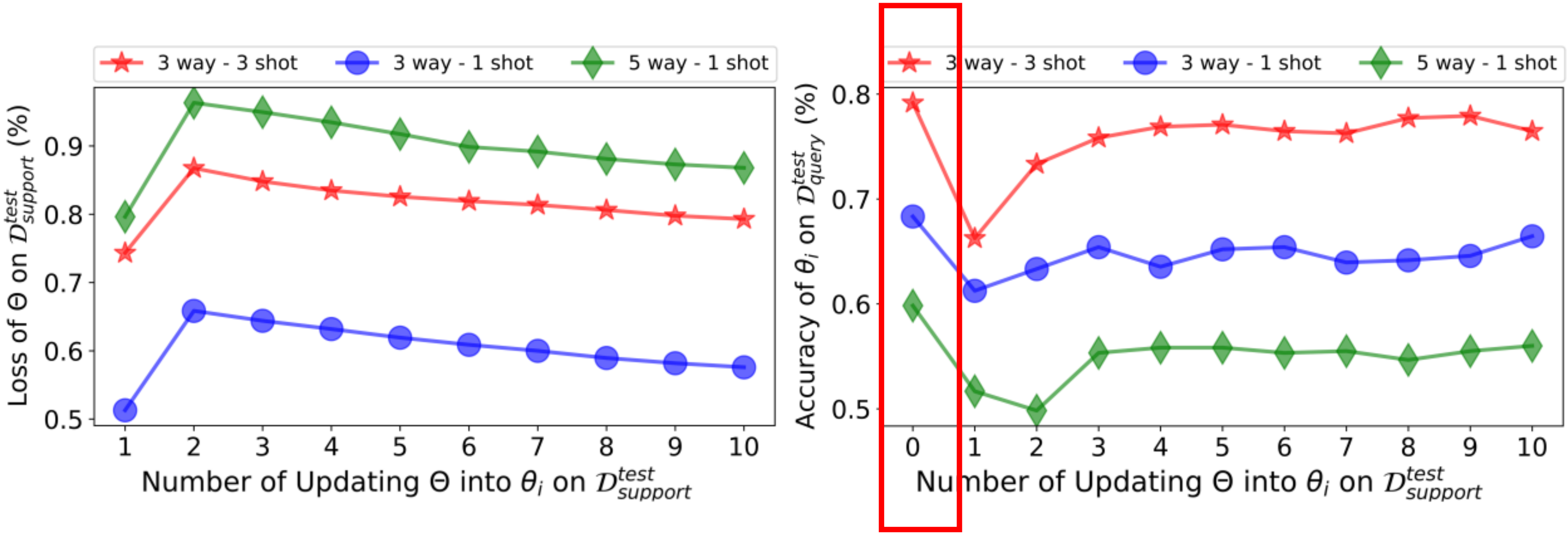
- Dataset #2: Human-contact network
  - 12 different classes (6 online, 6 offline) from 2600 graphs (2123 online, 477 offline)
  - Task: investigate whether online networks could provide knowledge to classify offline networks

- Performance

Method \ Few-shot Setting		3 way - 5 shot	3 way - 3 shot	3 way - 2 shot	3 way - 1 shot
Graph Kernel	Weisfeiler-Lehman Opt + kNN	0.3631 ± 0.0298	0.2941 ± 0.1023	0.2700 ± 0.1105	0.2133 ± 0.0388
	Neighborhood Hash + kNN	0.3938 ± 0.0371	0.3185 ± 0.1030	0.2178 ± 0.1500	0.3022 ± 0.1137
	Shortest Path + kNN	0.3996 ± 0.0317	0.3296 ± 0.1413	0.3556 ± 0.1139	0.3844 ± 0.0420
Graph Metric Learning	GL2Vec + kNN	0.2716 ± 0.0839	0.2637 ± 0.0482	0.1711 ± 0.0700	0.0000 ± 0.0000
	Graph2Vec + kNN	0.3360 ± 0.0352	0.3756 ± 0.0241	0.2400 ± 0.0149	0.1933 ± 0.0723
	TGAT + kNN	0.0289 ± 0.0096	0.0407 ± 0.0123	0.0333 ± 0.0068	0.0200 ± 0.0199
	CAW + kNN	0.0284 ± 0.0106	0.0378 ± 0.0178	0.0322 ± 0.0099	0.0333 ± 0.0192
	tdGraphEmbed + kNN	0.3600 ± 0.0208	0.3000 ± 0.1310	0.2767 ± 0.1185	0.2267 ± 0.0268
	GL2Vec + Protonet	0.3400 ± 0.0306	0.3822 ± 0.0290	0.2633 ± 0.0606	0.1933 ± 0.0723
	Graph2Vec + Protonet	0.3573 ± 0.0203	0.3711 ± 0.0279	0.2467 ± 0.0380	0.1933 ± 0.0723
	TGAT + ProtoNet	0.3227 ± 0.0171	0.3293 ± 0.0156	0.3243 ± 0.0110	0.3363 ± 0.0010
	CAW + ProtoNet	0.3340 ± 0.0113	0.3333 ± 0.0229	0.3380 ± 0.0155	0.3270 ± 0.0189
	tdGraphEmbed + ProtoNet	0.5083 ± 0.0121	0.4523 ± 0.0353	0.4670 ± 0.0199	0.3973 ± 0.0164
	Temp-GFSM (Ours)	**0.6161 ± 0.0139**	**0.5931 ± 0.0148**	**0.6074 ± 0.0164**	**0.5605 ± 0.0201**

Efficiency and Robustness

- Convergence Speed

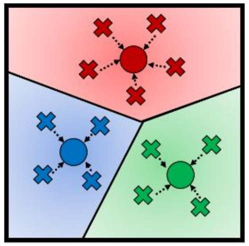


  - With the bi-level optimization, our Temp-GFSM could achieve fast adaption (e.g., 3-4 update iterations)
  - Interestingly, our Temp-GFSM could perform in the zero-shot setting due to the prototype construction during the meta-training

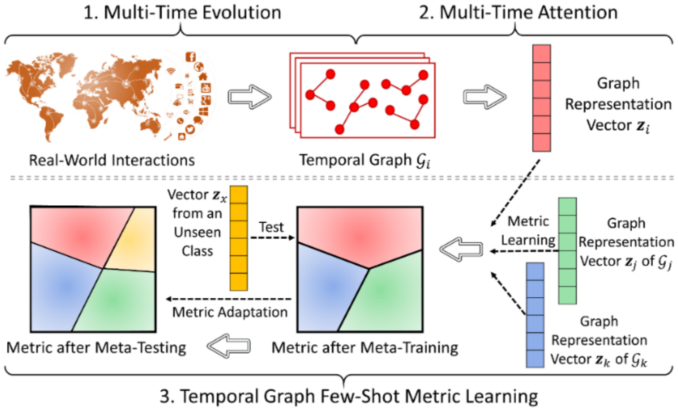
Roadmap

- Motivation
- Proposed Temp-GFSM Framework
- Experiments
- **Conclusion**

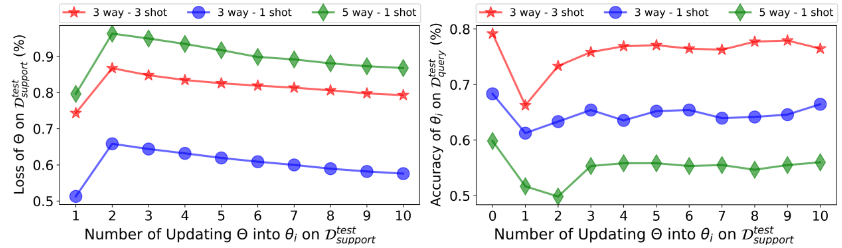
Conclusion



- Problem: Metric Learning over Temporal Graphs
- Algorithm: Temp-GFSM
    - Multi-Time Evolution
    - Multi-Time Attention
        - Node-level lifelong attention
        - Intra-snapshot attention
        - Inter-snapshot attention
    - Temporal graph few-shot learning via bi-level optimization



- Evaluation: Temporal Graph Classification
    - Effectiveness
    - Efficiency and Robustness
    - Ablation Studies
    - Parameter Sensitivity



Thanks!

Dongqi Fu*
(UIUC)

Liri Fang*
(UIUC)

Ross Maciejewski
(ASU)

Vetle I. Torvik
(UIUC)

Jingrui He
(UIUC)

Please refer to our paper and code at
https://github.com/DongqiFu/Temp-GFSM

NSF
U.S. Department of Homeland Security
Homeland Security
NIH National Institutes of Health
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN