MAGIC

MAchine learning for Graph Intelligence Computing

AAAI

Association for the Advancement of Artificial Intelligence

DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning with Selective State Space Models

Haonan Yuan, Qingyun Sun, Zhaonan Wang, Xingcheng Fu, Cheng Ji, Yongjian Wang, Bo Jin, Jianxin Li*

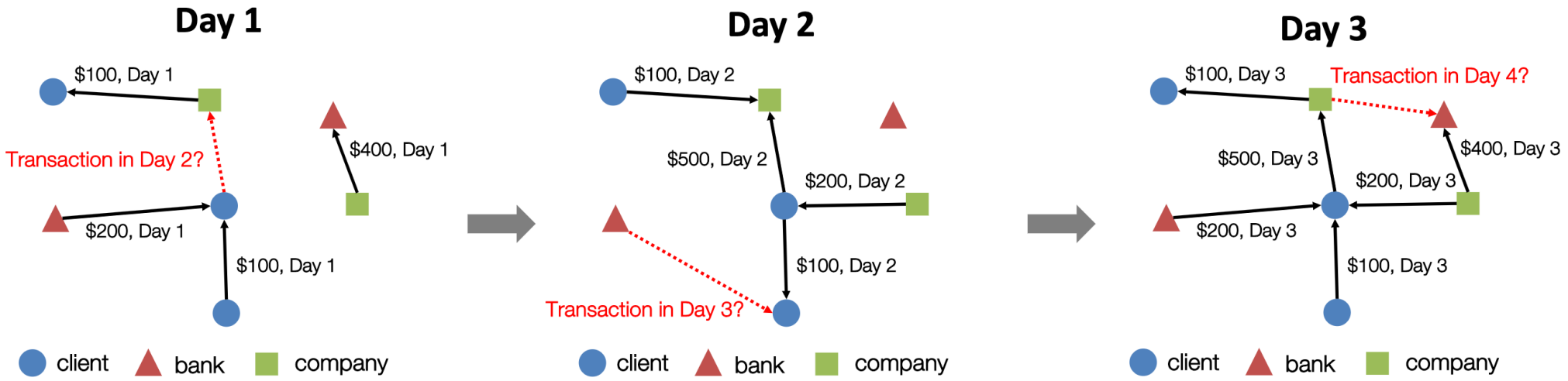
Email: yuanhn@buaa.edu.cn

Paper

Code

Dynamic Graphs: Core of Real-World Data Structures

- Dynamic Graph Neural Networks (DGNNs) have been proposed to tackle highly complex structural and temporal information

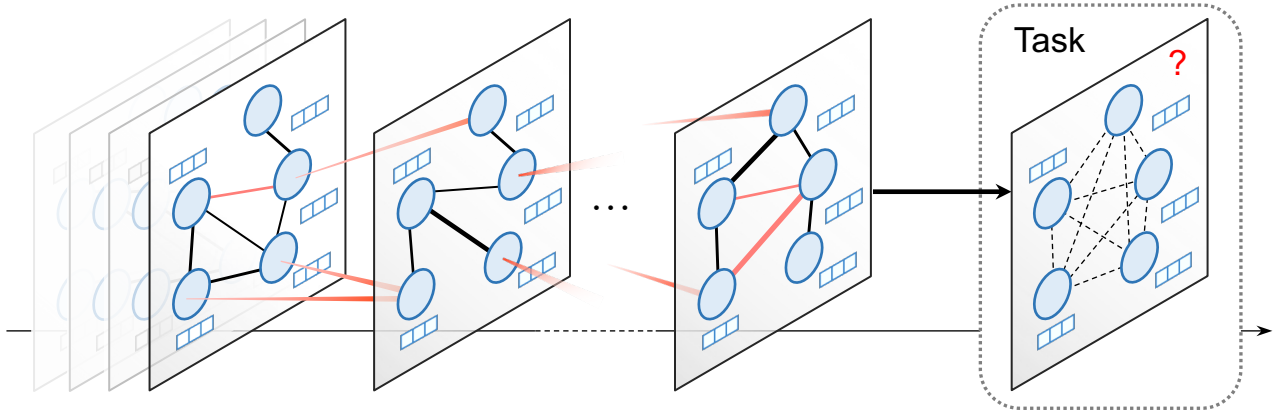


- **Downstream Task:** future link prediction on discrete dynamic graph (fixed nodes, varying links)
  Given $\mathbf{DG} = \{\mathcal{G}^t\}_{t=1}^{T}$, the future link prediction aims to train a model $f_{\boldsymbol{\theta}} : \mathcal{V} \times \mathcal{V} \mapsto \{0,1\}^{N \times N}$ that predicts edges at $T+1$ given historical graphs $\mathcal{G}^{1:T}$ and next-step node feature $\mathbf{X}^{T+1}$.

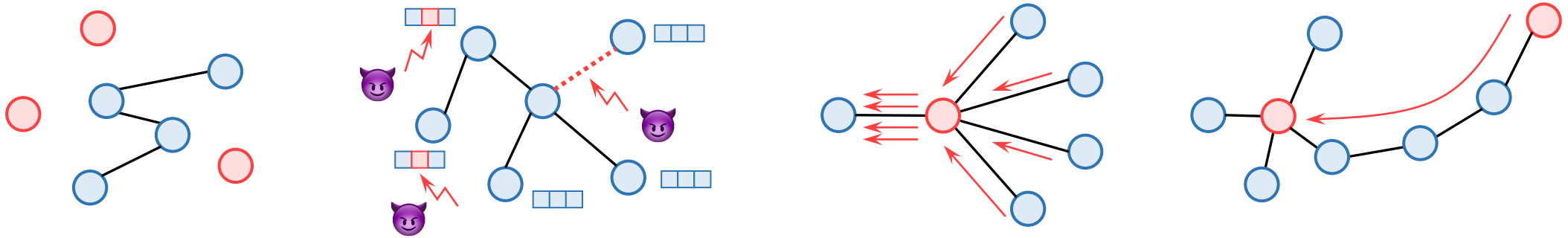
[Picture Credit] You J. *et al*. ROLAND: Graph Learning Framework for Dynamic Graphs. In *KDD 2022*.

Spatio-Temporal Structure Matters for Dynamic Graph Learning

- Structures function as features aggregation paths between and within graph snapshots

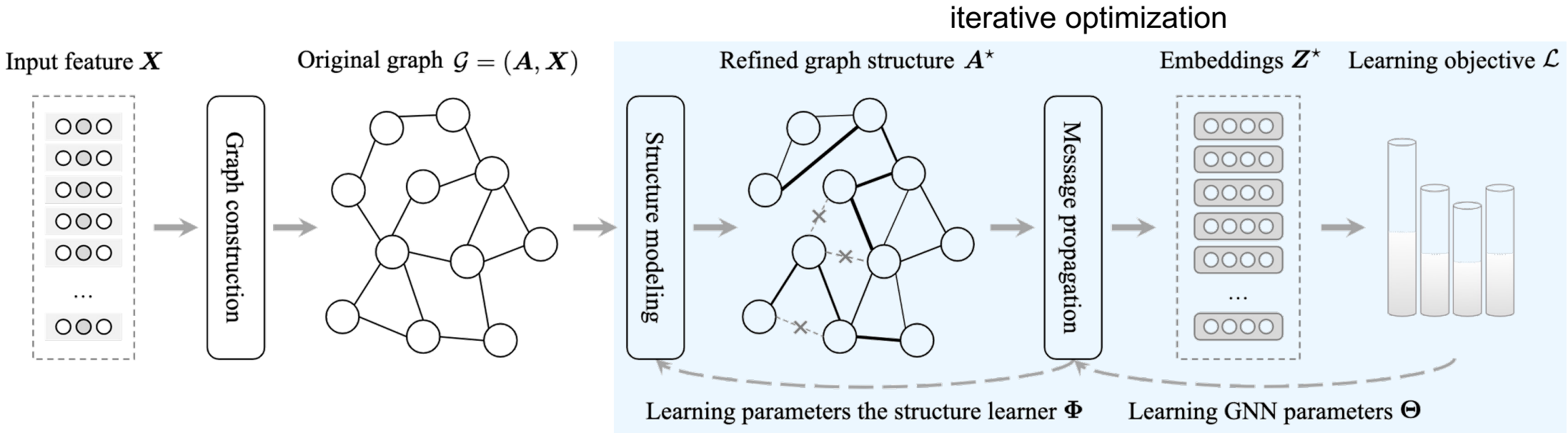


- Common scenarios when DGNNs show deficient performance:



Incompleteness | Noise (Attack) | Over-Squashing | Under-Reaching

One Possible Solution: Graph Structure Learning (GSL)



- GSL aims to jointly learn an optimized structure and corresponding graph representations [1].
- Purified structures are proven to boost robust representations for downstream graph tasks.

Question: Does Dynamic GSL (DGSL) see like Static GSL counterpart?

[1] Zhu Y. *et al*. Deep Graph Structure Learning for Robust Representations: A Survey. In *arXiv 2021*.

Extending to DGSL: Newborn Key Challenges

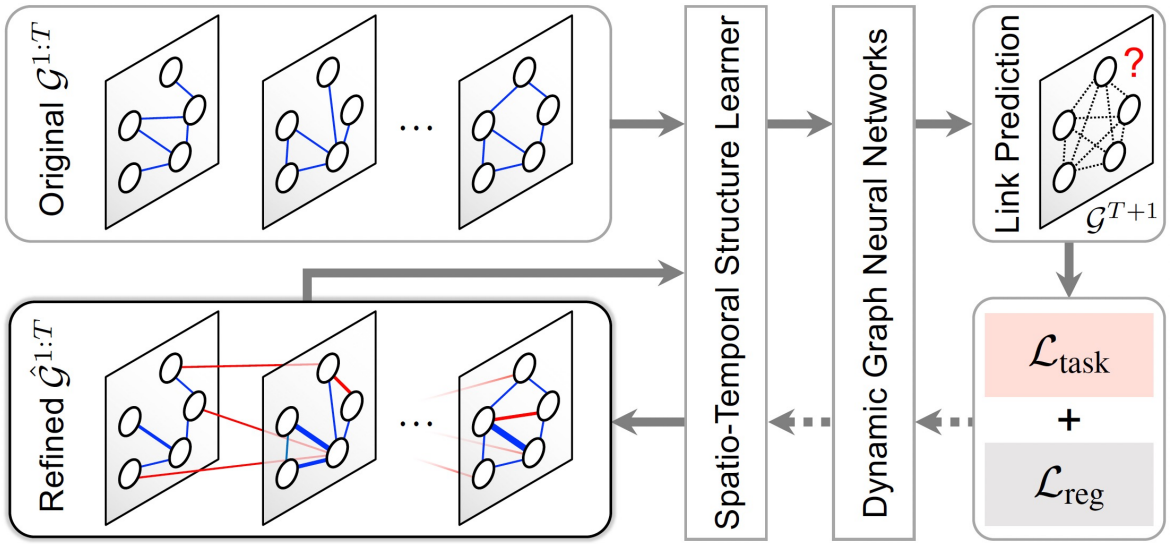


Figure 1: A general paradigm of DGSL.

Challenge 1: Unacceptable $\mathcal{O}(n^2)$ Complexity

Spatial Node-Pair Structure Attention: $\mathcal{O}(N^2)$

Temporal Node-Pair Structure Attention: $\mathcal{O}(N^{(t-1)}N^{(t)})$

Temporal Snapshot-Pair Attention: $\mathcal{O}(T^2)$

Spatio-temporal structure optimization face $\mathcal{O}(T^2N^2)$ complexity, hinders scalability for large-scale graphs.

Extending to DGSL: Newborn Key Challenges

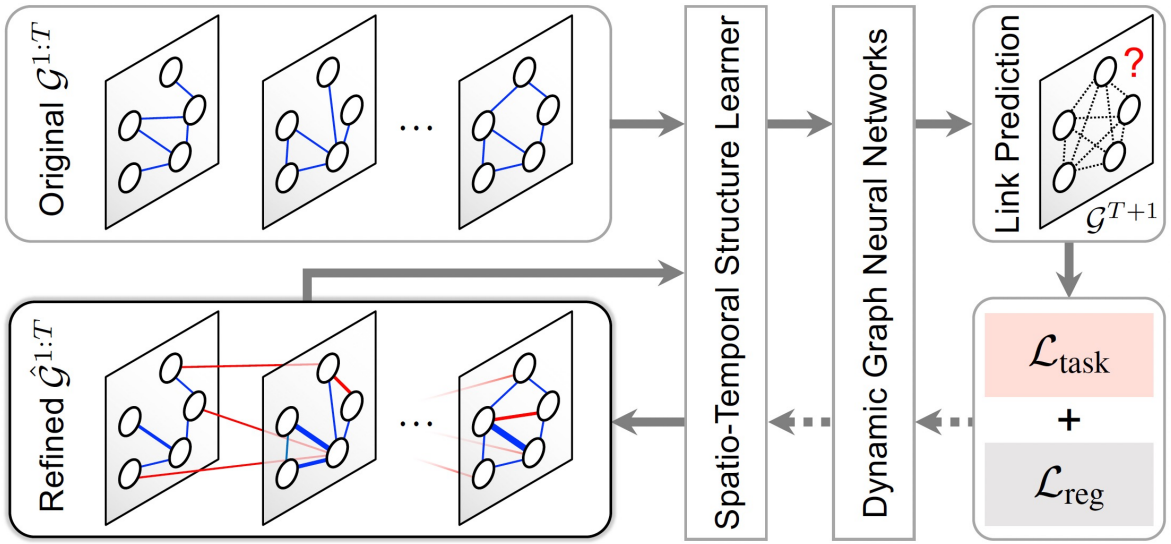
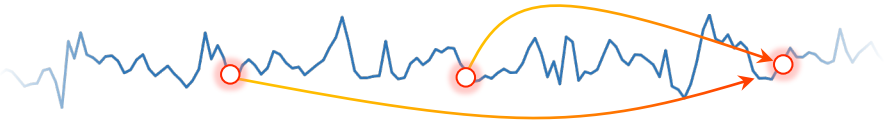


Figure 1: A general paradigm of DGSL.

- **Challenge 1: Unacceptable $\mathcal{O}(n^2)$ Complexity**
- **Challenge 2: Lack of Long-range Dependency**



Often emphasize on local structures, overlooking long-range dependencies that indicate predictive patterns.

Extending to DGSL: Newborn Key Challenges

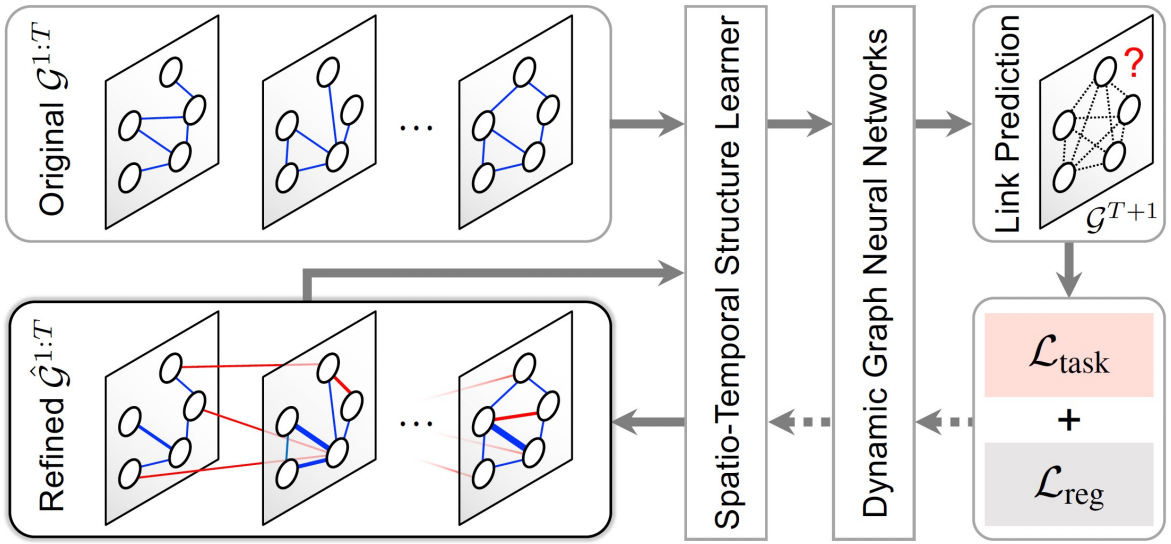
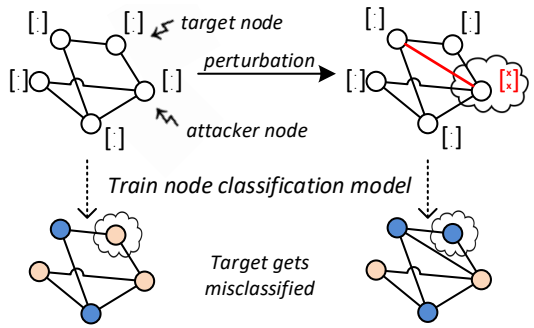
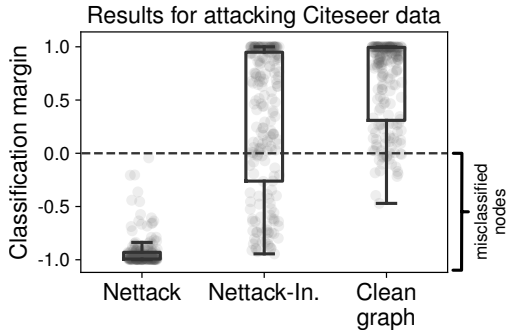


Figure 1: A general paradigm of DGSL.

- **Challenge 1: Unacceptable $\mathcal{O}(n^2)$ Complexity**
- **Challenge 2: Lack of Long-range Dependency**
- **Challenge 3: Undefendable Targeted Attacks**





Dynamic graphs are increasingly vulnerable to targeted adversarial attacks, making defense techs insufficient.

[Picture Credit] Zügner *et al*. Adversarial Attacks on Neural Networks for Graph Data. In *KDD 2018*.

Extending to DGSL: Newborn Key Challenges

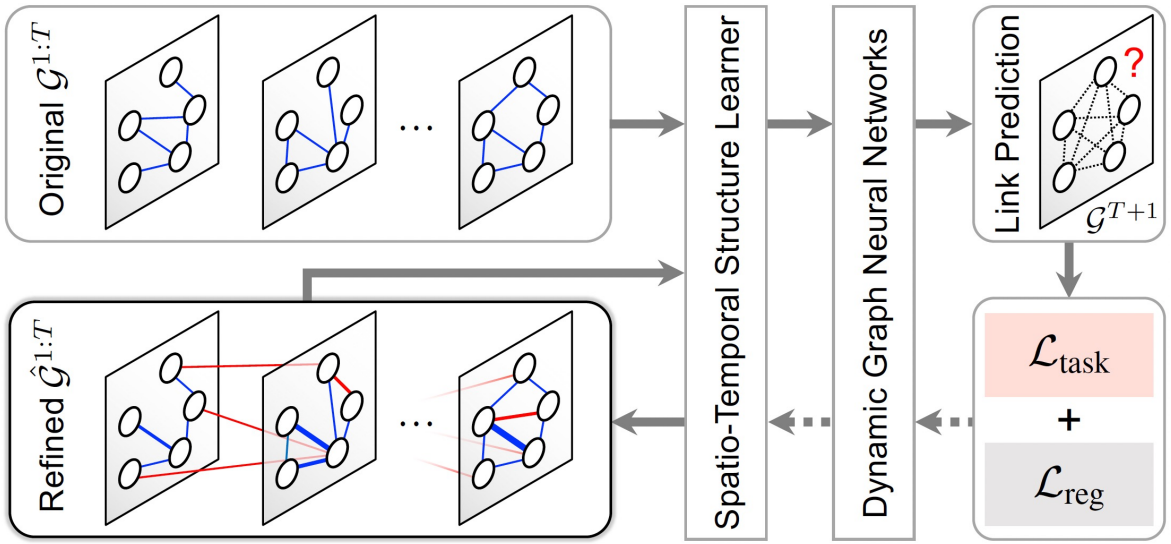
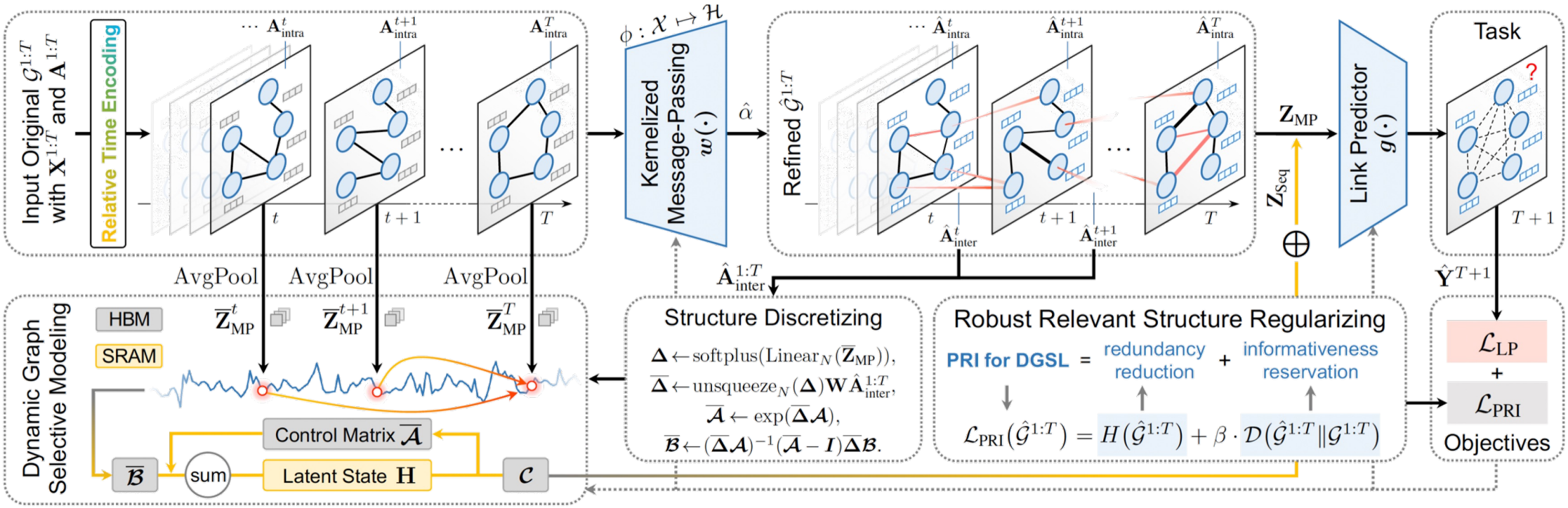


Figure 1: A general paradigm of DGSL.

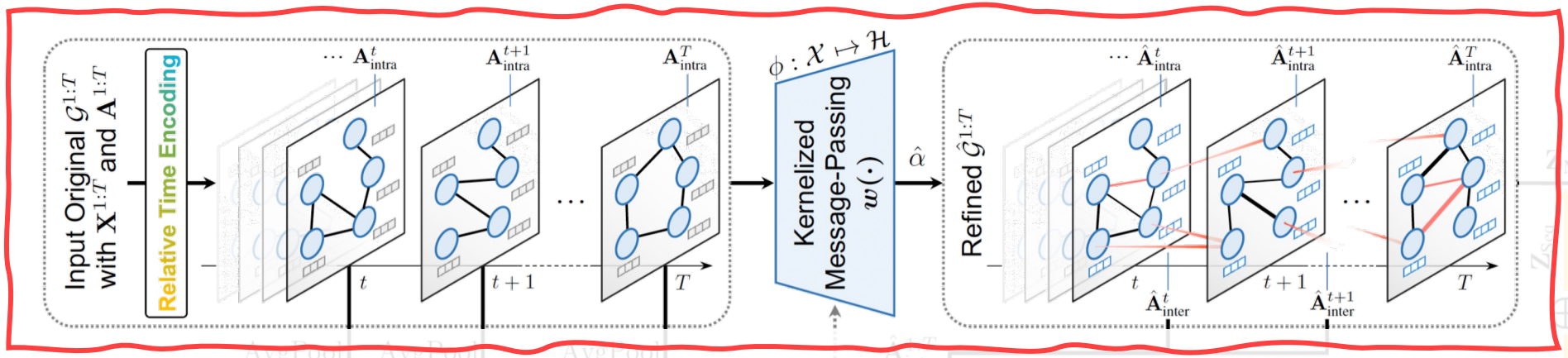
- **Challenge 1: Unacceptable $\mathcal{O}(n^2)$ Complexity**
  Spatio-temporal structure optimization face $\mathcal{O}(T^2N^2)$ complexity, hinders scalability for large-scale graphs.
- **Challenge 2: Lack of Long-range Dependency**
  Often emphasize on local structures, overlooking long-range dependencies that indicate predictive patterns.
- **Challenge 3: Undefendable Targeted Attacks**
  Dynamic graphs are increasingly vulnerable to targeted adversarial attacks, making defense techs insufficient.

Research Question: How to perform **robust** and **efficient DGSL** against random and targeted adversarial attacks?

DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning



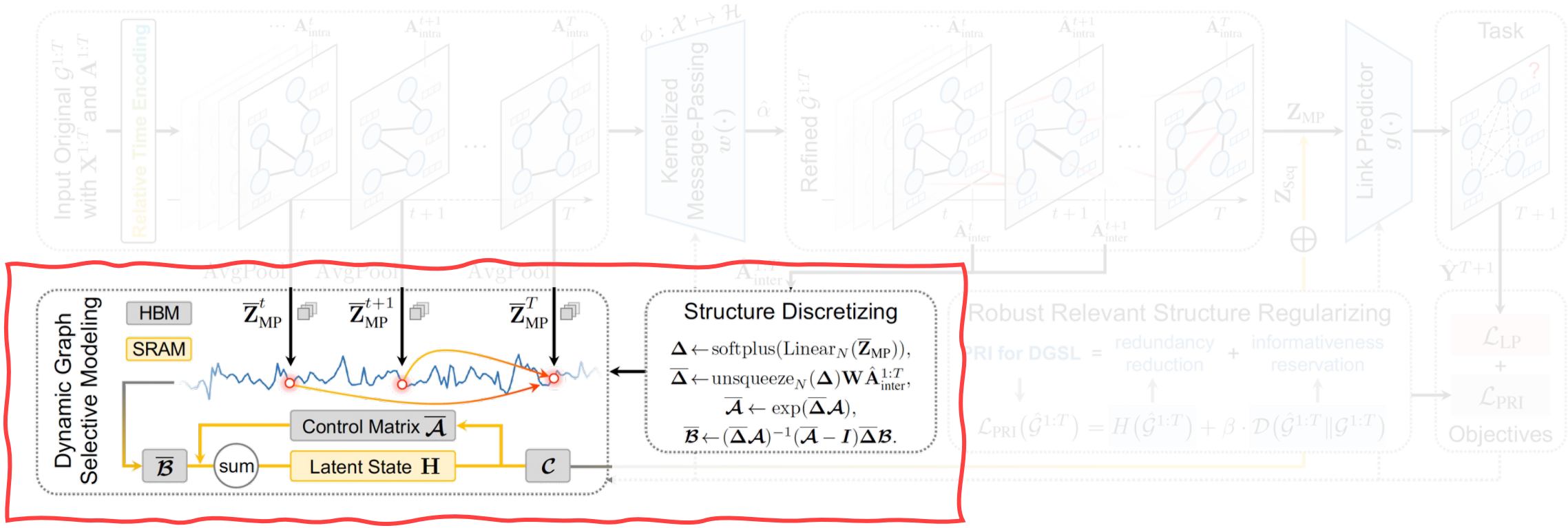
DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning



(a) Kernelized Message-Passing for Efficient Dynamic Graph Structure Learning

(▷ Challenge 1)

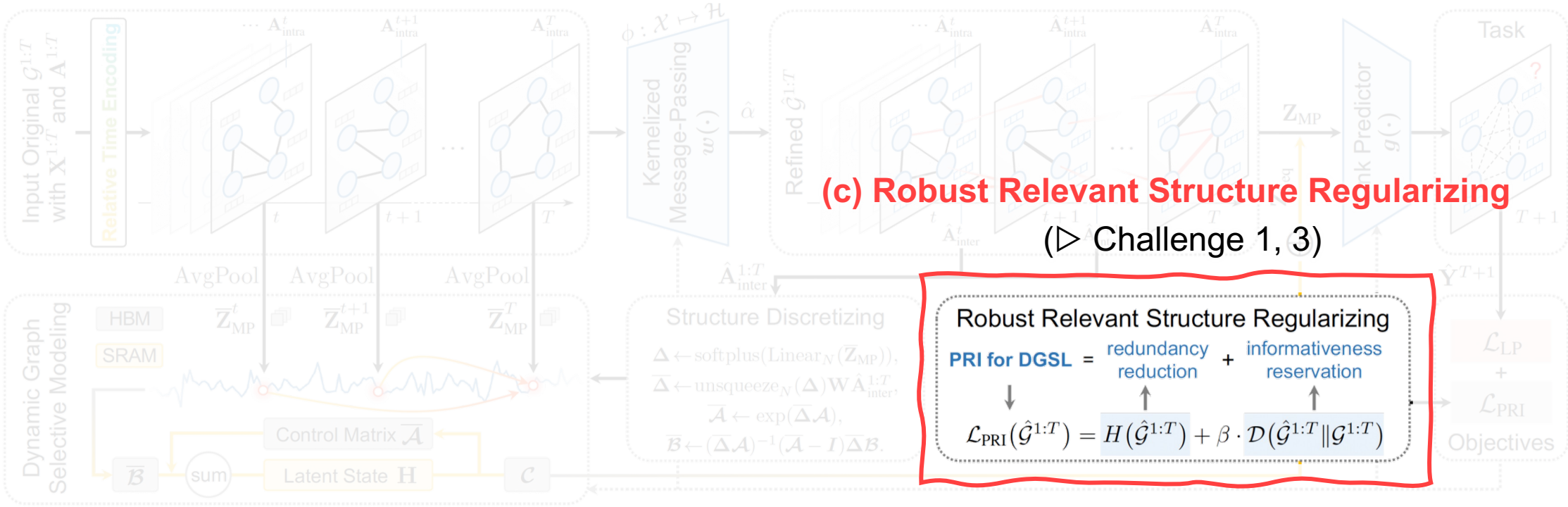
DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning



(b) Long-range Dependencies Selective Modeling

(▷ Challenge 1, 2)

DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning



Kernelized Spatio-Temporal Message Passing ▷

- **[Assumption 1] Dynamic Graph Markov Dependence**

  Assume $\mathbf{DG} = \{\mathcal{G}^t\}_{t=1}^T$ *follows the Markov Chain:* $\langle \mathcal{G}^1 \to \cdots \to \mathcal{G}^T \rangle$. *Given graph* $\mathcal{G}^t$ *at present, the next-step graph* $\mathcal{G}^{t+1}$ *is conditionally independent of the past* $\mathcal{G}^{<t}$, *i.e.,*

$$\mathbb{P}(\mathcal{G}^{t+1} \mid \mathcal{G}^{1:t}) = \mathbb{P}(\mathcal{G}^{t+1} \mid \mathcal{G}^t).$$

- **Attentive Aggregation:** $\mathbf{z}_u^{t(l+1)} = \sum_v \hat{\alpha}_{uv}^{t-1:t(l)} (\mathbf{W}\mathbf{z}_v^{\cdot(l)}), \text{for all } v \in \mathcal{N}(u)^{t-1:t}.$

- **Spatio-temporal Weights:** $\hat{\alpha}_{uv}^{t-1:t(l)} = \dfrac{k(\mathbf{W}\mathbf{z}_u^{t(l)}, \mathbf{W}\mathbf{z}_v^{\cdot(l)})}{\sum_m k(\mathbf{W}\mathbf{z}_u^{t(l)}, \mathbf{W}\mathbf{z}_m^{\cdot(l)})}.$ $[\mathcal{O}(T^2N^2)]$

- **Mercer's Theorem:** $k(\mathbf{x}_1, \mathbf{x}_2)_{\mathcal{X}} = \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle_{\mathcal{H}} \doteq \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2)$ if $k(\cdot,\cdot)$ is positive and definite.

- **Question:** How to find $\phi(\cdot)$? How does it decrease complexity?

Kernelized Spatio-Temporal Message Passing

Kernelized Aggregation:

$$\mathbf{z}_u^{t(l+1)} = \sum_v \frac{k(\mathbf{W}\mathbf{z}_u^{t(l)}, \mathbf{W}\mathbf{z}_v^{\cdot(l)})}{\sum_m k(\mathbf{W}\mathbf{z}_u^{t(l)}, \mathbf{W}\mathbf{z}_m^{\cdot(l)})} \cdot \mathbf{W}\mathbf{z}_v^{\cdot(l)}$$

$$= \sum_v \frac{\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \phi(\mathbf{W}\mathbf{z}_v^{\cdot(l)})}{\sum_m \phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \phi(\mathbf{W}\mathbf{z}_m^{\cdot(l)})} \cdot \mathbf{W}\mathbf{z}_v^{\cdot(l)}$$

$$= \frac{\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \sum_v \phi(\mathbf{W}\mathbf{z}_v^{\cdot(l)}) \mathbf{W}\mathbf{z}_v^{\cdot(l)\top}}{\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \sum_m \phi(\mathbf{W}\mathbf{z}_m^{\cdot(l)})} \cdot \quad [\mathcal{O}(TN)]$$

irreducible

compute once and stored

Positive Random Feature:

$$\phi(\mathbf{x}) = \sum_{i=1}^{m} \frac{1}{\sqrt{m}} \exp\left(\boldsymbol{\omega}_i^\top \mathbf{x} - \frac{\|\mathbf{x}\|_2^2}{2}\right).$$

Structure Query:

$$\hat{\mathbf{A}}_{\text{intra}}^t = \{\hat{\alpha}_{uv}^{t-1:t}\}^{N\times N}, \text{where } u, v \in \mathcal{V}^t,$$

$$\hat{\mathbf{A}}_{\text{inter}}^t = \{\hat{\alpha}_{uv}^{t-1:t}\}^{N\times N}, \text{where } u \in \mathcal{V}^t, v \in \mathcal{V}^{t-1}.$$

Long-range Dependencies Selective Modeling ▷

- **Dynamic Graph System Modeling:** $\mathbf{h}'(t) = \boldsymbol{\mathcal{A}}\mathbf{h}(t) + \boldsymbol{\mathcal{B}}\mathbf{x}(t),\ \mathbf{y}(t) = \boldsymbol{\mathcal{C}}\mathbf{h}(t).$

  how current state evolves (random initialized)

  how input influences state

  how current state translate to output

- **Selective Parameterizing:** (Attention)

$$\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}} \in \mathbb{R}^{B\times T\times D} \leftarrow \mathrm{Linear}_D(\overline{\mathbf{Z}}_{\mathrm{MP}}).$$
$$\boldsymbol{\Delta} \in \mathbb{R}^{B\times T\times N} \leftarrow \mathrm{softplus}(\mathrm{Linear}_N(\overline{\mathbf{Z}}_{\mathrm{MP}})),$$
$$\overline{\boldsymbol{\Delta}} \in \mathbb{R}^{B\times T\times N} \leftarrow \mathrm{unsqueeze}_N(\boldsymbol{\Delta}) \cdot \mathbf{W}\hat{\mathbf{A}}^{1:T}_{\mathrm{inter}}.$$

- **System Discretizing with Zero-Order Hold (ZOH) Rule:**

$$\overline{\boldsymbol{\mathcal{A}}} \leftarrow \exp(\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}}), \overline{\boldsymbol{\mathcal{B}}} \leftarrow (\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}})^{-1}(\exp(\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}}) - \boldsymbol{I})\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{B}}.$$

- **Output and Merge:** $\mathbf{H}_t = \overline{\boldsymbol{\mathcal{A}}}\mathbf{H}_{t-1} + \overline{\boldsymbol{\mathcal{B}}}(\overline{\mathbf{Z}}_{\mathrm{MP}})_t,\ (\mathbf{Z}_{\mathrm{Seq}})_t = \boldsymbol{\mathcal{C}}\mathbf{H}_t.$

$$\hat{\mathbf{Z}} = \mathbf{Z}_{\mathrm{MP}} + \lambda \cdot \mathrm{unsqueeze}_{D_0}(\mathbf{Z}_{\mathrm{Seq}}).$$

Robust Relevant Structure Regularizing ▷

[Definition 1] Principle of Relevant Information (PRI) for DGSL

Given dynamic graph $\mathcal{G}^{1:T}$, the Principle of Relevant Information for DGSL aims to regularize the refined graph $\hat{\mathcal{G}}^{1:T}$ by,

$$\mathcal{L}_{\text{PRI}}(\hat{\mathcal{G}}^{1:T}) = H(\hat{\mathcal{G}}^{1:T}) + \beta \cdot \mathcal{D}(\hat{\mathcal{G}}^{1:T} \| \mathcal{G}^{1:T}) \triangleq \mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}^{1:T}_{\text{intra}}) + \mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}^{1:T}_{\text{inter}}).$$

redundancy　　informativeness

Derivations:

- **intra-graph** structure regularize: $\mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}^{1:T}_{\text{intra}}) \doteq H(\hat{\mathbf{A}}^{1:T}_{\text{intra}}) + \beta_1 \cdot \mathcal{L}_{\text{edge}}$

$$\mathcal{L}_{\text{edge}} = -\frac{1}{NT} \sum_{t=1}^{T} \sum_{u,v \in \mathcal{E}^t} \frac{1}{d(u)} \log \hat{\alpha}^t_{uv}.$$

- **inter-graph** structure regularize: $\mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}^{1:T}_{\text{intra}}) \doteq H(\ \mathbf{Z}_{\text{Seq}}\ ) + \beta_2 \cdot \mathcal{D}(\ \mathbf{Z}_{\text{Seq}} \| \overline{\mathbf{Z}}_{\text{MP}}\ )$

Optimization and Complexity Analysis

Algorithm 1: Overall training pipeline of DG-Mamba.

Input: Dynamic graph $\mathcal{G}^{1:T}$; Node features $\mathbf{X}^{1:T+1}$; Labels $\mathbf{Y}^{1:T}$ of the link occurrence; Model $f_{\boldsymbol{\theta}} = w \circ g$.

Parameter: Number of training epochs E; Number of layers L; Hyperparameters β_1, β_1, γ, λ and μ.

Output: Refined dynamic graph $\hat{\mathcal{G}}^{1:T}$; Optimized parameter $f_{\boldsymbol{\theta}}^{\star} = w \circ g$; Predicted label $\hat{\mathbf{Y}}^{T+1}$ of the link occurrence at time step $T+1$.

Initialize weights and model parameters $\boldsymbol{\theta}$ randomly;
Initialize: $\mathbf{Z}^{1:T+1(0)} \leftarrow \mathbf{X}^{1:T+1}$;
Relative time encoding: $\mathbf{Z}^{1:T+1(0)} \leftarrow \mathrm{RTE}(\mathbf{Z}^{1:T+1(0)})$;
for epochs $e = 1, 2, \cdots, E$ **do**
  for layer $l = 1, 2, \cdots, L$ **do**
    Kernelized message-passing: $\mathbf{Z}_{\mathrm{MP}} \leftarrow$ Eq. (7);
    Average pooling: $\overline{\mathbf{Z}}_{\mathrm{MP}} \leftarrow \mathrm{AvgPool}(\mathbf{Z}_{\mathrm{MP}})$;
    Structure query: $\hat{\mathbf{A}}_{\mathrm{intra}}^{1:T}, \hat{\mathbf{A}}_{\mathrm{inter}}^{1:T} \leftarrow$ Eq. (9), (10);
    Long-range dependencies selective modeling: $\mathbf{Z}_{\mathrm{Seq}} \leftarrow \mathrm{DGSS}(\overline{\mathbf{Z}}_{\mathrm{MP}}, \hat{\mathbf{A}}_{\mathrm{inter}}^{1:T})$ (see Algorithm 2);
    Node feature update: $\hat{\mathbf{Z}} \leftarrow$ Eq. (17);
  end for
  Predict: $\hat{\mathbf{Y}}^{T+1} = g(\hat{\mathbf{Z}}^{T+1(L)})$;
  $\mathcal{L}_{\mathrm{LP}} \leftarrow \mathrm{CE}(\mathbf{Y}^{T+1}, \hat{\mathbf{Y}}^{T+1})$; $\mathcal{L}_{\mathrm{PRI}}(\hat{\mathcal{G}}^{1:T}) \leftarrow$Eq. (19);
  Calculate the overall loss: $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{LP}} + \mu \cdot \mathcal{L}_{\mathrm{PRI}}(\hat{\mathcal{G}}^{1:T})$;
  Update $\boldsymbol{\theta}$ by minimizing $\mathcal{L}$ and back-propagation.
end for

Algorithm 2: Dynamic Graph Selective Scan (DGSS).

Input: The average-pooled representation $\overline{\mathbf{Z}}_{\mathrm{MP}} \in \mathbb{R}^{B \times T \times N}$; The inter-graph structures $\hat{\mathbf{A}}_{\mathrm{inter}}^{1:T} \in \mathbb{R}^{B \times T \times N \times N}$.

Parameter: The batch size B; The dynamic graph length T; The number of latent states N; The hidden feature dimension D; The weight matrix $\mathbf{W}$.

Output: The temporal semantics and long-range dependencies emerged $\mathbf{Z}_{\mathrm{Seq}} \in \mathbb{R}^{B \times T \times N}$.

Randomly initialize $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N \times D}$;
$\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}} \in \mathbb{R}^{B \times T \times D} \leftarrow \mathrm{Linear}_D(\overline{\mathbf{Z}}_{\mathrm{MP}})$;
$\boldsymbol{\Delta} \in \mathbb{R}^{B \times T \times N} \leftarrow \mathrm{softplus}(\mathrm{Linear}_N(\overline{\mathbf{Z}}_{\mathrm{MP}}))$;
$\overline{\boldsymbol{\Delta}} \in \mathbb{R}^{B \times T \times N} \leftarrow \mathrm{unsqueeze}_N(\boldsymbol{\Delta}) \cdot \mathbf{W}\hat{\mathbf{A}}_{\mathrm{inter}}^{1:T}$;
$\overline{\boldsymbol{\mathcal{A}}} \in \mathbb{R}^{B \times T \times N \times D} \leftarrow \exp(\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}})$;
$\overline{\boldsymbol{\mathcal{B}}} \in \mathbb{R}^{B \times T \times N \times D} \leftarrow (\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}})^{-1}(\exp(\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}}) - \boldsymbol{I})\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{B}}$;
Initialize latent state $\mathbf{H} \in \mathbb{R}^{B \times T \times N \times D}$ as zeros;
Initialize the output $\mathbf{Z}_{\mathrm{Seq}} \in \mathbb{R}^{B \times T \times N}$ as empty set;
for t from 1 to $T-1$ **do**
  $\mathbf{H}_t = \overline{\boldsymbol{\mathcal{A}}}_t \mathbf{H}_{t-1} + \overline{\boldsymbol{\mathcal{B}}}(\overline{\mathbf{Z}}_{\mathrm{MP}})_t$;
  $(\mathbf{Z}_{\mathrm{Seq}})_t = \boldsymbol{\mathcal{C}}_t \mathbf{H}_t$;
end for
return $\mathbf{Z}_{\mathrm{Seq}} \in \mathbb{R}^{B \times T \times N}$.

Computational Complexity: $\mathcal{O}(T(|\mathcal{V}| + |\mathcal{E}|))$.

(in **linear** with the length, averaged #nodes, and #edges)

Dynamic Graph Datasets

- **COLLAB:** Academic citation network.
- **Yelp:** Customer reviews network.
- **ACT:** MOOC actions network.

Dataset	# Node	# Link	# Link Type	Length (Split)	Temporal Granularity
COLLAB	23,035	151,790	5	16 (10/1/5)	year
Yelp	13,095	65,375	5	24 (15/1/8)	month
ACT	20,408	202,339	5	30 (20/2/8)	day

Baselines

- **Static GNNs:** GAE, VGAE
- **Dynamic GNNs:** GCRN, EvolveGCN, DySAT, SpoT-Mamba*
- **DGSL:** RDGSL, TGSL
- **Robust (D)GNNs:** RGCN, WinGNN, DGIB

Adversarial Attack Settings

- **Non-targeted**: for structures, randomly remove links; for node features, add random noise
- **Targeted:** apply NETTACK library to perform both evasion attack and poisoning attack
  - **evasion:** train on clean datasets and perform attacking on each graph snapshot in testing.
  - **poisoning:** attack the whole dataset before model training and testing.

Against Non-Targeted (random) Adversarial Attacks

Dataset	COLLAB					Yelp					ACT				
Model	Clean	Structure Attack	Feature Attack			Clean	Structure Attack	Feature Attack			Clean	Structure Attack	Feature Attack		
			$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 1.5$			$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 1.5$			$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 1.5$
GAE	77.15±0.5	74.04±0.8	50.59±0.8	44.66±0.8	43.12±0.8	70.67±1.1	64.45±5.0	51.05±0.6	45.41±0.6	41.56±0.9	72.31±0.5	60.27±0.4	56.56±0.5	52.52±0.6	50.36±0.9
VGAE	86.47±0.0	74.95±1.2	56.75±0.6	50.39±0.7	48.68±0.7	76.54±0.5	65.33±1.4	55.53±0.7	49.88±0.8	45.08±0.6	79.18±0.5	66.29±1.3	60.67±0.7	57.39±0.8	55.27±1.0
GAT	88.26±0.4	77.29±1.8	58.13±0.9	51.41±0.9	49.77±0.9	77.93±0.1	69.35±1.6	56.72±0.3	52.51±0.5	46.21±0.5	85.07±0.3	77.55±1.2	66.05±0.4	61.85±0.3	59.05±0.3
GCRN	82.78±0.5	69.72±0.5	54.07±0.9	47.78±0.8	46.18±0.9	68.59±1.0	54.68±7.6	52.68±0.6	46.85±0.6	40.45±0.6	76.28±0.5	64.35±1.2	59.48±0.7	54.16±0.6	53.88±0.7
EvolveGCN	86.62±1.0	76.15±0.9	56.82±1.2	50.33±1.0	48.55±1.0	78.21±0.0	53.82±2.0	57.91±0.5	51.82±0.3	45.32±1.0	74.55±0.3	63.17±1.0	61.02±0.5	53.34±0.5	51.62±0.7
DySAT	88.77±0.2	76.59±0.2	58.28±0.3	51.52±0.3	49.32±0.5	78.87±0.6	66.09±1.4	58.46±0.4	52.33±0.7	46.24±0.7	78.52±0.4	66.55±1.2	61.94±0.8	56.98±0.8	54.14±0.7
SpoT-Mamba	84.34±0.4	74.39±0.2	54.76±0.8	48.64±0.9	47.25±0.7	77.01±1.0	60.56±1.2	54.72±0.8	50.11±0.8	44.95±0.8	73.29±1.0	61.27±0.9	59.92±0.7	52.19±0.8	51.33±0.9
RDGSL	82.29±0.5	71.36±0.9	52.33±0.5	48.50±0.7	45.21±0.6	75.92±0.6	58.30±0.9	52.29±0.5	48.66±0.4	44.59±0.5	73.15±0.6	62.45±1.0	60.14±0.6	53.05±0.5	51.07±0.5
TGSL	84.09±0.5	73.66±1.0	55.29±0.4	51.34±0.4	50.28±0.3	76.55±0.4	73.29±1.1	60.21±0.3	51.01±0.3	49.87±0.4	80.53±0.5	70.32±0.9	67.19±0.4	60.27±0.5	58.39±0.5
RGCN	88.21±0.1	78.66±0.7	61.29±0.5	54.29±0.6	52.99±0.6	77.28±0.3	74.29±0.4	59.72±0.3	52.88±0.3	50.40±0.2	87.22±0.2	82.66±0.4	68.51±0.2	62.67±0.2	61.31±0.2
WinGNN	90.33±0.1	82.34±0.6	64.69±0.9	56.87±1.1	54.44±0.6	76.46±1.0	74.59±0.8	60.45±0.4	55.80±1.0	52.73±0.8	90.12±0.4	85.36±0.4	71.60±0.9	65.40±0.3	63.32±0.8
DGIB-Bern	92.17±0.2	83.58±0.1	63.54±0.9	56.92±1.0	56.24±1.0	76.88±0.2	75.61±0.0	**63.91±0.9**	**59.28±0.9**	54.77±1.0	94.49±0.2	87.75±0.1	73.05±0.9	68.49±0.9	66.27±0.9
DGIB-Cat	92.68±0.1	84.16±0.1	63.99±0.5	57.76±0.8	55.63±1.0	79.53±0.2	**77.72±0.1**	61.42±0.9	55.12±0.7	51.90±0.9	94.89±0.2	88.27±0.2	73.92±0.8	68.88±0.9	65.99±0.7
DG-Mamba	**93.60±0.3**	**92.60±0.3**	**68.53±1.5**	**60.88±1.0**	**56.95±0.8**	**81.54±0.6**	77.40±0.7	61.82±0.9	57.42±0.6	**55.97±1.2**	**96.67±0.3**	**96.14±0.3**	**79.36±0.8**	**73.76±0.7**	**70.21±0.7**

Table 1: AUC score (% ± standard deviation for five runs) of the future link prediction task on real-world datasets against **non-targeted** (random) adversarial attacks. The best results are shown in **bold** type and the runner-ups are underlined.

Against Targeted Adversarial Attacks

Dataset	Model	Clean	Evasion Attack				Poisoning Attack			
			$n=1$ ($\Delta\%\downarrow$)	$n=2$ ($\Delta\%\downarrow$)	$n=3$ ($\Delta\%\downarrow$)	Avg. $\Delta\%\downarrow$	$n=1$ ($\Delta\%\downarrow$)	$n=2$ ($\Delta\%\downarrow$)	$n=3$ ($\Delta\%\downarrow$)	Avg. $\Delta\%\downarrow$
COLLAB	GAT	88.26±0.4	76.21±0.1 (13.7)	66.56±0.1 (24.6)	57.92±0.1 (34.4)	24.2	66.59±0.5 (24.6)	55.31±0.6 (37.3)	51.34±0.7 (41.8)	34.6
	DySAT	88.77±0.2	77.91±0.1 (12.2)	68.22±0.1 (23.1)	58.82±0.1 (33.7)	23.0	69.02±0.3 (22.2)	57.62±0.3 (35.1)	52.76±0.3 (40.6)	32.6
	SpoT-Mamba	84.34±0.4	71.45±0.2 (15.3)	65.88±0.2 (21.9)	52.14±0.3 (38.2)	25.1	66.45±0.5 (21.2)	55.36±0.9 (34.4)	53.17±0.6 (37.0)	30.8
	TGSL	84.09±0.5	72.09±0.3 (14.3)	65.30±0.2 (22.3)	52.09±0.3 (38.1)	24.9	66.57±0.3 (20.8)	54.21±0.2 (35.5)	55.36±0.3 (34.2)	30.2
	WinGNN	90.33±0.1	79.35±0.2 (12.2)	68.24±0.1 (24.5)	61.07±0.3 (32.4)	23.0	71.53±0.8 (20.8)	61.57±1.1 (31.8)	55.27±1.0 (38.8)	30.5
	DGIB-Cat	92.68±0.1	81.29±0.0 (12.3)	71.32±0.1 (23.0)	62.03±0.1 (33.1)	22.8	72.55±0.2 (21.7)	60.99±0.3 (34.2)	55.62±0.4 (40.0)	32.0
	DG-Mamba	**93.60±0.3**	**81.78±0.6** (12.6)	**80.87±0.6** (13.6)	**68.75±1.3** (26.5)	**17.6**	**79.48±0.2** (15.1)	**67.45±0.1** (27.9)	**64.99±0.6** (30.6)	**24.5**
Yelp	GAT	77.93±0.1	67.96±0.1 (12.8)	59.47±0.1 (23.7)	50.27±0.1 (35.5)	24.0	65.34±0.5 (16.2)	54.51±0.2 (30.1)	50.24±0.4 (35.5)	27.2
	DySAT	78.87±0.6	69.77±0.1 (11.5)	60.66±0.1 (23.1)	52.16±0.1 (33.9)	22.8	66.87±0.6 (15.2)	56.31±0.3 (28.6)	50.44±0.6 (36.0)	26.6
	SpoT-Mamba	77.01±1.0	65.25±0.2 (15.3)	54.33±0.2 (29.5)	47.75±0.2 (38.0)	27.6	64.39±1.0 (16.4)	55.21±0.9 (28.3)	50.33±1.1 (34.6)	26.4
	TGSL	76.55±0.4	65.03±0.3 (15.0)	54.29±0.3 (29.1)	47.81±0.3 (37.5)	27.2	64.08±0.8 (16.3)	56.27±0.6 (26.5)	51.20±0.8 (33.1)	25.3
	WinGNN	76.46±1.0	66.25±1.0 (13.4)	60.22±0.9 (21.2)	51.38±0.8 (32.8)	22.5	67.88±0.9 (11.2)	56.36±0.9 (26.3)	52.74±1.0 (31.0)	22.8
	DGIB-Cat	79.53±0.2	70.17±0.0 (11.8)	62.25±0.1 (21.7)	**52.69±0.1** (33.7)	22.4	67.38±0.3 (15.3)	57.02±0.2 (28.3)	51.39±0.2 (35.4)	26.3
	DG-Mamba	**81.54±0.6**	**70.88±0.3** (13.1)	**69.77±0.5** (14.4)	49.93±0.6 (38.8)	**22.1**	**73.10±0.4** (10.4)	**64.65±0.1** (20.7)	**54.67±0.3** (33.0)	**21.3**
ACT	GAT	85.07±0.3	75.14±0.1 (11.7)	67.25±0.1 (20.9)	59.75±0.1 (29.8)	20.8	71.26±0.9 (16.2)	61.43±1.1 (27.8)	57.35±1.1 (32.6)	25.5
	DySAT	78.52±0.4	70.64±0.1 (10.0)	63.35±0.0 (19.3)	56.36±0.0 (28.2)	19.2	66.21±0.9 (15.7)	56.28±0.9 (28.3)	53.45±1.1 (31.9)	25.3
	SpoT-Mamba	73.29±1.0	65.64±1.1 (10.4)	61.99±0.9 (15.4)	51.08±0.8 (30.3)	18.7	62.89±0.9 (14.9)	58.04±1.3 (20.8)	51.04±1.2 (30.4)	22.0
	TGSL	80.53±0.5	72.26±0.3 (12.8)	67.34±0.3 (16.4)	61.55±0.3 (23.6)	17.6	68.10±0.9 (15.4)	61.07±1.0 (24.2)	59.39±1.0 (26.3)	22.0
	WinGNN	90.12±0.4	80.16±0.4 (11.1)	72.50±0.3 (19.6)	63.21±0.4 (29.9)	20.2	81.26±0.9 (9.8)	67.33±1.1 (25.3)	61.25±1.0 (32.0)	22.4
	DGIB-Cat	94.89±0.2	84.98±0.1 (10.4)	76.78±0.1 (19.1)	**67.69±0.1** (28.7)	19.4	80.16±0.4 (15.5)	68.71±0.5 (27.6)	64.38±0.6 (32.2)	25.1
	DG-Mamba	**96.67±0.3**	**86.62±0.1** (10.4)	**85.58±0.1** (11.5)	67.12±0.5 (30.6)	**17.5**	**85.53±0.6** (11.5)	**75.62±0.2** (21.8)	**65.65±0.5** (32.1)	**21.8**

Table 2: AUC score (% ± standard deviation for five runs) of the future link prediction task on real-world datasets against **targeted** adversarial attacks. The best results are shown in **bold** type and the runner-ups are underlined. "Δ%" indicates the relative performance decrease after targeted adversarial attacks compared to that on the clean datasets.

Scaling Efficiency Analysis

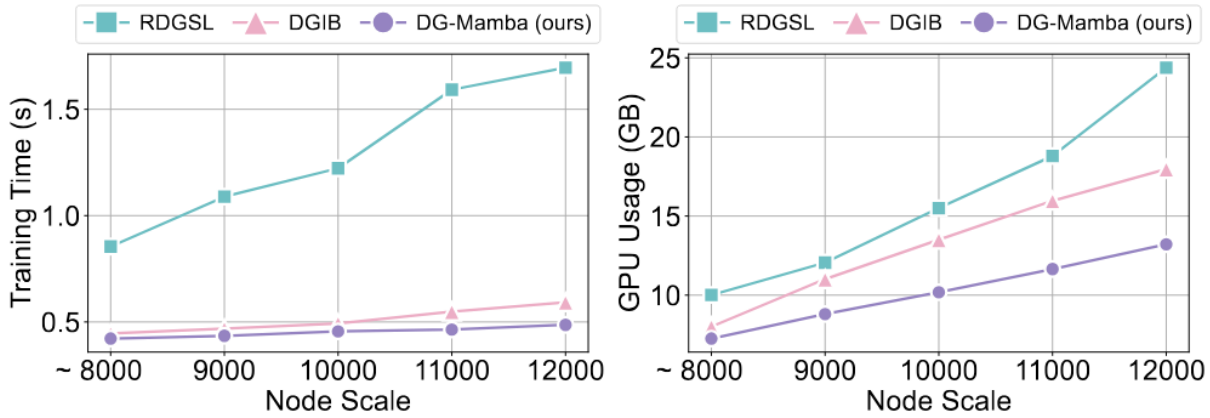


Figure 3: Node scaling efficiency evaluation on Yelp.

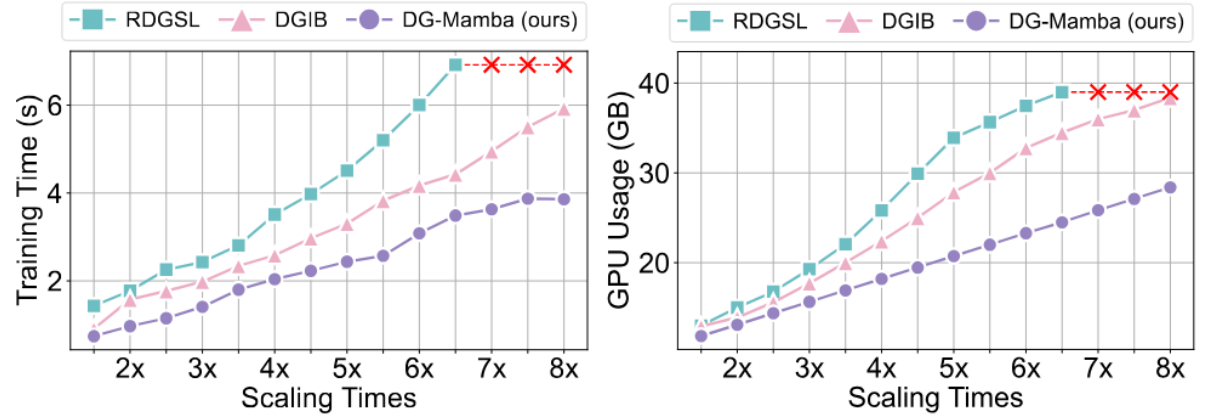


Figure 4: Sequence scaling efficiency evaluation on Yelp.

Ablation Study

- **DG-Mamba (w/o KMP)**: We replace the efficient spatio-temporal kernelized message-passing with the vanilla attention-based message-passing mechanism.
- **DG-Mamba (w/o SM):** We remove the long-range dependencies selective modeling.
- **DG-Mamba (w/o PRI):** We remove Principle of Relevant Information for DGSL regularizing term.

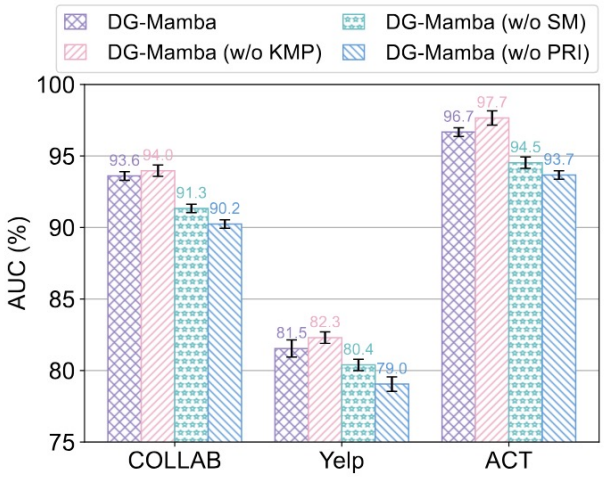
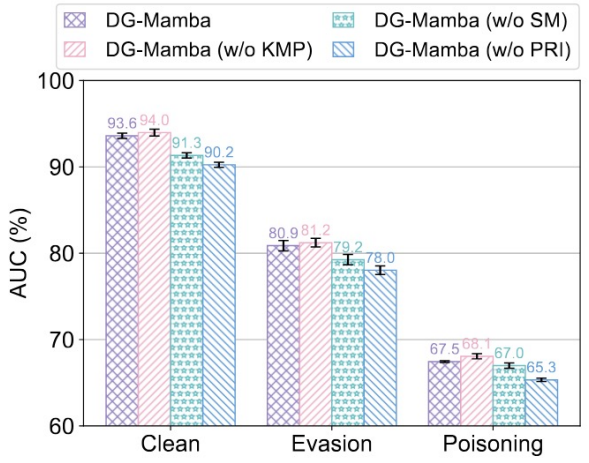
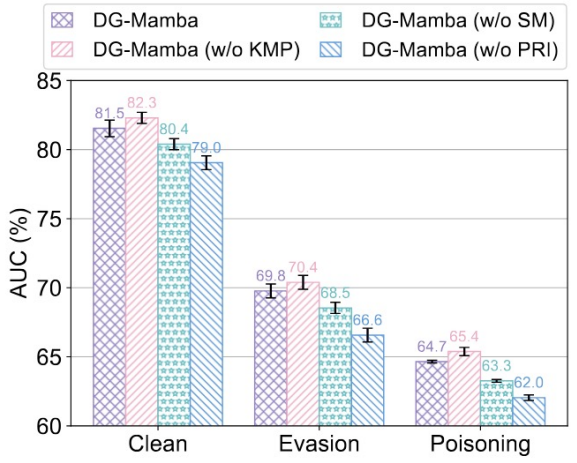


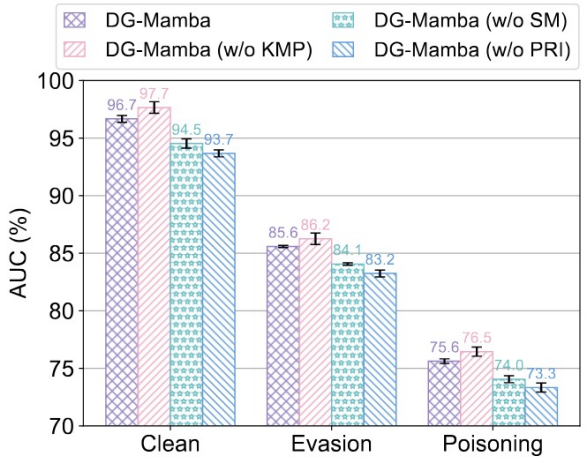
Figure 5: Ablation study.



(a) COLLAB.



(b) Yelp.



(c) ACT.

(Clean)

Hyperparameter Sensitivity Analysis

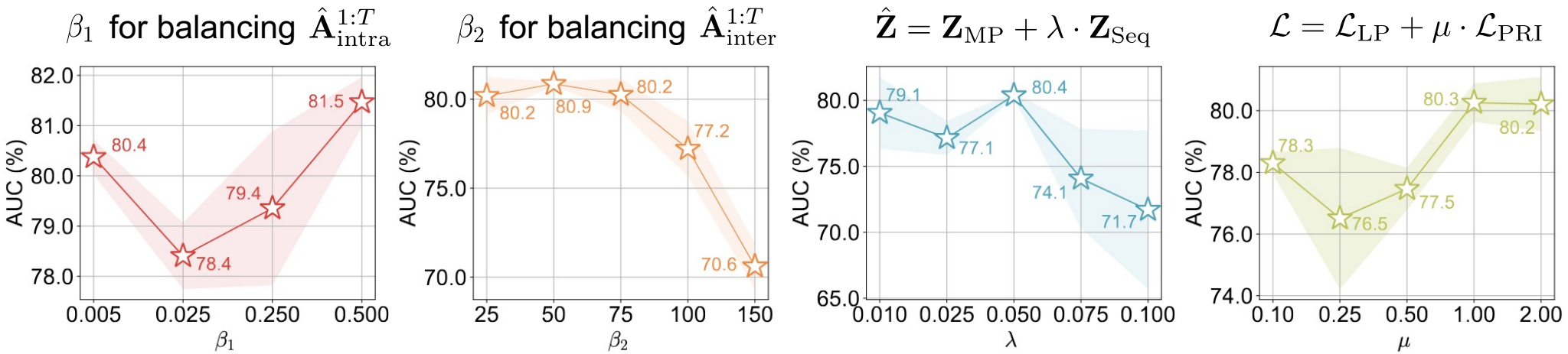


Figure D.6: Hyperparameter sensitivity analysis on Yelp.

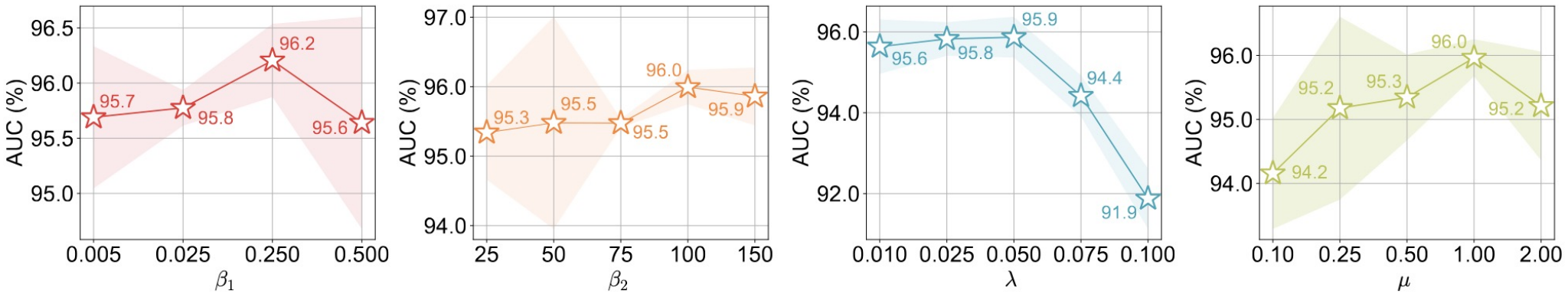


Figure D.7: Hyperparameter sensitivity analysis on ACT.

Visualization of Learned Dynamic Structures

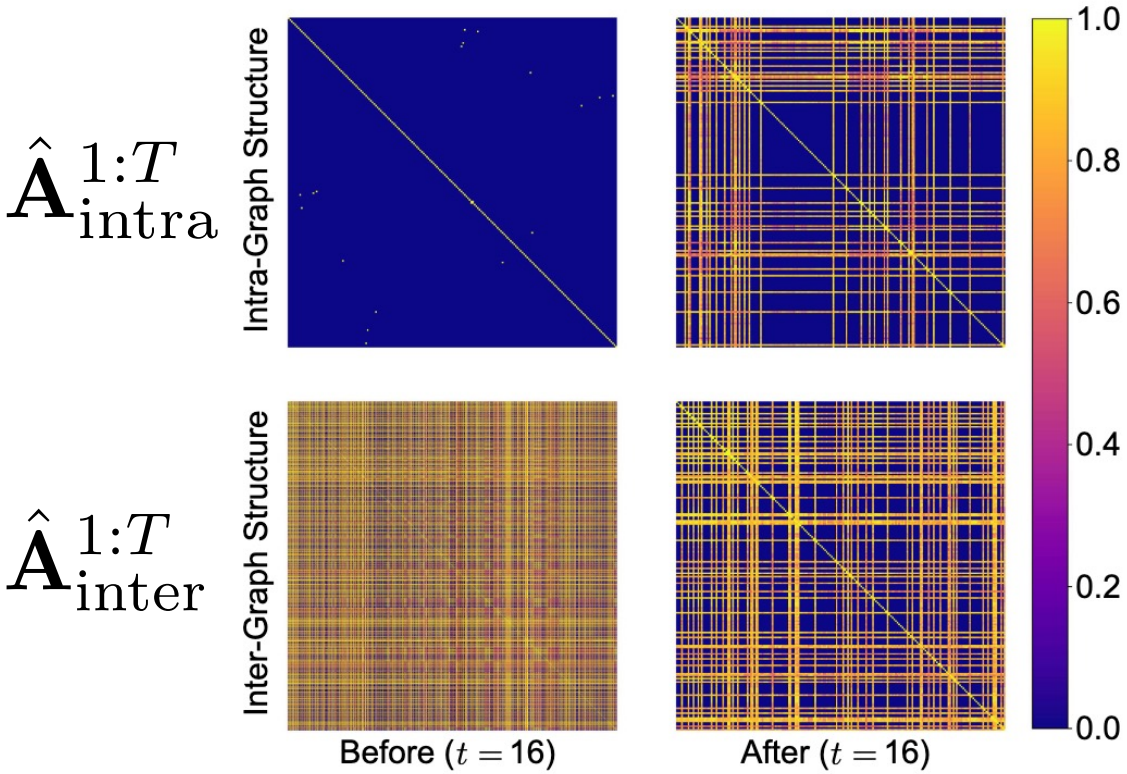


Figure D.8: Structure visualization on COLLAB.

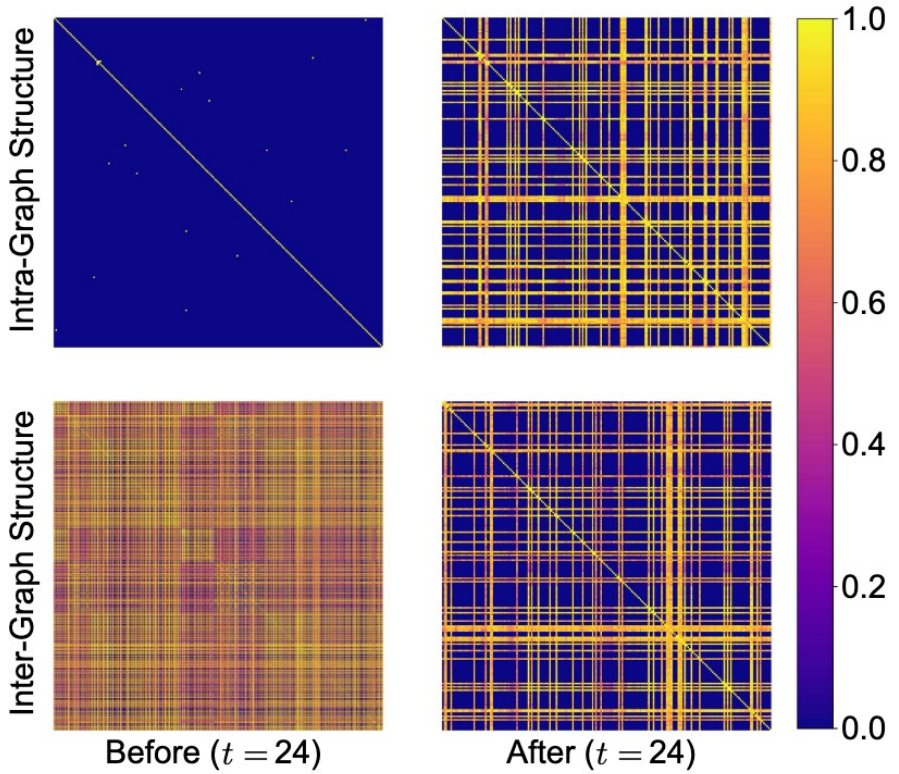


Figure D.9: Structure visualization on Yelp.

Paper Highlights

- We propose a **robust** and **efficient** DGSL framework named **DG-Mamba** with **linear time complexity** for robust representations against adversarial attacks.

- The **kernelized** dynamic graph message-passing operator behaves efficient DGSL with the help of **state-discretized SSM**. The structural information between the original and the learned is regularized with the proposed **PRI for DGSL** to enhance the robustness of the global representation.

- Experiments show **effectiveness**, **robustness**, and **efficiency**, demonstrating its superiority over 12 state-of-the-art baselines against **adversarial attacks**.

MAGIC

MAchine learning for Graph Intelligence Computing

AAAI

Association for the Advancement of Artificial Intelligence

DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning with Selective State Space Models

Haonan Yuan, Qingyun Sun, Zhaonan Wang, Xingcheng Fu, Cheng Ji, Yongjian Wang, Bo Jin, Jianxin Li*

Email: yuanhn@buaa.edu.cn

Paper

Code