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

Haonan Yuan[1], Qingyun Sun[1], Zhaonan Wang[2], Xingcheng Fu[3], Cheng Ji[1], Yongjian Wang[4], Bo Jin[4], Jianxin Li[1]*

[1]SKLCCSE, School of Computer Science and Engineering, Beihang University, China
[2]National Superior College for Engineers, Beihang University, China
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
[4]The Third Research Institute of Ministry of Public Security, China
{yuanhn, sunqy, wangzn, jicheng, lijx}@buaa.edu.cn, fuxc@gxnu.edu.cn, wangyongjian@mcst.org.cn, jinbo@gass.ac.cn

Abstract

Dynamic graphs exhibit intertwined spatio-temporal evolutionary patterns, widely existing in the real world. Nevertheless, the structure incompleteness, noise, and redundancy result in poor robustness for Dynamic Graph Neural Networks (DGNNs). Dynamic Graph Structure Learning (DGSL) offers a promising way to optimize graph structures. However, aside from encountering unacceptable quadratic complexity, it overly relies on heuristic priors, making it hard to discover underlying predictive patterns. How to efficiently refine the dynamic structures, capture intrinsic dependencies, and learn robust representations, remains under-explored. In this work, we propose the novel **DG-Mamba**, a robust and efficient **D**ynamic **G**raph structure learning framework with the Selective State Space Models (**Mamba**). To accelerate the spatio-temporal structure learning, we propose a kernelized dynamic message-passing operator that reduces the quadratic time complexity to linear. To capture global intrinsic dynamics, we establish the dynamic graph as a self-contained system with State Space Model. By discretizing the system states with the cross-snapshot graph adjacency, we enable the long-distance dependencies capturing with the selective snapshot scan. To endow learned dynamic structures more expressive with informativeness, we propose the self-supervised Principle of Relevant Information for DGSL to regularize the most relevant yet least redundant information, enhancing global robustness. Extensive experiments demonstrate the superiority of the robustness and efficiency of our DG-Mamba compared with the state-of-the-art baselines against adversarial attacks.

1 Introduction

Dynamic graphs are ubiquitous in real world, spanning domains such as social media (Sun et al. 2022a), traffic networks (Guo et al. 2021), financial transactions (Pareja et al. 2020), and human mobility (Zhou et al. 2023), *etc.* Their complex spatial and temporal correlation patterns present significant challenges across various downstream deployments. Leveraging exceptional expressive capabilities, Dynamic Graph Neural Networks (DGNNs) intrinsically excel at dynamic graph representation learning by modeling both spatial and temporal predictive patterns, which enjoy the combined merits of both GNNs and sequential models.

*Corresponding author.

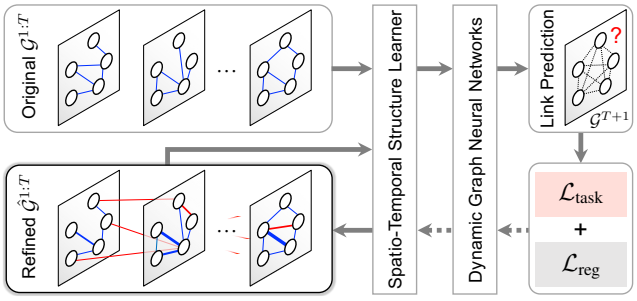


Figure 1: A general paradigm of DGSL.

Recently, there has been a growing research trend on enhancing the efficacy of DGNNs, including focus on improving their ability to capture the intricate spatio-temporal correlations that surpass the 1-WL graph isomorphism test. Most of the DGNNs perform spatio-temporal message-passing over potentially flawed graph structures, assuming the observed graph structures can reflect the ground-truth relationships between nodes. However, this fundamental assumption often leads to suboptimal robustness and generalization performance due to the inherent incompleteness, noise, and redundancy in graph structures, which also make the learned representations susceptible to noise and adversarial attacks (Zügner, Akbarnejad, and Günnemann 2018).

Graph Structure Learning (GSL) has emerged as a crucial graph learning paradigm for iteratively optimizing structures and representations (Sun et al. 2022b,c; Wei et al. 2024; Fu et al. 2023). Similarly, the GSL for dynamic graphs (DGSL) aims to refine spatial- and temporal-wise structures (Figure 1). Despite its potential, DGSL faces several challenges. On one hand, the intricate coupling of spatial and temporal dimensions makes it particularly vulnerable to noise and adversarial attacks in open data environments, which significantly hampers its performance and robustness. Moreover, the predictive patterns indicate the causal behind real-world decision-making. However, current approaches often emphasize the optimization of local structures, overlooking latent long-range dependencies. This oversight results in a notable decline in performance on long-sequence dynamic graph prediction tasks as the sequence length increases.

On the other hand, DGSL demonstrates vast complexity challenges. In addition to the overwhelming quadratic complexity within individual graphs caused by node-pair probabilistic measuring (Wu et al. 2022), attention-based sequential models (*e.g.*, Transformer (Vaswani et al. 2017)) also encounter quadratic complexity when computing step-pair attentions (Shen et al. 2021). As the scale of nodes and sequence length grow explosively, this complexity bottleneck severely hinders the advancement of existing DGSL frameworks. Several works have attempted to address the complexity problems from spatial and temporal perspectives (Wu et al. 2022; Gu and Dao 2023). However, dynamic graphs function as a system with intricate spatio-temporal couplings, necessitating a holistic framework for complexity reduction across both dimensions. Notably, the complexities of both dimensions are not orthogonal but interdependent. Attempting to isolate and reduce spatial complexity without considering temporal complexity, or vice versa, is insufficient. Their interdependence significantly amplifies the overall computational burden. This multifaceted framework is essential for achieving efficient and robust dynamic graph learning, ensuring it can not only handle quadratic complexity but also capture insightful correlations.

Research Question: *How to capture long-range intrinsic dependency of underlying predictive patterns to derive robust representations against adversarial attacks over the denoised structures while* ***simultaneously*** *reduce both spatial and temporal time complexity from quadratic to linear?*

Present Work. In this work, we introduce **DG-Mamba**, a robust and efficient **D**ynamic **G**raph structure learning framework with the selective state space models (**Mamba**). We propose a kernelized dynamic message-passing operator that reduces the quadratic time complexity to linear to accelerate the spatio-temporal structure learning. To break the local Markovian dependence assumption limitations and capture global intrinsic dynamics, we model the dynamic graph with the State Space Model as a system, and discretize the system states with the cross-snapshot graph adjacency. To endow the learned dynamic structures with informative expressiveness, we propose the self-supervised Principle of Relevant Information for DGSL to regularize the most relevant yet least redundant information, enhancing global robustness for downstream tasks. **Our contributions are:**

- We propose a robust and efficient dynamic graph structure learning framework DG-Mamba with linear time invariance property for robust representations against adversarial attacks. To the best of our knowledge, this is the first trial in which the spatio-temporal computational complexity of DGSL has been simultaneously reduced to linear.
- 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 on real-world and synthetic dynamic graphs validate the effectiveness, robustness, and efficiency of the proposed DG-Mamba, demonstrating its superiority over **12** state-of-the-art baselines against adversarial attacks.

2 Related Work

2.1 Robust Dynamic Graph Learning

Dynamic Graph Neural Networks (DGNNs) are prevalent in learning representations by inherently modeling both spatial and temporal features (Han et al. 2021). However, dynamic graphs naturally contain noise and redundant features irrelevant to the target, which compromises DGNN performance. Additionally, DGNNs are prone to the over-smoothing phenomenon, making them less robust and vulnerable to perturbations and adversarial attacks (Zhu et al. 2023). Compared to robust GNNs for static graphs, there are no DGNNs tailored for efficient robust representation learning, currently.

2.2 Dynamic Graph Structure Learning

Graph Structure Learning (GSL) has gained much attention in recent years, aiming to simultaneously learn a denoised structure and robust representations (Zhu et al. 2021), where existing works have successfully investigated GSL methods for static graphs. However, structure learning for dynamic graphs (DGSL) remains largely under-explored, which faces the significant challenge of the computational efficiency bottleneck, as existing methods exhibit quadratic complexity in both spatial and temporal dimensions, rendering them impractical for large-scale and long-sequence dynamic graphs.

2.3 Graph Modeling with State Space Models

State Space Models (SSMs) are foundations for modeling dynamic systems. Recently, Mamba (Gu and Dao 2023) has shown promising performance in efficient sequence modeling. Intuitively, there are several explorations of applying SSMs to graph modeling by converting the non-Euclidean structures to token sequence (Behrouz and Hashemi 2024; Wang et al. 2024; Huang, Miao, and Li 2024) but present unique challenges due to the lack of canonical node ordering. Further, simply transforming dynamic graphs into sequences for handling by SSMs is less satisfying (Behrouz and Hashemi 2024; Wang et al. 2024), as the spatio-temporal coupling of long-range dependencies is difficult to capture, and informative feature patterns are overlooked.

3 Preliminary

Notation. We primarily consider the discrete dynamic representation learning. A discrete dynamic graph is denoted as $\mathbf{DG} = \{\mathcal{G}^t\}_{t=1}^{T}$, where T is the time length. $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ is the graph at time t, where $\mathcal{V}^t$ is the node set and $\mathcal{E}^t$ is the edge set. Let $\mathbf{A}^t \in \{0,1\}^{N\times N}$ be the adjacency matrix and $\mathbf{X}^t \in \mathbb{R}^{N\times d}$ be the node features, where $N = |\mathcal{V}^t|$ denotes the number of nodes and d denotes the feature dimension.

Dynamic Graph Representation Learning. As the most challenging task of dynamic graph representation learning, 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 the existence of edges at $T+1$ given historical graphs $\mathcal{G}^{1:T}$ and next-step node features $\mathbf{X}^{T+1}$. Concretely, the $f_{\boldsymbol{\theta}} = w \circ g$ is compound of a encoder $w(\cdot)$ and a link predictor $g(\cdot)$, *i.e.*, $\mathbf{Z}^{T+1} = w(\mathcal{G}^{1:T}, \mathbf{X}^{T+1})$ and $\hat{\mathbf{Y}}^{T+1} = g(\mathbf{Z}^{T+1})$. The target is to iteratively learn the refined graph $\hat{\mathcal{G}}^{1:T}$ with corresponding robust dynamic graph representations for downstream tasks efficiently.

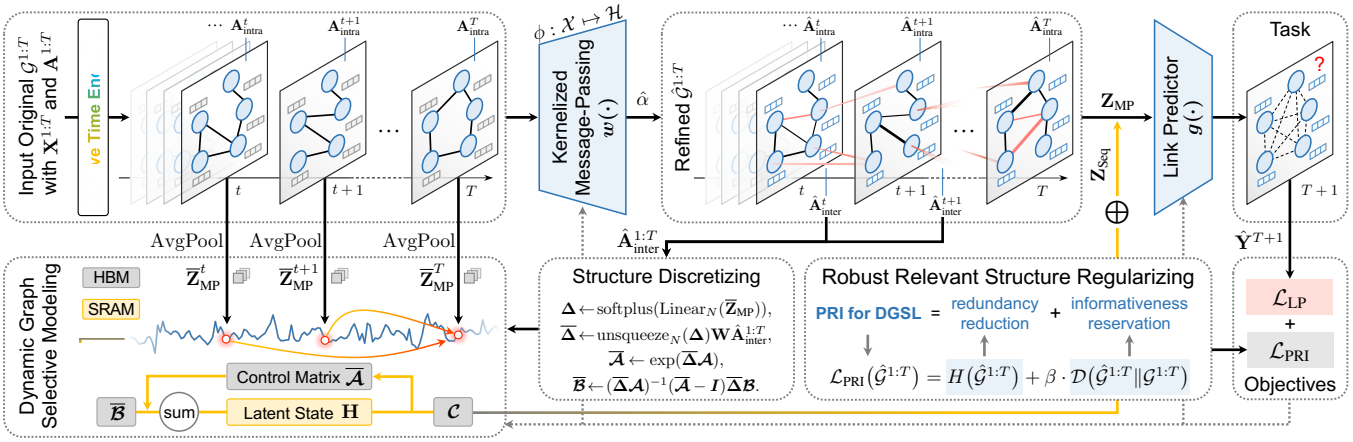


Figure 2: The framework of DG-Mamba. (a) Kernelized message-passing mechanism learns both intra- and inter-graph weights with linear time complexity. (b) Long-range dependencies are strengthened by selective modeling with parameters discretized by learned inter-graph structures. (c) PRI for DGSL is proposed to guarantee robustness against noise and adversarial attacks.

4 DG-Mamba: Robust and Efficient Dynamic Graph Structure Learning

This section elaborates on DG-Mamba with its framework shown in Figure 2. First, we propose a kernelized dynamic graph message-passing operator to accelerate structure learning. Then, we model and discretize the dynamic graph system with inter-graph structures to capture long-range dependencies and intrinsic dynamics. Lastly, we promote the robustness of representations by the self-supervised Principle of Relevant Information.

4.1 Kernelized Message-Passing for Efficient Dynamic Graph Structure Learning

To efficiently learn spatio-temporal structures, we propose the kernelized message-passing mechanism performing on a dynamic graph attention network, where the learnable edge weights play a role in both structure refinement and attentive feature aggregation. As most literature presumed, we make the following assumption.

Assumption 1 (Dynamic Graph Markov Dependence). Assume the $\mathbf{DG} = \{\mathcal{G}^t\}_{t=1}^T$ 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.*,

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

Assumption 1 declares the local dependencies that sculpt the weighted message-passing routes between graph pairs. Given node u in $\mathcal{G}^t$ at the l-th layer, the attentive aggregation at the next $(l+1)$-th layer is,

$$\mathbf{z}_u^{t(l+1)} = \sum_v \hat{\alpha}_{uv}^{t-1:t(l)} (\mathbf{W}\mathbf{z}_v^{\cdot(l)}), \text{for all } v \in \mathcal{N}(u)^{t-1:t}, \tag{2}$$

where $\mathbf{W}$ is learnable matrix. $\mathcal{N}(u)^{t-1:t}$ denotes u's neighbors in $\mathcal{G}^{t-1}$ and $\mathcal{G}^t$. $\hat{\alpha}_{uv}^{t-1:t(l)}$ contains structure weights for $\mathcal{G}^t$ and message-passing routes between $\mathcal{G}^{t-1}$ and $\mathcal{G}^t$, *i.e.*,

$$\hat{\alpha}_{uv}^{t-1:t(l)} = \frac{\exp(\sigma((\mathbf{W}\mathbf{z}_u^{t(l)})^\top (\mathbf{W}\mathbf{z}_v^{\cdot(l)})))}{\sum_m \exp(\sigma((\mathbf{W}\mathbf{z}_u^{t(l)})^\top (\mathbf{W}\mathbf{z}_m^{\cdot(l)})))}. \tag{3}$$

Note that, we omit the limits of the summations always for nodes in the $\mathcal{N}(u)^{t-1:t}$ for brevity. Intuitively, Softmax pairwise edge weights updating and representation aggregation in Eq. (2) and Eq. (3) contribute to unacceptable quadratic complexity for dynamic graph structure learning. Inspired by kernel-based methods that employ kernel functions to model edge weights (Zhu et al. 2021), we combine Eq. (2) and Eq. (3) with kernel $k(\cdot,\cdot)$ for measuring similarity, *i.e.*,

$$\mathbf{z}_u^{t(l+1)} = \sum_v \frac{k(\mathbf{W}\mathbf{z}_u^{t(l)}, \mathbf{W}\mathbf{z}_v^{\cdot(l)})}{\sum_m k(\mathbf{W}\mathbf{z}_u^{t(l)}, \mathbf{W}\mathbf{z}_m^{\cdot(l)})} \cdot \mathbf{W}\mathbf{z}_v^{\cdot(l)}. \tag{4}$$

Kernel Estimation by Random Features. Instead of explicitly finding a feature map φ from representation space $\mathcal{X}$ to the reproducing kernel Hilbert space $\mathcal{H}$ and calculate the kernel $k(\cdot,\cdot)$ by inner production $\langle \cdot,\cdot \rangle_{\mathcal{H}}$, the Mercer's theorem guarantees an implicitly defined function ϕ exists if and only if $k(\cdot,\cdot)$ is a positive definite kernel (Mercer 1909), *i.e.*,

$$k(\mathbf{x}_1, \mathbf{x}_2)_{\mathcal{X}} = \langle \varphi(\mathbf{x}_1), \varphi(\mathbf{x}_2) \rangle_{\mathcal{H}} \doteq \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2). \tag{5}$$

In this way, Eq. (4) can be converted into a simpler form,

$$\mathbf{z}_u^{t(l+1)} = \sum_v \frac{\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \phi(\mathbf{W}\mathbf{z}_v^{\cdot(l)})}{\sum_m \phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \phi(\mathbf{W}\mathbf{z}_m^{\cdot(l)})} \cdot \mathbf{W}\mathbf{z}_v^{\cdot(l)} \tag{6}$$

$$= \frac{\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \sum_v \phi(\mathbf{W}\mathbf{z}_v^{\cdot(l)}) \mathbf{W}\mathbf{z}_v^{\cdot(l)\top}}{\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top \sum_m \phi(\mathbf{W}\mathbf{z}_m^{\cdot(l)})}. \tag{7}$$

Note that, $\phi(\mathbf{W}\mathbf{z}_u^{t(l)})^\top$ is irreducible as it is the matrices operations. It is noteworthy that the two summations greatly contribute to decreasing the quadratic complexity as they can be computed once and stored for each u. Intuitively, kernel $k(\cdot,\cdot)$ can be estimated by the Positive Random Features (PRF) (Choromanski et al. 2020) for Softmax approximation in Lamma 1 that satisfies the Mercer's theorem, *i.e.*,

$$\phi(\mathbf{x}) = \sum_{i=1}^m \frac{1}{\sqrt{m}} \exp\left(\boldsymbol{\omega}_i^\top \mathbf{x} - \frac{\|\mathbf{x}\|_2^2}{2}\right), \tag{8}$$

where m is the projection dimension of kernel $k(\cdot,\cdot)$, and $\boldsymbol{\omega}_i$ is the random feature shifting to the target embedding space. Under such settings, the structure updating and representation aggregation are differentiable by the Gumbel-Softmax reparameterization trick (Wu et al. 2022).

Intra- and Inter-Structure Efficient Query. As Eq. (7) merges both the edge reweighting and message-passing process in a unified and implicit manner, we can still explicitly obtain the optimized edge weights $\hat{\alpha}_{uv}^{t-1:t}$ by efficiently querying the approximated kernels of any node pairs with details in Appendix B.1. We decompose overall $\hat{\alpha}_{uv}^{t-1:t}$ into intra-graph and inter-graph weights, which constructed two types of adjacency matrices for the consequent refining, *i.e.*,

$$\hat{\mathbf{A}}_{\text{intra}}^{t} = \{\hat{\alpha}_{uv}^{t-1:t}\}^{N\times N}, \text{where } u, v \in \mathcal{V}^t, \tag{9}$$

$$\hat{\mathbf{A}}_{\text{inter}}^{t} = \{\hat{\alpha}_{uv}^{t-1:t}\}^{N\times N}, \text{where } u \in \mathcal{V}^t, v \in \mathcal{V}^{t-1}, \tag{10}$$

where $\hat{\mathbf{A}}_{\text{intra}}^{t}$ and $\hat{\mathbf{A}}_{\text{inter}}^{t}$ are then for the intra-graph and inter-graph structure regularizing (Eq. (19)), respectively.

4.2 Long-range Dependencies Selective Modeling

The success of spatio-temporal kernel is contingent upon Assumption 1, which substantially compromises with the Markov condition. However, real-world dynamic graphs can be exceedingly long and exhibit uncertain periodic variations, characterized by long-range dependencies between graph snapshots, where the local dependencies constraints significantly hinder their selective feature capturing.

Dynamic Graph System Modeling. To strengthen the global long-range dependencies selective modeling without increasing the spatial computational complexity, we propose constructing the dynamic graph as a self-contained system with the State Space Models. Specifically, SSMs are defined with the state transition matrix $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n\times n}$ and two projection matrices $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{n\times 1}$, $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{1\times n}$. Given the continuous input sequence $\mathbf{x}(t) \in \mathbb{R}^l$, the SSM updates the latent state $\mathbf{h}(t) \in \mathbb{R}^{n\times l}$ and output $\mathbf{y}(t) \in \mathbb{R}^l$, *i.e.*,

$$\mathbf{h}'(t) = \boldsymbol{\mathcal{A}}\mathbf{h}(t) + \boldsymbol{\mathcal{B}}\mathbf{x}(t),\ \mathbf{y}(t) = \boldsymbol{\mathcal{C}}\mathbf{h}(t), \tag{11}$$

where $\boldsymbol{\mathcal{A}}$ controls how current state evolves over time in a global view, $\boldsymbol{\mathcal{B}}$ describes how the input influences the state, and $\boldsymbol{\mathcal{C}}$ responses how the current state translate to the output.

To effectively integrate Eq. (11) within the deep learning settings, it is essential to discretize the continuous system. However, there are two critical problems to address: How to enable SSMs attention-aware to each time step in replacing the self-attention mechanism that consumes quadratic complexity? And how we incorporate the refined inter-graph structures into the state updating process such that weighted message-passing routes between graphs can be considered?

Selective Discretizing and Parameterizing with Inter-Graph Structures. To address the aforementioned problems, we propose the Dynamic Graph Selective Scan Mechanism that discretizes the system parameters ($\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$, *etc.*) function of each step input to selectively control which part of the graph sequence with how much attention can flow into the hidden state, and parameterized system states with the inter-graph structures $\hat{\mathbf{A}}_{\text{inter}}^{1:T}$ to integrate local dependencies into the long-range global dependencies capturing.

Denote $\mathbf{Z}_{\text{MP}} \in \mathbb{R}^{B\times T\times N\times D_0}$ as the learned node embeddings after spatio-temporal message-passing once in Eq. (7), where B means the batch size, T denotes the dynamic graph length, and D_0 represents the latent feature dimension. Such that, the sequential input $\overline{\mathbf{Z}}_{\text{MP}} \in \mathbb{R}^{B\times T\times N}$ is the average pooling on $\mathbf{Z}_{\text{MP}}$, which implies the current latent state for each step. $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{N\times D}$ is randomly initialized. $\boldsymbol{\mathcal{B}}$ and $\boldsymbol{\mathcal{C}}$ are further parameterized to each step input, *i.e.*,

$$\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}} \in \mathbb{R}^{B\times T\times D} \leftarrow \text{Linear}_D(\overline{\mathbf{Z}}_{\text{MP}}). \tag{12}$$

Additionally, a timestep-wise parameter $\boldsymbol{\Delta} \in \mathbb{R}^{B\times T\times N}$ initialized by the input $\overline{\mathbf{Z}}_{\text{MP}}$ is utilized to discrete the dynamic graph system with the learned inter-graph structures, *i.e.*,

$$\boldsymbol{\Delta} \in \mathbb{R}^{B\times T\times N} \leftarrow \text{softplus}(\text{Linear}_N(\overline{\mathbf{Z}}_{\text{MP}})), \tag{13}$$

$$\overline{\boldsymbol{\Delta}} \in \mathbb{R}^{B\times T\times N} \leftarrow \text{unsqueeze}_N(\boldsymbol{\Delta}) \cdot \mathbf{W}\hat{\mathbf{A}}_{\text{inter}}^{1:T}. \tag{14}$$

Following the continuous signal reconstructing zero-order hold (ZOH) rules, parameters $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are discretized by,

$$\overline{\boldsymbol{\mathcal{A}}} \leftarrow \exp(\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}}), \overline{\boldsymbol{\mathcal{B}}} \leftarrow (\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}})^{-1}(\exp(\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{A}}) - \boldsymbol{I})\overline{\boldsymbol{\Delta}}\boldsymbol{\mathcal{B}}. \tag{15}$$

Consequently, the output of the dynamic graph system is,

$$\mathbf{H}_t = \overline{\boldsymbol{\mathcal{A}}}\mathbf{H}_{t-1} + \overline{\boldsymbol{\mathcal{B}}}(\overline{\mathbf{Z}}_{\text{MP}})_t,\ (\mathbf{Z}_{\text{Seq}})_t = \boldsymbol{\mathcal{C}}\mathbf{H}_t, \tag{16}$$

where $\mathbf{H}$ is the latent states. The detailed dynamic graph selective scan is described in line 2 to line 11 in Algorithm 2. The output $\mathbf{Z}_{\text{Seq}} \in \mathbb{R}^{B\times T\times N}$ selectively emerged the temporal semantics with its long-range dependencies, which is then acting as the supervision on $\mathbf{Z}_{\text{MP}}$, leading to the combined node representation, *i.e.*,

$$\hat{\mathbf{Z}} = \mathbf{Z}_{\text{MP}} + \lambda \cdot \text{unsqueeze}_{D_0}(\mathbf{Z}_{\text{Seq}}), \tag{17}$$

where λ is the hyperparameter.

4.3 Robust Relevant Structure Regularizing

The last milestone for the research goal is to strengthen the robustness of the updated representations against potential noise and adversarial attacks in surrounding environments. This is a dual-purpose objective: while the learned implicit structure represents intrinsic dependencies, the physically-structured raw graphs contain rich interpretability semantics. Robust DGSL should expect to learn minimal but sufficient structural information from an information-theoretic view.

Principle of Relevant Information (PRI). To reduce redundant structural information as well as reserve critical predictive patterns, we utilize the self-supervised PRI (Principe 2010) to formulate the criteria for dynamic graph structure learning, which plays the role of structural regularizers.

Definition 1 (**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,

$$\mathcal{L}_{\text{PRI}}(\hat{\mathcal{G}}^{1:T}) = H(\hat{\mathcal{G}}^{1:T}) + \beta \cdot \mathcal{D}(\hat{\mathcal{G}}^{1:T} \| \mathcal{G}^{1:T}), \tag{18}$$

where $H(\cdot)$ denotes the Shannon entropy that measures the redundancy of $\hat{\mathcal{G}}^{1:T}$. $\mathcal{D}(\cdot\|\cdot)$ is the divergence that reflects the discrepancy between two terms. The hyperparameter β plays the trade-off between the redundancy reduction and predictive patterns reservation. Larger β leads to more information reserved from the input dynamic graphs, and vice versa.

PRI for DGSL is indispensable for strengthening structure robustness in a self-supervised manner, for it encourages DG-Mamba emphasizes on the informative, discriminative, and invariant structural patterns across historical graph snapshots while filtering out potential noise and redundant information that damages the parameter fitting process.

Derivation of PRI for DGSL. As optimize Eq. (18) straightforwardly is indifferentiable, we approximately decompose it into respective spatial- and temporal-wise regularizing with learned intra- and inter-graph structures, *i.e.*,

$$\mathcal{L}_{\text{PRI}}(\hat{\mathcal{G}}^{1:T}) \triangleq \mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}_{\text{intra}}^{1:T}) + \mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}_{\text{inter}}^{1:T}). \tag{19}$$

To regularize intra-graph structures, we transform the divergence term into edge-level constraints with the loss equivalence guarantee in Appendix B.2, *i.e.*,

$$\mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}_{\text{intra}}^{1:T}) \doteq H(\hat{\mathbf{A}}_{\text{intra}}^{1:T}) + \beta_1 \cdot \mathcal{L}_{\text{edge}}, \tag{20}$$

$$\text{and} \quad \mathcal{L}_{\text{edge}} = -\frac{1}{NT}\sum_{t=1}^{T}\sum_{u,v\in\mathcal{E}^t}\frac{1}{d(u)}\log\hat{\alpha}_{uv}^{t}, \tag{21}$$

where β_1 is the hyperparameter, and $d(\cdot)$ measures the node degree. Eq. (21) is the maximum likelihood estimation for edges in $\mathcal{E}^t$. For inter-graph structure regularizing, as there is no feasible gound-truth supervision for the original $\mathbf{A}_{\text{inter}}^{1:T}$, we utilize the structure-aware $\mathbf{Z}_{\text{Seq}}$ and $\overline{\mathbf{Z}}_{\text{MP}}$ instead, *i.e.*,

$$\mathcal{L}_{\text{PRI}}(\hat{\mathbf{A}}_{\text{inter}}^{1:T}) \doteq H(\mathbf{Z}_{\text{Seq}}) + \beta_2 \cdot \mathcal{D}(\mathbf{Z}_{\text{Seq}} \| \overline{\mathbf{Z}}_{\text{MP}}), \tag{22}$$

where β_2 is a hyperparameter, and the KL-divergence is implemented to the divergence term $\mathcal{D}(\cdot,\cdot)$.

4.4 Optimization and Complexity Analysis

The overall optimization objective of DG-Mamba is,

$$\mathcal{L} = \mathcal{L}_{\text{LP}}(\mathbf{Y}^{T+1}, \hat{\mathbf{Y}}^{T+1}) + \mu \cdot \mathcal{L}_{\text{PRI}}(\hat{\mathcal{G}}^{1:T}), \tag{23}$$

where $\mathcal{L}_{\text{LP}}$ is implemented by the cross-entropy loss for future link prediction, $\mathcal{L}_{\text{PRI}}(\hat{\mathcal{G}}^{1:T})$ is derived by Eq. (20) and Eq. (22). μ is the Lagrangian hyperparameter. The training pipeline is illustrated in Algorithm 1 and 2 (Appendix A).

Computational Complexity. We denote $|\mathcal{V}|$ and $|\mathcal{E}|$ as the average number of nodes and edges in each graph snapshot, respectively, and T denotes the graph length. The computational complexity of the kernelized message-passing (Eq. (7)) is $\mathcal{O}(T|\mathcal{V}|)$, and the intra- (Eq. (9)) and inter-graph structure query (Eq. (10)) contributes $\mathcal{O}(T|\mathcal{E}|)$. For dynamic graph selective scan, we implement a hardware-aware algorithm to accelerate the long-range dependencies modeling by the kernel fusion and recomputation (Gu and Dao 2023), which approximately requires $\mathcal{O}(T)$. We omit the computational complexity brought by feature projection and aggregation for brevity as the feature dimensions are significantly smaller than $|\mathcal{V}|$ and $|\mathcal{E}|$. Such that, the overall computational complexity of DG-Mamba is linear with the length, averaged number of nodes, and edges, *i.e.*,

$$\mathcal{O}(T(|\mathcal{V}| + |\mathcal{E}|)), \tag{24}$$

which is superior efficient than state-of-the-art DGNNs, especially when the original graphs are less dense. Detailed complexity analysis can be found in Appendix.

5 Experiment

In this section, we conduct extensive experiments on both real-world and synthetic dynamic graph datasets to evaluate the effectiveness, robustness, and efficiency of DG-Mamba against multi-type adversarial attacks. Detailed settings and additional results can be found in the Appendix.

5.1 Experimental Settings

Dynamic Graph Datasets. We evaluate DG-Mamba on the challenging future link prediction with three real-world dynamic graph datasets. ① **COLLAB** (Tang et al. 2012) is an academic collaboration dataset with papers published in 16 years. ② **Yelp** (Sankar et al. 2020) contains customer reviews for 24 months. ③ **ACT** (Kumar, Zhang, and Leskovec 2019) describes actions taken by users on a popular MOOC website within 30 days, and each action has a binary label. Statistics of the datasets are concluded in Appendix.

Baselines. We compare with four categories, 12 baselines. ① **Static GNNs:** GAE and VGAE (Kipf and Welling 2016), GAT (Veličković et al. 2018). ② **DGNNs:** GCRN (Seo et al. 2018), EvolveGCN (Pareja et al. 2020), DySAT (Sankar et al. 2020), and SpoT-Mamba (Choi et al. 2024). ③ **DGSL:** RDGSL (Zhang et al. 2023b), TGSL (Zhang et al. 2023a). ④ **Robust (D)GNNs:** RGCN (Zhu et al. 2019), WinGNN (Zhu et al. 2023), and DGIB (Yuan et al. 2024).

Adversarial Attack Settings. We compare baselines and DG-Mamba under two typical adversarial attack scenarios.

- **Non-targeted:** We make synthetic datasets by attacking graph structures and node features, respectively. ① **Structure Attack:** We randomly remove one out of five types of edges in the training and validation graphs. This removal makes the task more challenging than the real-world situations as the model cannot access any information on the removed edges. ② **Feature Attack:** Gaussian noise $\lambda \cdot r \cdot \epsilon$ is added to the node features, where r is the reference amplitude of the original features, and $\epsilon \sim N(\mathbf{0}, \mathbf{I})$. Parameter $\lambda \in \{0.5, 1.0, 1.5\}$ controls the degree of the attack.
- **Targeted:** We apply the prevailing NETTACK (Zügner, Akbarnejad, and Günnemann 2018), a targeted adversarial attack library on graphs designed to target nodes by altering their connected edges or node features. We simultaneously consider the evasion and poisoning attack. ① **Evasion Attack:** Train on clean data, test on the attacked data. ② **Poisoning Attack:** The entire dataset is attacked before model training and testing. In both scenarios, we use GAT (Veličković et al. 2018) as the surrogate model. The number of perturbations n is set to $\{1, 2, 3\}$.

Parameter Settings. We set number of layers as 2 for baselines, 1 for DG-Mamba to avoid overfitting. Latent dimension is set to 128. Baseline hyperparameters follow recommended values from their papers and are fine-tuned for fairness. Configuration files provide values for β_1, β_2, λ, and μ. We optimize with Adam (Kingma and Ba 2014), selecting the learning rate from {1e-02, 1e-03, 1e-04, 1e-05}. The maximum number of epochs is 1,000, with early stopping.

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.

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. "$\Delta\%$" indicates the relative performance decrease after targeted adversarial attacks compared to that on the clean datasets.

5.2 Against Non-Targeted Adversarial Attacks

In this section, we evaluate the model performance on future link prediction and its robustness to non-targeted (random) adversarial attacks on structures and node features. We report results using AUC (%) scores from five runs in Table 1.

Analysis. In most cases, DG-Mamba outperforms other baselines significantly. Static GNNs are not well-suited for dynamic scenarios, struggling to adapt when structures and features evolve. Dynamic GNNs underperform due to their insufficient handling of complex coupling dynamics. DGSL baselines exhibit sensitivity to noise caused by lacking modeling of the underlying predictive patterns, leading to sharp drops in performance under high-intensity feature attacks, especially in datasets with strong temporal relations. Though DGIB slightly surpasses DG-Mamba in a few cases, it generally fails due to drawbacks brought by its strong Markov condition assumption, which greatly damages capturing of the long-range dependencies for strengthening robustness.

5.3 Against Targeted Adversarial Attacks

We continue to evaluate with competitive baselines standing out in Table 1, focusing on link prediction performance and defense against targeted adversarial attacks with a relative decrease. Results of AUC (%) are reported in Table 2.

Analysis. DG-Mamba consistently demonstrates strong robustness across all datasets compared to other competitive baselines, showing the lowest average percentage decrease under both evasion and poisoning attacks. While baselines like WinGNN and DGIB also demonstrate relative robustness, they are more affected by these sophisticated attacks, particularly in the ACT dataset, which appears to be the most challenging for most baselines with larger drops in the AUC scores across the board. Larger AUC decreases witnessed in baselines like TGSL, DySAT, and especially SpoT-Mamba generally exhibits the highest vulnerability among the baselines considered, further highlighting the challenge of defending against targeted adversarial attacks in the real world.

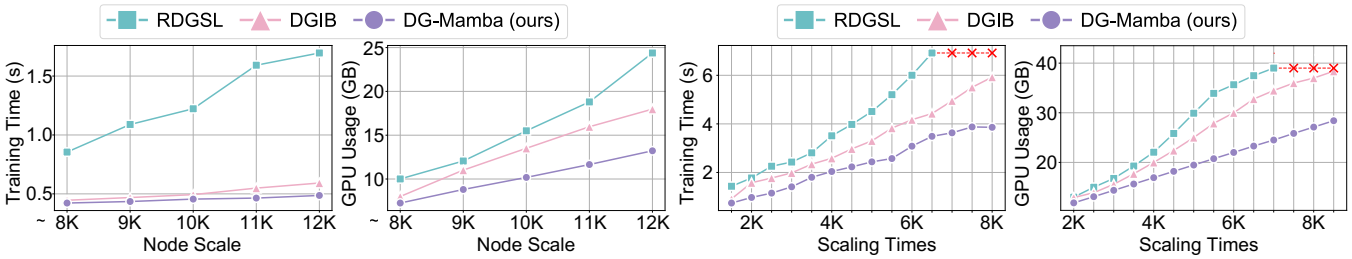
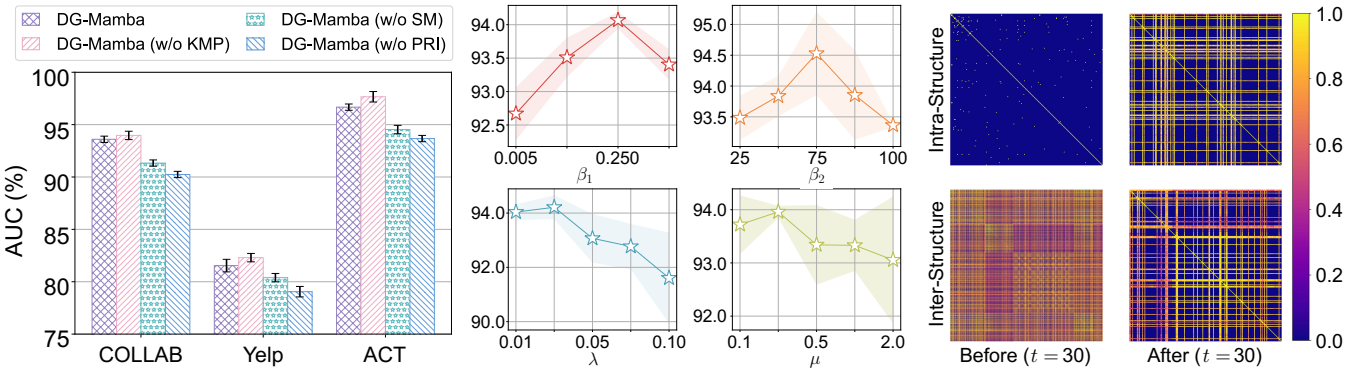


Figure 3: Scaling efficiency analysis on Yelp. Left: Node scaling efficiency. Right: Sequence scaling efficiency.



(a) Ablation study. (b) Hyperparameter analysis. (c) Learned structure visualization.

Figure 4: Additional results for ablation studies, hyperparameter analysis, and structure visualizations.

5.4 Scaling Efficiency Analysis

To evaluate efficiency of DG-Mamba, we generate synthetic datasets by manipulating the node scale and sequence length for the datasets introduced in Section 5.1. We plot training time per epoch and GPU usage at peak time in Figure 3.

Analysis. The results highlight superior efficiency of DG-Mamba in both the node scale and sequence length scaling scenarios, where its training time and GPU usage scale up near linearly, which is consistent with the theoretical analysis conclusion. For node scaling, compared with RDGSL, DG-Mamba reduced training time and GPU usage up to 71.3% and 45.8%, respectively. For sequence length scaling, compared with DGIB-Cat, DG-Mamba reduced training time and GPU usage up to 38.4%, and 28.9%, respectively. Beyond 6.5 times scaling, while RDGSL and DGIB both fail stuking by OOM, DG-Mamba still manages to operate within GPU limits even at the highest scaling factor tested, demonstrating its efficiency in long-span dynamic graphs.

5.5 Ablation Study

We analyze the effectiveness of the three variants:

- **DG-Mamba (*w/o KMP*):** We replace the efficient spatio-temporal kernelized message-passing in Section 4.1 with the vanilla attention-based message-passing mechanism.
- **DG-Mamba (*w/o SM*):** We remove the long-range dependencies selective modeling proposed in Section 4.2.
- **DG-Mamba (*w/o PRI*):** We remove Principle of Relevant Information for DGSL regularizing term in Section 4.3.

Analysis. Overall, the DG-Mamba outperforms the other two variants, which validates the indispensable effectiveness of the dependencies selective modeling mechanism and PRI for DGSL. We claim that the exceeding performance of DG-Mamba (*w/o KMP*) is within our expectation as the kernelized message-passing sacrifices effectiveness for efficiency.

5.6 Hyperparameter Sensitivity Analysis

We evaluate sensitivity of important hyperparameters in Figure 4b, where β_1 and β_2 control importance of the distortion in PRI, μ and λ play the trade-off role between node embeddings and loss terms. Results demonstrate the performance is sensitive to different values and contains a reasonable range.

5.7 Visualization of Learned Dynamic Structures

We visualize edge weights for intra- and inter-graph structures of ACT before and after training in Figure 4c. Results demonstrate DG-Mamba can effectively emphasize on key structure patterns for prediction as well as denoising irrelevant features, which contributes to improving robustness.

6 Conclusion

In this paper, we present a robust and efficient DGSL framework named DG-Mamba with linear time complexity. The kernelized message-passing behaves efficiently with state-discretized SSM. Learned structures are regularized with the proposed PRI for DGSL. Long-range dependencies and underlying predictive patterns are uncovered to strengthen robustness. Extensive experiments demonstrate its superiority.

Acknowledgments

The corresponding author is Jianxin Li. Authors of this paper are supported by the National Natural Science Foundation of China through grants No.623B2010, No.62225202, and No.62302023, and the Fundamental Research Funds for the Central Universities. We extend our sincere thanks to all authors for their valuable efforts and contributions.

References

Behrouz, A.; and Hashemi, F. 2024. Graph mamba: Towards learning on graphs with state space models. *arXiv preprint arXiv:2402.08678*.

Choi, J.; Kim, H.; An, M.; and Whang, J. J. 2024. SpoT-Mamba: Learning Long-Range Dependency on Spatio-Temporal Graphs with Selective State Spaces. *arXiv preprint arXiv:2406.11244*.

Choromanski, K. M.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J. Q.; Mohiuddin, A.; Kaiser, L.; et al. 2020. Rethinking Attention with Performers. In *International Conference on Learning Representations*.

Fu, X.; Wei, Y.; Sun, Q.; Yuan, H.; Wu, J.; Peng, H.; and Li, J. 2023. Hyperbolic geometric graph representation learning for hierarchy-imbalance node classification. In *Proceedings of the ACM Web Conference 2023*, 460–468.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Guo, S.; Lin, Y.; Wan, H.; Li, X.; and Cong, G. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5415–5428.

Han, Y.; Huang, G.; Song, S.; Yang, L.; Wang, H.; and Wang, Y. 2021. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7436–7456.

Huang, Y.; Miao, S.; and Li, P. 2024. What Can We Learn from State Space Models for Machine Learning on Graphs? *arXiv preprint arXiv:2406.05815*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kumar, S.; Zhang, X.; and Leskovec, J. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1269–1278.

Mercer, J. 1909. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458): 415–446.

Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; and Leiserson, C. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5363–5370.

Principe, J. C. 2010. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media.

Sankar, A.; Wu, Y.; Gou, L.; Zhang, W.; and Yang, H. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, 519–527.

Seo, Y.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*, 362–373. Springer.

Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; and Li, H. 2021. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3531–3539.

Sun, L.; Zhang, Z.; Wang, F.; Ji, P.; Wen, J.; Su, S.; and Philip, S. Y. 2022a. Aligning dynamic social networks: An optimization over dynamic graph autoencoder. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5597–5611.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Fu, X.; Ji, C.; and Philip, S. Y. 2022b. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4165–4174.

Sun, Q.; Li, J.; Yuan, H.; Fu, X.; Peng, H.; Ji, C.; Li, Q.; and Yu, P. S. 2022c. Position-aware structure learning for graph topology-imbalance by relieving under-reaching and over-squashing. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1848–1857.

Tang, J.; Wu, S.; Sun, J.; and Su, H. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1285–1293.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, C.; Tsepa, O.; Ma, J.; and Wang, B. 2024. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*.

Wei, Y.; Yuan, H.; Fu, X.; Sun, Q.; Peng, H.; Li, X.; and Hu, C. 2024. Poincaré Differential Privacy for Hierarchy-aware Graph Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9160–9168.

Wu, Q.; Zhao, W.; Li, Z.; Wipf, D. P.; and Yan, J. 2022. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35: 27387–27401.

Yuan, H.; Sun, Q.; Fu, X.; Ji, C.; and Li, J. 2024. Dynamic Graph Information Bottleneck. In *Proceedings of the ACM on Web Conference 2024*, 469–480.

Zhang, H.; Han, X.; Xiao, X.; and Bai, J. 2023a. Time-aware Graph Structure Learning via Sequence Prediction on Temporal Graphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3288–3297.

Zhang, S.; Xiong, Y.; Zhang, Y.; Sun, Y.; Chen, X.; Jiao, Y.; and Zhu, Y. 2023b. RDGSL: Dynamic Graph Representation Learning with Structure Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3174–3183.

Zhou, Z.; Yang, K.; Liang, Y.; Wang, B.; Chen, H.; and Wang, Y. 2023. Predicting collective human mobility via countering spatiotemporal heterogeneity. *IEEE Transactions on Mobile Computing*.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1399–1407.

Zhu, Y.; Cong, F.; Zhang, D.; Gong, W.; Lin, Q.; Feng, W.; Dong, Y.; and Tang, J. 2023. Wingnn: Dynamic graph neural networks with random gradient aggregation window. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3650–3662.

Zhu, Y.; Xu, W.; Zhang, J.; Liu, Q.; Wu, S.; and Wang, L. 2021. Deep graph structure learning for robust representations: A survey. *arXiv preprint arXiv:2103.03036*, 14: 1–1.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2847–2856.