Graph2Video: Leveraging Video Models to Model Dynamic Graph Evolution

Hua Liu[1], Yanbin Wei[1, 2], Fei Xing[3], Tyler Derr[4], Haoyu Han[5], Yu Zhang[1]*

[1]Southern University of Science and Technology
[2]The Hong Kong University of Science and Technology
[3]City University of Hong Kong
[4]Vanderbilt University
[5]Michigan State University
{liuh5, zhangy7}@sustech.edu.cn, yanbin.ust@gmail.com, fxing8-c@my.cityu.edu.hk, tyler.derr@vanderbilt.edu, hanhaoy1@msu.edu

Abstract

Dynamic graphs are common in real-world systems such as social media, recommender systems, and traffic networks. Existing dynamic graph models for link prediction often fall short in capturing the complexity of temporal evolution. They tend to overlook fine-grained variations in temporal interaction order, struggle with dependencies that span long time horizons, and provide limited modeling of pair-specific relational dynamics. To address these challenges, we propose **Graph2Video**, a video-inspired framework that views the temporal neighborhood of a target link as a sequence of "graph frames". By stacking temporally ordered subgraph frames into a "graph video", Graph2Video leverages the inductive biases of video foundation models to capture both fine-grained local variations and long-range temporal dynamics. It generates a link-level embedding that serves as a lightweight, plug-and-play, link-centric memory unit. This embedding integrates seamlessly into existing dynamic graph encoders, effectively addressing the limitations of prior approaches. Extensive experiments on benchmark datasets show that Graph2Video outperforms state-of-the-art baselines in the link prediction task in most cases. The results highlight that borrowing spatio-temporal modeling techniques from computer vision provides a principled and effective avenue for advancing dynamic graph learning.

Code — https://github.com/hualiu829/Graph2Video

1 Introduction

In recent years, dynamic graphs have attracted substantial research interest, owing to the fact that many real-world graph-structured data are inherently dynamic, such as those found in social networks (Shu et al. 2017; Kumar, Zhang, and Leskovec 2019; Alvarez-Rodriguez et al. 2021; Liu et al. 2020), recommender systems (Fan et al. 2019; Zhang et al. 2022; Gao et al. 2022), and traffic networks (Wang et al. 2020; Bui, Cho, and Yi 2022; Sharma et al. 2023). Although graph neural networks (GNNs) (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2017) have achieved remarkable progress on static graphs (Liu et al. 2021, 2023) with fixed nodes and edges, they cannot be directly extended to dynamic graphs (Barros et al. 2021; Kazemi et al. 2020), where nodes and edges evolve over time. In dynamic graphs, complex temporal dependencies and structural changes arise, posing challenges that exceed the modeling capabilities of conventional static GNNs. As a result, there is a growing demand for dynamic graph learning models that can effectively capture temporal evolution, which enables more precise modeling and analysis of complex, time-evolving systems in practical applications.

Early methods for dynamic graph learning often struggle to capture fine-grained temporal dynamics. Most approaches fail to model the precise order and timing of interactions (Sankar et al. 2020; Cong et al. 2022; You, Du, and Leskovec 2022). To address these limitations, recent years have witnessed advanced models, including sequence-based methods (Wang et al. 2021a; Cong et al. 2023; Yu et al. 2023; Peng, Wei, and Ye 2025), temporal random walk methods (Wang et al. 2021b; Li et al. 2023; Lu et al. 2024), and memory-based methods (Rossi et al. 2020; Su, Zou, and Wu 2024; Sheng et al. 2024; Ji et al. 2024). Although these approaches have advanced the modeling of temporal dynamics, each still exhibits inherent limitations when applied to the *dynamic link prediction* task, which is the primary focus of this work. Specifically, **sequence-based methods** build interaction sequences for the two endpoints of a candidate link and encode them using sequential models such as Transformer-based (Vaswani et al. 2017) or RNN-based (Schuster and Paliwal 1997) architectures. While effective for modeling temporal dependencies, those models (Yu et al. 2023) are limited by the sequence length and may fail to capture long-range dynamics. **Temporal random walk methods** utilize time-respecting paths to explore higher-order neighborhoods and capture contextual information beyond immediate interactions. However, to remain computationally tractable, those methods require truncating the walk length, inevitably discarding portions of the historical context. As a result, they may overlook informative yet temporally distant interactions that could be crucial for predicting future link formation (Wang et al. 2021b). **Memory-based methods** address the limitations of the aforementioned sequence-based and temporal random walk methods by maintaining a persistent memory state for each node, which is continuously updated through recurrent mechanisms to capture long-term dependency information. While

*Corresponding author.

effective in preserving node-level historical context, they often lack explicit representations of pairwise relational dynamics (Ji et al. 2024), which are crucial for accurately predicting link formation (Ma et al. 2024). The above limitations motivate us to explore a novel perspective on dynamic graph learning that can simultaneously capture local fine-grained variations and global long-range temporal dependencies, while explicitly modeling pairwise relational dynamics crucial for the dynamic link prediction task.

Graph-as-Video: A New Perspective. To overcome the above limitations, drawing inspiration from the success of computer vision and recent progress in vision-enhanced graph learning (Wei et al. 2024, 2025), we introduce a novel *graph-as-video* perspective. Specifically, large-scale video foundation models (e.g., VideoMAE (Tong et al. 2022; Wang et al. 2023), InternVideo2 (Wang et al. 2024)) have demonstrated the ability to hierarchically disentangle *spatial appearance* from *cross-frame motion*, thereby perceiving both local fine-grained variations and long-range temporal patterns spanning thousands of frames. We observe that the evolving node-edge topology in dynamic graphs is highly isomorphic to the spatio-temporal transformations of pixel distributions in video sequences: a single frame corresponds to a **structural snapshot** of the dynamic graph at a specific time, while cross-frame motions align with the **evolutionary trajectories** of node interactions. Thus, by serializing a candidate link and its spatio-temporal neighborhood into a "graph video", we can effectively inherit the inductive biases of video models for complex temporal regularities at virtually no additional cost. Building on this perspective, we propose **Graph2Video**, a lightweight framework that implements the graph-as-video idea for dynamic graph learning, which can be seamlessly integrated into existing methods.

Graph2Video addresses three major challenges in the dynamic link prediction task. **First**, by rendering temporally ordered subgraph frames, Graph2Video preserves *fine-grained motion cues*: for example, the precise order in which a triangle closes becomes observable as edge appearances across consecutive frames. **Second**, by feeding the entire graph video into a video model optimized for capturing *long-range dependencies*, the model naturally attends to motifs and interaction patterns whose influence may extend across thousands of events. **Third**, the resulting link-level spatio-temporal embedding acts as a **link-centric memory unit**, explicitly encoding the relational dynamics of the target link. This complements memory-based methods, which mainly focus on node-centric memories and often overlook pairwise structures critical for link prediction.

To instantiate Graph2Video, at each time step, we capture the local neighborhood surrounding the two endpoints of a candidate link and render it as a structural frame. In those frames, endpoints are explicitly highlighted, together with the timing of edge formations and the connectivity among their historical neighbors. Arranging these frames chronologically yields a compact "graph video" clip that depicts how the substructure of the candidate link evolves over time. Feeding this video into a frozen video model produces a rich link-level embedding that can be seamlessly integrated into existing dynamic graph encoders. This significantly enhances performance on the dynamic link prediction task without increasing sequence length or random-walk depth, while also addressing the limitations of node-centric memory paradigms in modeling pairwise relational dynamics.

Our main contributions are summarized as follows.

- **Graph2Video Framework.** We propose Graph2Video, a novel and model-agnostic framework that reformulates link-centered dynamic graph neighborhoods as "graph videos", enabling principled modeling of spatio-temporal dynamics. The framework is lightweight and plug-and-play, making it applicable to a wide range of dynamic graph models without architectural modifications.
- **Link-Centric Memory Injection.** We introduce a lightweight mechanism that augments dynamic graph encoders with link-level spatio-temporal embeddings, explicitly capturing pairwise relational dynamics while simultaneously preserving fine-grained temporal cues and long-range dependencies.
- **Empirical Gains.** Extensive experiments on five public benchmarks show that Graph2Video consistently enhances diverse backbones and achieves state-of-the-art performance in dynamic link prediction tasks, demonstrating both its effectiveness and generality.

2 Related Work

2.1 Dynamic Graph Learning

Early efforts in dynamic graph learning struggled to model fine-grained temporal dynamics. Approaches based on GNNs (Xu et al. 2020; Wang et al. 2021a; Yu et al. 2023) or recurrent architectures (Schuster and Paliwal 1997; Vaswani et al. 2017) captured overall structural and sequential patterns but often failed to preserve the exact order and timing of interaction information (Sankar et al. 2020; Cong et al. 2022; You, Du, and Leskovec 2022).

Recent progress falls into three main research lines. Sequence-based methods build chronologically ordered interaction sequences for link endpoints and encode them with RNN-based or Transformer-based architectures to capture both short-term variations and long-range dependencies. Although models such as TCL (Wang et al. 2021a), DyGFormer (Yu et al. 2023), and TIDFormer (Peng, Wei, and Ye 2025) enhance Transformers with dedicated temporal encodings and interaction-aware mechanisms, the quadratic cost of self-attention necessitates short sequence length, limiting their ability to capture very long-range dynamics.

Temporal random-walk methods recover higher-order historical context by sampling time-consistent paths. CAWN (Wang et al. 2021b) aggregates causal anonymous walks; PINT (Souza et al. 2022) performs injective temporal message passing with walk-count positional features; TPNet (Lu et al. 2024) projects temporal walk matrices via random feature propagation. Despite these advances, all three must truncate walk length or branching factor for tractability, inevitably discarding temporally distant yet informative interactions that could reveal future links.

Memory-based methods maintain a persistent trainable state for each node, updated after every event to accumulate long-term dependency information. Early works such as DyRep (Trivedi et al. 2019) and JODIE (Kumar, Zhang, and Leskovec 2019) use recurrent updates, while more recent models like TGN (Rossi et al. 2020) and MemMap (Ji et al. 2024) introduce richer aggregation and hierarchical memory. Although effective, these approaches remain node-centric and lack explicit modeling of pairwise relational dynamics crucial for link prediction.

While these methods have achieved notable success, existing paradigms still face fundamental limitations: Sequence models are constrained by limited sequence lengths, random walk models suffer from sampling truncation, and memory-based models overlook the explicit evolution of pairwise relationships. These limitations motivate the need for new perspectives that can jointly capture local fine-grained variations and global long-range dependencies in dynamic graphs for the link prediction task.

Video Foundation Model Early work relied on handcrafted features or optical flow, followed by 3D CNNs such as I3D and C3D to capture local spatiotemporal patterns, and later Transformer-based architectures like TimeSformer (Bertasius, Wang, and Torresani 2021) and MViT (Fan et al. 2021) for global modeling. More recently, large-scale video foundation models have significantly improved cross-task transferability and generalization. Among them, the VideoMAE series represents the masked autoencoding paradigm. VideoMAE (Tong et al. 2022) randomly masks most spatiotemporal patches and reconstructs the video cube to learn disentangled appearance–motion features. Its successor, VideoMAE V2 (Wang et al. 2023), scales up both the model and the training pipeline, achieving strong performance across benchmarks. Another influential line is the SAM series, such as SAM2 (Ravi et al. 2024), which extends prompt-driven segmentation to videos with a streaming memory mechanism, enabling high-precision multi-frame tracking and segmentation.

As model and data scales grow, the spatiotemporal inductive biases of video foundation models offer natural advantages for dynamic graph modeling. Motivated by this, we propose Graph2Video, which converts dynamic graphs into temporally ordered "graph videos". At each timestamp, we extract the local subgraph, divide it into topological patches, and concatenate these frames to form a temporal graph representation suitable for video models. This enables direct use of masked self-supervised strategies and hierarchical spatiotemporal encoders. Compared with static graph encoders, Graph2Video yields time-aware link representations and naturally incorporates video-inspired memory mechanisms to capture long-range dependencies, providing a new perspective for dynamic link prediction.

3 Preliminary

Dynamic Graph. Given a node set $\mathcal{V}$ and an edge set $\mathcal{E}$, a dynamic graph $\mathcal{G}$ can be represented as a sequence of non-decreasing chronological interactions denoted by $\mathcal{G} = \{(u_1, v_1, t_1), (u_2, v_2, t_2), \cdots, (u_k, v_k, t_k)\}$ with $0 \leq t_1 \leq t_2 \leq \cdots \leq t_k$, where $u_i, v_i \in \mathcal{V}$ are the source and destination nodes of the new occurred link at timestamp t_i, respectively. Each node $v \in \mathcal{V}$ can be associated with node feature $\boldsymbol{x}_v \in \mathbb{R}^{d_V}$, and each interaction $(u, v, t) \in \mathcal{E}_t$ has link feature $\boldsymbol{e}_{u,v}(t) \in \mathbb{R}^{d_E}$. Here, d_V and d_E denote the dimensions of the node feature and the link feature, respectively. For non-attributed graphs, we simply set the node feature and link feature to zero vectors, i.e., $\boldsymbol{x}_u = \mathbf{0}$ and $\boldsymbol{e}_{u,v}(t) = \mathbf{0}$.

Dynamic Graph Link Prediction. Given two nodes u and v, a timestamp t, and the historical interactions prior to t (i.e., $\mathcal{H}_{<t} = \{(u', v', t') \mid t' < t\}$), the goal of dynamic graph link prediction is to learn a function

$$f : (u, v, t, \mathcal{H}_{<t}) \to \{0, 1\}$$

that predicts whether a link between u and v will occur at time t, based on the temporal and structural context encoded in the historical interaction set $\mathcal{H}_{<t}$.

4 Methodology

We introduce Graph2Video, a lightweight framework that injects vision-aware temporal patterns into existing dynamic graph link predictors. As shown in Figure 1, Graph2Video comprises four modular components: (1) *Graph-Video Construction*, (2) *Graph-Motion Extraction*, (3) *Plug-and-Play Dynamic Graph Encoder*, and (4) *Adaptive Feature Fusion*. We now describe each component in detail.

4.1 Graph-Video Construction

We first convert the temporal evolution of a target link (u, v) into a fixed-length "graph video". The key to capturing link dynamics lies in modeling *two complementary scopes*: (i) the **spatial scope**, i.e., which part of the graph is observed around a query link, and (ii) the **temporal scope**, i.e., how the observed structure evolves over time. We therefore construct a *scope-decoupled subgraph video* that isolates spatial and temporal contexts before feeding them into the frozen video backbone.

Temporal Scope (uniform slicing). Given a target link (u, v) and a prediction time t^*, let t_0 be the timestamp of the earliest event incident on u or v. We define the temporal window as the interval $[t_0, t^*]$, which captures all historical patterns that may affect the formation of (u, v). This interval is uniformly divided into F equal-length segments, and the graph snapshots at the end of each segment yield an ordered frame set: $\{\mathcal{G}_{t_1}^{(u,v)}, \ldots, \mathcal{G}_{t_F}^{(u,v)}\}$, with $t_1 < \cdots < t_F = t^*$.

Spatial Scope (local structural context). At each time point t_i, we construct a local subgraph centered on the target link (u, v) by inducing the k-hop temporal neighborhood of each endpoint. For each node $x \in \{u, v\}$, we define $\mathcal{V}_x^k(t_i) = \mathrm{Top}^s\left(N_k(x, t_i)\right)$, where $N_k(x, t_i)$ denotes the multi-set of nodes within k-hop temporal neighborhoods that interacted with x prior to t_i, and $\mathrm{Top}^s(\cdot)$ selects the s most recent neighbors ranked by interaction time. The full node set of the subgraph is then given by $\mathcal{V}_{u,v}^k(t_i) = \{u, v\} \cup \mathcal{V}_u^k(t_i) \cup \mathcal{V}_v^k(t_i)$, and the corresponding induced subgraph is $\mathcal{G}_{t_i}^{(u,v)} = \left(\mathcal{V}_{u,v}^k(t_i),\, \mathcal{E}_{u,v}^k(t_i)\right)$, where $\mathcal{E}_{u,v}^k(t_i)$ includes all edges among nodes in $\mathcal{V}_{u,v}^k(t_i)$ at time t_i.

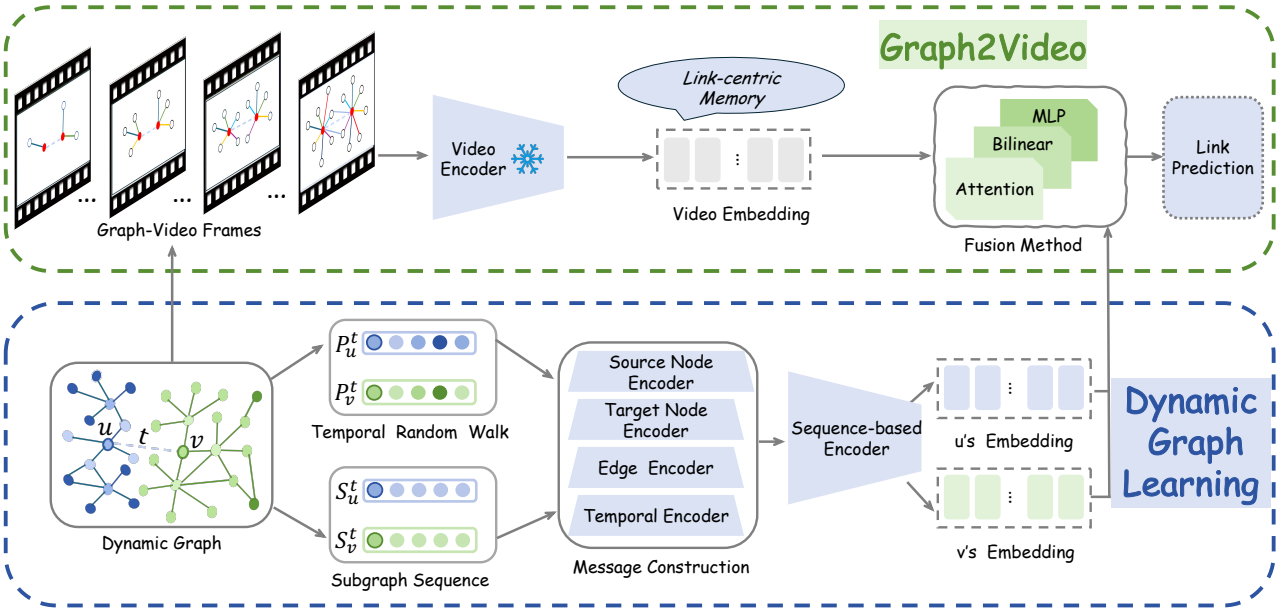

Figure 1: The framework of Graph2Video.

Qualitative Subgraph Visualization To qualitatively verify that our scope-decoupled construction retains meaningful temporal dynamics, we visualize each subgraph frame $\mathcal{G}_{t_i}^{(u,v)}$ using automated graph visualization tools, including `Graphviz` (Gansner and North 2000) and `Matplotlib` (Tosi 2009). This visualization also provides an intuitive view of how local structures evolve over time. To ensure consistent layouts and avoid introducing noise caused by layout variation, we carefully standardize every aspect of the visualization canvas, including the graph layout algorithm, node colors, and node shapes. Specifically, the two endpoints are pinned to predefined anchor positions; visual cues are added by coloring endpoints in red, arranging their respective s most recent neighbors in separate clusters, color-coding edges by interaction time, and inserting dummy placeholder nodes for absent neighbors so that node ordering and layout remain stable across frames. Each resulting high-resolution PNG is converted to RGB and resized to a fixed spatial resolution (H, W), yielding a video tensor $\boldsymbol{X}^{(i)} \in \mathbb{R}^{3\times H\times W}$. Stacking the F frame tensors along the temporal axis produces the final video input $\boldsymbol{X} \in \mathbb{R}^{F\times 3\times H\times W}$, which is fed *unchanged* into the frozen video backbone. Therefore, this visualization pipeline offers interpretability without influencing training or inference and yields a temporally ordered tensor stack that we interpret as a "graph motion sequence", where the appearance or disappearance of edges around (u, v) constitutes the motion. This construction captures a temporally aligned, layout-invariant structural context around the link, facilitating consistent modeling across dynamic snapshots. As shown in Figure 1, the top-left panel labeled "Graph-Video Frames" illustrates the stacked frames. Each frame depicts the 1-hop local subgraph centered on the target link (u, v) across different temporal slices.

4.2 Graph-Motion Extraction

Having constructed scope-decoupled graph videos, the next step is to extract expressive spatio-temporal features that capture the structural dynamics around a candidate link. To this end, we leverage pre-trained video backbones, which are well-suited for recognizing both fine-grained motions and long-range temporal dependencies. However, directly training or fine-tuning such large-scale models on graph videos is often infeasible, motivating a frozen but expressive design.

Graph–Motion Feature Extractor. Each scope-decoupled subgraph video is represented as a video tensor $\boldsymbol{X}$, which we feed into a *pre-trained, frozen* video backbone ϕ^{video}. The backbone produces an embedding

$$\boldsymbol{f}_{uv}^{\text{video}} = \phi^{\text{video}}(\boldsymbol{X}) \in \mathbb{R}^{d_{\text{vid}}},$$

where d_{vid} is the output dimensionality of the chosen backbone. In principle, ϕ^{video} can be instantiated by any off-the-shelf video model that performs spatial–temporal attention (e.g., TimeSformer (Bertasius, Wang, and Torresani 2021), ViViT (Arnab et al. 2021)). In our implementation, we adopt VideoMAE V2 (Wang et al. 2023), a masked-autoencoding Transformer that reconstructs sparsely sampled video cubes; substituting another backbone requires no changes to the rest of the pipeline. Because the video encoder is frozen, without any additional training, the backbone outputs a spatio-temporal embedding $\boldsymbol{f}_{uv}^{\text{video}}$ encoding high-level "structural motions" such as triadic closures or neighbor dispersal patterns around the link (u, v). We regard this embedding as a **link–centric memory unit**: it preserves the spatio–temporal history of the pair (u, v), augments the node–centric states maintained by conventional dynamic graph encoders, supplies the fine–grained and long–range relational cues that those encoders lack.

Frozen but Expressive. Fine-tuning such powerful video models on graph videos is both computationally expensive and prone to overfitting, due to the lack of rich low-level textures in graph-structured inputs. Instead, we freeze the backbone $\phi^{\text{video}} : \mathbb{R}^{F\times 3\times H\times W} \to \mathbb{R}^{d_{\text{vid}}}$ and use it solely as a feature generator. This design brings three notable benefits. First, it improves **efficiency**, as video embeddings are computed once and cached, thereby reducing both training time and memory consumption. Second, it enhances **regularization**, since freezing prevents catastrophic forgetting and preserves the motion-sensitive filters learned from large-scale video corpora. Third, it provides strong **domain robustness**, as shown in our experiments (Section 5), where frozen video features transfer effectively despite the modality gap, thereby validating the "graph-as-video" hypothesis.

4.3 Plug-and-Play Dynamic Graph Encoder

In this work, we focus on learning expressive representations for dynamic link prediction. A key challenge of this task is capturing *long-range temporal dependencies*, which arise not only from extended interaction histories but also from higher-order structural contexts. To tackle this challenge, we adopt two representative dynamic graph encoders, DyGFormer and TPNet, that approach long-term dependencies from complementary perspectives. **DyGFormer** (Yu et al. 2023) is a *sequence-based* encoder that applies time-aware self-attention over chronologically ordered event histories, capturing long-range dependencies through temporal interaction patterns. **TPNet** (Lu et al. 2024) is a *temporal random walk-based* encoder that models higher-order structural dependencies by projecting temporal walk matrices via random feature propagation, avoiding explicit path sampling while maintaining efficiency.

To demonstrate that Graph2Video integrates seamlessly with dynamic graph encoders and boosts their ability to capture long-range dependencies without architectural modifications or additional supervision, we retain the original architectures and hyper-parameter settings of the encoders.

4.4 Adaptive Feature Fusion

Dynamic Link Prediction. Given two nodes u, v and time t, what is the probability that a new interaction will occur (or re-occur) between them? After obtaining the visual subgraph embedding $\boldsymbol{f}_{uv}^{\text{video}}$, we fuse this *vision-aware cue* with the topology-aware node states $\boldsymbol{h}_u(t), \boldsymbol{h}_v(t)$ ($\boldsymbol{h}_u, \boldsymbol{h}_v$ for brevity) produced by the dynamic graph encoder, then feed the fused embeddings to a standard link decoder to estimate this probability denoted as $\widehat{p}_{uv}(t)$.

Let $\boldsymbol{h}_x \in \mathbb{R}^d$ with $x \in \{u, v\}$ be the node embeddings and $\boldsymbol{f}_{uv}^{\text{video}} \in \mathbb{R}^{d_{\text{vid}}}$ the visual feature. We project every modality into a common d-dimensional space:

$$\tilde{\boldsymbol{h}}_u = \boldsymbol{W}_u \boldsymbol{h}_u, \quad \tilde{\boldsymbol{h}}_v = \boldsymbol{W}_v \boldsymbol{h}_v, \quad \tilde{\boldsymbol{f}} = \boldsymbol{W}_f \boldsymbol{f}_{uv}^{\text{video}}.$$

We provide three plug-and-play modules, *attention-guided*, *bilinear*, and *MLP*, that produce vision-enhanced node embeddings $\widehat{\boldsymbol{y}}_x \in \mathbb{R}^d$, $x \in \{u, v\}$ without altering the underlying dynamic graph backbone.

- **Attention-guided.** Cross-modal multi-head attention aligns $\tilde{\boldsymbol{h}}_x$ with $\tilde{\boldsymbol{f}}$, then mixes them via a learnable gate $\alpha = \psi(\theta)$:

  $$\boldsymbol{q}_x = \text{FFN}\big(\text{Attn}(\tilde{\boldsymbol{h}}_x, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{f}})\big), \ \widehat{\boldsymbol{y}}_x = (1-\alpha)\tilde{\boldsymbol{h}}_x + \alpha\, \boldsymbol{q}_x.$$

- **Bilinear.** A tensor $\boldsymbol{W} \in \mathbb{R}^{d\times d\times d}$ captures second-order interactions:

  $$\widehat{\boldsymbol{y}}_x^{(k)} = \tilde{\boldsymbol{h}}_x^{\top} \boldsymbol{W}_k \tilde{\boldsymbol{f}} + \boldsymbol{b}_k,$$

  where $\boldsymbol{W}_k \in \mathbb{R}^{d\times d}$ is the k-th slice and $\boldsymbol{b}_k$ is a bias term.

- **MLP.** A lightweight two-layer perceptron on the concatenation $[\tilde{\boldsymbol{h}}_x \parallel \tilde{\boldsymbol{f}}]$:

  $$\widehat{\boldsymbol{y}}_x = \sigma\Big(\boldsymbol{W}_2 \cdot \sigma\Big(\boldsymbol{W}_1[\tilde{\boldsymbol{h}}_x \parallel \tilde{\boldsymbol{f}}] + \boldsymbol{b}_1\Big) + \boldsymbol{b}_2\Big).$$

Edge predictor. Finally, any differentiable decoder $g(\cdot,\cdot)$ (e.g., dot-product or MLP) converts the fused embeddings into a link probability:

$$\widehat{p}_{uv}(t) = \pi\big(g(\widehat{\boldsymbol{y}}_u, \widehat{\boldsymbol{y}}_v)\big),$$

where π is the sigmoid. Because only the input embeddings are upgraded with vision cues, the downstream link prediction layer remains unchanged.

5 Experiments

5.1 Experimental Settings

Datasets and Baselines. We conduct experiments on five real-world datasets: Reddit, MOOC, Enron, Can. Parl., and UCI, collected by (Poursafaei et al. 2022) which cover diverse domains. To validate the effectiveness of our model, we compare against six popular dynamic graph learning baselines which are based on various techniques, including memory networks (i.e., TGN (Rossi et al. 2020)), random walks (i.e., CAWN (Wang et al. 2021b) and TPNet (Lu et al. 2024)), MLP models (i.e., Graph-Mixer (Cong et al. 2023)), and sequential models (i.e., TCL (Wang et al. 2021a) and DyGFormer (Yu et al. 2023)). In addition, we integrate our approach into DyGFormer and TPNet to demonstrate the effectiveness and plug-and-play adaptability of Graph2Video. We refer to them as DyGFormer+ and TPNet+. Details about datasets and baselines can be found in Appendix A.

Evaluation Tasks and Metrics. The evaluation of our experiment centers on the dynamic prediction task followed by prior works (Wu et al. 2019; Rossi et al. 2020; Wang et al. 2021b; Poursafaei et al. 2022). We utilize two settings: a transductive setting, where the objective is to predict future links between nodes observed during training, and an inductive setting, aiming to predict future links involving previously unseen nodes. Average Precision (AP) and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) are chosen as the evaluation metrics. Similar to (Poursafaei et al. 2022), three negative sampling strategies (NSS) are used to evaluate our model, i.e., random (rnd), historical (hist), and inductive (ind). The latter two strategies are more challenging due to their inherent complexities, see details in (Poursafaei et al. 2022). We perform a chronological split of the dataset, assigning 70% of the data to training, 15% to validation, and 15% to testing.

NSS	Datasets	TGN	CAWN	TCL	GraphMixer	DyGFormer	DyGFormer+	TPNet	TPNet+
rnd	Reddit	$98.63_{\pm 0.06}$	$99.11_{\pm 0.01}$	$97.53_{\pm 0.02}$	$97.31_{\pm 0.01}$	$99.22_{\pm 0.01}$	$99.23_{\pm 0.01}$	$\underline{99.27_{\pm 0.00}}$	$\mathbf{99.32}_{\pm 0.01}$
	MOOC	$89.15_{\pm 1.60}$	$80.15_{\pm 0.25}$	$82.38_{\pm 0.24}$	$82.78_{\pm 0.15}$	$87.52_{\pm 0.49}$	$87.33_{\pm 0.43}$	$\mathbf{96.39}_{\pm 0.09}$	$\underline{96.38_{\pm 0.03}}$
	Enron	$86.53_{\pm 1.11}$	$89.56_{\pm 0.09}$	$79.70_{\pm 0.71}$	$82.25_{\pm 1.02}$	$92.47_{\pm 0.12}$	$92.32_{\pm 0.18}$	$\underline{92.90_{\pm 0.17}}$	$\mathbf{93.01}_{\pm 0.10}$
	UCI	$92.34_{\pm 1.04}$	$95.18_{\pm 0.06}$	$89.57_{\pm 1.63}$	$93.25_{\pm 0.57}$	$95.79_{\pm 0.17}$	$96.24_{\pm 0.04}$	$\underline{97.35_{\pm 0.04}}$	$\mathbf{97.37}_{\pm 0.10}$
	Can. Parl.	$70.28_{\pm 2.34}$	$69.82_{\pm 2.34}$	$68.67_{\pm 2.67}$	$77.04_{\pm 0.96}$	$97.36_{\pm 0.45}$	$\underline{97.60_{\pm 0.40}}$	$90.28_{\pm 0.37}$	$\mathbf{98.70}_{\pm 0.13}$
	Avg. rank	5.60	6.00	7.60	6.40	3.60	3.40	2.20	1.20
hist	Reddit	$81.22_{\pm 0.61}$	$80.82_{\pm 0.45}$	$77.14_{\pm 0.16}$	$78.44_{\pm 0.18}$	$\underline{81.57_{\pm 0.67}}$	$81.41_{\pm 1.6}$	$81.02_{\pm 1.31}$	$\mathbf{82.19}_{\pm 1.40}$
	MOOC	$87.06_{\pm 1.93}$	$74.05_{\pm 0.95}$	$77.06_{\pm 0.41}$	$77.77_{\pm 0.92}$	$85.85_{\pm 0.66}$	$87.56_{\pm 0.89}$	$\underline{92.69_{\pm 0.95}}$	$\mathbf{92.94}_{\pm 0.17}$
	Enron	$73.91_{\pm 1.76}$	$64.73_{\pm 0.36}$	$70.66_{\pm 0.39}$	$77.98_{\pm 0.92}$	$75.63_{\pm 0.73}$	$77.46_{\pm 0.72}$	$80.79_{\pm 1.68}$	$\mathbf{82.52}_{\pm 0.84}$
	UCI	$80.43_{\pm 2.12}$	$65.30_{\pm 0.43}$	$80.25_{\pm 2.74}$	$84.11_{\pm 1.35}$	$82.17_{\pm 0.82}$	$82.55_{\pm 1.13}$	$\underline{86.34_{\pm 0.80}}$	$\mathbf{86.57}_{\pm 0.79}$
	Can. Parl.	$68.42_{\pm 3.07}$	$66.53_{\pm 2.77}$	$65.93_{\pm 3.00}$	$74.34_{\pm 0.87}$	$97.00_{\pm 0.31}$	$\underline{97.46_{\pm 0.33}}$	$86.61_{\pm 3.47}$	$\mathbf{97.66}_{\pm 0.31}$
	Avg. rank	5.20	7.40	7.40	4.80	4.00	3.20	3.00	1.00
ind	Reddit	$88.10_{\pm 0.24}$	$\mathbf{91.67}_{\pm 0.24}$	$87.45_{\pm 0.29}$	$85.26_{\pm 0.11}$	$91.11_{\pm 0.40}$	$\underline{91.41_{\pm 0.71}}$	$88.19_{\pm 0.33}$	$88.62_{\pm 1.10}$
	MOOC	$77.50_{\pm 2.91}$	$73.51_{\pm 0.94}$	$74.65_{\pm 0.54}$	$74.27_{\pm 0.92}$	$81.24_{\pm 0.69}$	$82.24_{\pm 1.20}$	$\underline{88.18_{\pm 0.97}}$	$\mathbf{88.22}_{\pm 0.69}$
	Enron	$70.89_{\pm 2.72}$	$75.15_{\pm 0.58}$	$71.29_{\pm 0.32}$	$75.01_{\pm 0.19}$	$\underline{77.41_{\pm 0.89}}$	$\mathbf{78.60}_{\pm 0.89}$	$75.36_{\pm 1.81}$	$77.03_{\pm 0.86}$
	UCI	$70.94_{\pm 0.71}$	$64.61_{\pm 0.48}$	$76.01_{\pm 1.11}$	$\mathbf{80.10}_{\pm 0.51}$	$72.25_{\pm 1.71}$	$72.35_{\pm 1.78}$	$77.26_{\pm 1.57}$	$\underline{78.48_{\pm 1.27}}$
	Can. Parl.	$65.34_{\pm 2.87}$	$67.75_{\pm 1.00}$	$65.85_{\pm 1.75}$	$69.48_{\pm 0.63}$	$95.44_{\pm 0.57}$	$\underline{96.39_{\pm 0.54}}$	$85.59_{\pm 3.08}$	$\mathbf{98.76}_{\pm 0.11}$
	Avg. rank	6.80	5.60	6.20	5.40	3.60	2.60	3.60	2.20

Table 1: AP (%) for *transductive* dynamic link prediction on real-world datasets with three sampling strategies (NSS).

NSS	Datasets	TGN	CAWN	TCL	GraphMixer	DyGFormer	DyGFormer+	TPNet	TPNet+
rnd	Reddit	$97.50_{\pm 0.07}$	$98.62_{\pm 0.01}$	$94.09_{\pm 0.07}$	$95.26_{\pm 0.02}$	$98.84_{\pm 0.02}$	$\underline{98.86_{\pm 0.02}}$	$98.86_{\pm 0.01}$	$\mathbf{98.95}_{\pm 0.03}$
	MOOC	$89.04_{\pm 1.17}$	$81.42_{\pm 0.24}$	$82.38_{\pm 0.24}$	$81.41_{\pm 0.21}$	$86.96_{\pm 0.43}$	$86.92_{\pm 0.28}$	$\underline{95.07_{\pm 0.26}}$	$\mathbf{95.10}_{\pm 0.10}$
	Enron	$77.94_{\pm 1.02}$	$86.35_{\pm 0.51}$	$79.70_{\pm 0.71}$	$75.88_{\pm 0.48}$	$89.76_{\pm 0.34}$	$89.88_{\pm 0.12}$	$\mathbf{90.34}_{\pm 0.28}$	$\underline{90.18_{\pm 0.27}}$
	UCI	$88.12_{\pm 2.05}$	$92.73_{\pm 0.06}$	$89.57_{\pm 1.63}$	$91.19_{\pm 0.42}$	$94.54_{\pm 0.12}$	$94.89_{\pm 0.04}$	$\underline{95.74_{\pm 0.05}}$	$\mathbf{95.83}_{\pm 0.16}$
	Can. Parl.	$70.88_{\pm 2.34}$	$69.82_{\pm 2.34}$	$68.67_{\pm 2.67}$	$77.04_{\pm 0.46}$	$87.74_{\pm 0.71}$	$\underline{87.89_{\pm 1.01}}$	$68.09_{\pm 1.55}$	$\mathbf{95.50}_{\pm 0.49}$
	Avg. rank	5.80	5.60	6.80	6.60	3.80	3.00	3.00	1.20
hist	Reddit	$64.85_{\pm 0.85}$	$63.67_{\pm 0.41}$	$60.83_{\pm 0.25}$	$64.50_{\pm 0.26}$	$\mathbf{65.37}_{\pm 0.60}$	$\underline{65.07_{\pm 0.91}}$	$62.15_{\pm 1.72}$	$64.88_{\pm 2.21}$
	MOOC	$77.07_{\pm 3.41}$	$74.68_{\pm 0.68}$	$77.06_{\pm 0.41}$	$74.00_{\pm 0.97}$	$80.82_{\pm 0.30}$	$\underline{82.17_{\pm 1.31}}$	$81.85_{\pm 1.60}$	$\mathbf{83.96}_{\pm 0.91}$
	Enron	$62.91_{\pm 1.16}$	$60.70_{\pm 0.36}$	$70.66_{\pm 0.39}$	$72.37_{\pm 1.37}$	$67.07_{\pm 0.62}$	$67.20_{\pm 1.19}$	$\underline{74.60_{\pm 1.35}}$	$\mathbf{76.19}_{\pm 1.14}$
	UCI	$70.78_{\pm 0.78}$	$64.54_{\pm 0.47}$	$\underline{80.25_{\pm 0.39}}$	$\mathbf{81.66}_{\pm 0.49}$	$72.13_{\pm 1.87}$	$72.23_{\pm 1.49}$	$78.48_{\pm 1.18}$	$79.57_{\pm 1.33}$
	Can. Parl.	$68.42_{\pm 3.07}$	$66.53_{\pm 2.77}$	$65.93_{\pm 3.00}$	$74.34_{\pm 0.87}$	$87.40_{\pm 0.85}$	$\underline{87.54_{\pm 0.57}}$	$68.97_{\pm 1.60}$	$\mathbf{95.20}_{\pm 0.57}$
	Avg. rank	5.80	7.20	5.60	4.20	4.00	3.20	4.20	1.80
ind	Reddit	$64.84_{\pm 0.84}$	$63.65_{\pm 0.41}$	$60.81_{\pm 0.26}$	$64.49_{\pm 0.25}$	$\mathbf{65.35}_{\pm 0.60}$	$64.66_{\pm 0.12}$	$62.14_{\pm 1.72}$	$\underline{64.88_{\pm 2.22}}$
	MOOC	$77.07_{\pm 3.40}$	$74.69_{\pm 0.68}$	$74.65_{\pm 0.54}$	$73.99_{\pm 0.97}$	$80.82_{\pm 0.30}$	$\underline{82.17_{\pm 1.30}}$	$81.85_{\pm 1.60}$	$\mathbf{83.96}_{\pm 0.91}$
	Enron	$62.90_{\pm 1.16}$	$60.72_{\pm 0.36}$	$71.29_{\pm 0.32}$	$72.37_{\pm 1.38}$	$67.07_{\pm 0.62}$	$67.20_{\pm 1.19}$	$\underline{74.60_{\pm 1.35}}$	$\mathbf{76.19}_{\pm 1.14}$
	UCI	$70.73_{\pm 0.79}$	$64.54_{\pm 0.47}$	$76.01_{\pm 1.11}$	$\mathbf{81.64}_{\pm 0.49}$	$72.13_{\pm 1.86}$	$72.28_{\pm 1.49}$	$78.50_{\pm 1.18}$	$\underline{79.52_{\pm 1.56}}$
	Can. Parl.	$65.34_{\pm 2.87}$	$67.75_{\pm 1.00}$	$65.85_{\pm 1.75}$	$69.48_{\pm 0.63}$	$87.22_{\pm 0.82}$	$\underline{87.78_{\pm 0.67}}$	$68.87_{\pm 1.68}$	$\mathbf{95.25}_{\pm 0.56}$
	Avg. rank	6.00	6.80	6.00	4.20	4.00	3.60	4.00	1.40

Table 2: AP (%) for *inductive* dynamic link prediction on real-world datasets with three sampling strategies (NSS).

Implementation Details. To ensure fair and consistent comparisons, we adopt the settings from (Poursafaei et al. 2022); details in Appendix B.

5.2 Performance Comparison and Discussions

Tables 1 and 2 present the AP scores of DyGFormer+ and TPNet+, compared against prior state-of-the-art dynamic graph models across five real-world datasets. All results are reported under transductive and inductive settings, evaluated with three negative-sampling strategies (random, historical, and inductive). For readability, all numerical values are multiplied by 100, with the highest and second-highest results in each column highlighted in **bold** and underline, respectively. Additional results on AUC for both transductive and inductive link prediction tasks are provided in Appendix C.

The results demonstrate that integrating Graph2Video consistently enhances the performance of both backbones across nearly all datasets, sampling strategies, and transduc-

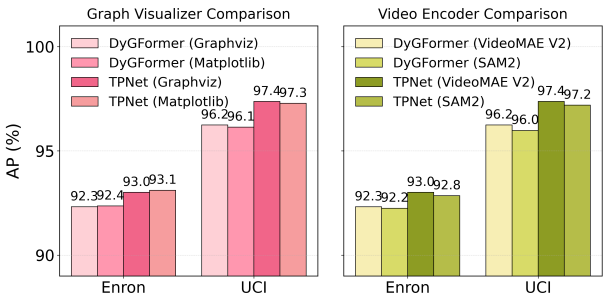

Figure 2: Graph Visualizer and Video Encoder Comparison.

Model	s	Enron	UCI
DyGFormer	8	**92.35**$_{\pm 0.20}$	96.19$_{\pm 0.14}$
	16	92.32$_{\pm 0.18}$	**96.24**$_{\pm 0.04}$
	32	92.31$_{\pm 0.16}$	96.10$_{\pm 0.38}$
TPNet	8	**93.05**$_{\pm 0.18}$	97.23$_{\pm 0.21}$
	16	93.01$_{\pm 0.10}$	**97.37**$_{\pm 0.10}$
	32	93.02$_{\pm 0.19}$	97.30$_{\pm 0.10}$

Table 3: Performance comparison when varying the number of most recent neighbors s in the video frame.

tive/inductive settings, thereby setting new state-of-the-art results on the benchmarks. The improvements are particularly evident under the inductive setting and challenging sampling strategies. For instance, TPNet+ achieves an average AP gain of **5.49%** in the inductive setting with random sampling strategies across five datasets. Notably, under the historical and inductive sampling strategies for inductive link prediction, TPNet+ delivers average improvements of **6.75%** and **6.77%**, respectively, where modeling long-range dependencies is especially challenging. The largest improvement is observed on the Can. Parl. with average AP gains of **10.88%** in the transductive setting and **26.67%** in the inductive setting across three sampling strategies. We attribute this remarkable boost primarily to the coarse-grained annual partitioning scheme of the dataset, which diminishes fine-grained temporal signals and increases reliance on long-range dependencies and higher-order neighborhood overlaps. We also report the average ranks (Avg. rank) of the model's performance. The results show that integrated models consistently outperform the baselines. For example, TPNet+ achieves an average AP rank of 1.47 in both transductive and inductive settings across three sampling strategies. Thus, we can conclude that by capturing structural motions across aggregated frames, Graph2Video effectively restores local variations and global trends, thereby overcoming the limitations of current approaches.

5.3 Ablation Study and Sensitivity Analysis

Graph Visualizer Selection. Figure 2 (left) shows the performance with different graph visualizers. The results demonstrate stable performance across graph visualizers.

Video Encoder Selection. Figure 2 (right) reports the performance with different video encoders. The results indicate that VideoMAE V2 consistently achieves superior effectiveness, owing to its advanced masked tube reconstruction strategy and dual masking design.

Model	Fusion Strategy	Enron	UCI
DyGFormer	MLP	**92.40**$_{\pm 0.12}$	96.23$_{\pm 0.06}$
	Bilinear	91.60$_{\pm 0.29}$	96.15$_{\pm 0.10}$
	Attention	92.32$_{\pm 0.18}$	**96.24**$_{\pm 0.04}$
TPNet	MLP	92.98$_{\pm 0.08}$	97.14$_{\pm 0.24}$
	Bilinear	93.18$_{\pm 0.12}$	97.04$_{\pm 0.21}$
	Attention	**93.01**$_{\pm 0.10}$	**97.37**$_{\pm 0.10}$

Table 4: Fusion strategies performance comparison.

Model	Sam2-F	Enron	UCI
DyGFormer	8	92.22$_{\pm 0.13}$	95.93$_{\pm 0.06}$
	16	92.25$_{\pm 0.13}$	95.98$_{\pm 0.08}$
	32	**92.31**$_{\pm 0.14}$	**96.01**$_{\pm 0.09}$
TPNet	8	92.82$_{\pm 0.09}$	97.16$_{\pm 0.17}$
	16	92.85$_{\pm 0.23}$	97.19$_{\pm 0.29}$
	32	**93.01**$_{\pm 0.11}$	**97.29**$_{\pm 0.12}$

Table 5: Performance comparison when varying the number of Frames F in the Sam2 video model.

s-Most Recent Neighbors. Table 3 reports the performance with varying s most recent neighbors for graph video construction, showing stable results across s values.

Feature Fusion Strategy. Table 4 presents the performance for different feature fusion strategies. The experimental results indicate, in most cases, default attention-guided fusion outperforms both bilinear fusion and MLP fusion.

Number of Frames F. Table 5 shows that larger F consistently yields better performance, confirming that finer temporal granularity benefits Graph2Video.

6 Conclusion

We introduced Graph2Video, a novel "graph-as-video" viewpoint for dynamic graph learning. By serializing the evolving neighborhood of each candidate link into a sequence of structural frames, our framework constructs a compact "graph video" representation that captures both local fine-grained variations and long-range temporal dependencies. This representation is processed through a lightweight, plug-and-play module that derives link-level spatio-temporal embeddings, making the approach broadly applicable to diverse dynamic graph encoders without architectural modifications. Comprehensive experiments on five benchmarks verify that Graph2Video consistently improves strong baselines and achieves state-of-the-art performance in the majority of settings, including diverse sampling strategies and inductive scenarios. Beyond link prediction, our findings indicate a broader opportunity: importing the mature inductive biases of video models can advance other dynamic graph tasks such as temporal motif discovery, anomaly detection, and event forecasting.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under grant number 2022YFA1004102 and National Natural Science Foundation of China (NSFC) under grant number 62136005.

References

Alvarez-Rodriguez, U.; Battiston, F.; de Arruda, G. F.; Moreno, Y.; Perc, M.; and Latora, V. 2021. Evolutionary dynamics of higher-order interactions in social networks. *Nature Human Behaviour*, 5(5): 586–595.

Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; and Schmid, C. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6836–6846.

Barros, C. D.; Mendonça, M. R.; Vieira, A. B.; and Ziviani, A. 2021. A survey on embedding dynamic graphs. *ACM Computing Surveys (CSUR)*, 55(1): 1–37.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is Space-Time Attention All You Need for Video Understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*.

Bui, K.-H. N.; Cho, J.; and Yi, H. 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3): 2763–2774.

Cong, W.; Wu, Y.; Tian, Y.; Gu, M.; Xia, Y.; Mahdavi, M.; and Chen, C.-c. J. 2022. Dynamic graph representation learning via graph transformer networks. *OpenReview*.

Cong, W.; Zhang, S.; Kang, J.; Yuan, B.; Wu, H.; Zhou, X.; Tong, H.; and Mahdavi, M. 2023. Do we really need complicated model architectures for temporal networks? *International Conference on Learning Representations*.

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6824–6835.

Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, 417–426.

Gansner, E. R.; and North, S. C. 2000. An open graph visualization system and its applications to software engineering. *Software: practice and experience*, 30(11): 1203–1233.

Gao, C.; Wang, X.; He, X.; and Li, Y. 2022. Graph neural networks for recommender system. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 1623–1625.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Ji, S.; Liu, M.; Sun, L.; Liu, C.; and Zhu, T. 2024. Memmap: An adaptive and latent memory structure for dynamic graph learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1257–1268.

Kazemi, S. M.; Goel, R.; Jain, K.; Kobyzev, I.; Sethi, A.; Forsyth, P.; and Poupart, P. 2020. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70): 1–73.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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.

Li, Y.; Shen, Y.; Chen, L.; and Yuan, M. 2023. Zebra: When temporal graph neural networks meet temporal personalized pagerank. *Proceedings of the VLDB Endowment*, 16(6): 1332–1345.

Liu, H.; Han, H.; Jin, W.; Liu, X.; and Liu, H. 2023. Enhancing graph representations learning with decorrelated propagation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1466–1476.

Liu, H.; Qu, C.; Niu, Y.; and Wang, G. 2020. The evolution of structural balance in time-varying signed networks. *Future Generation Computer Systems*, 102: 403–408.

Liu, X.; Jin, W.; Ma, Y.; Li, Y.; Liu, H.; Wang, Y.; Yan, M.; and Tang, J. 2021. Elastic graph neural networks. In *International Conference on Machine Learning*, 6837–6849. PMLR.

Lu, X.; Sun, L.; Zhu, T.; and Lv, W. 2024. Improving temporal link prediction via temporal walk matrix projection. *Advances in Neural Information Processing Systems*, 37: 141153–141182.

Ma, L.; Han, H.; Li, J.; Shomer, H.; Liu, H.; Gao, X.; and Tang, J. 2024. Mixture of link predictors on graphs. *Advances in Neural Information Processing Systems*, 37: 16043–16070.

Peng, J.; Wei, Z.; and Ye, Y. 2025. TIDFormer: Exploiting Temporal and Interactive Dynamics Makes A Great Dynamic Graph Transformer. *arXiv preprint arXiv:2506.00431*.

Poursafaei, F.; Huang, S.; Pelrine, K.; and Rabbany, R. 2022. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35: 32928–32941.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; and Bronstein, M. 2020. Temporal graph networks for deep learning on dynamic graphs. *ICML 2020 Workshop on Graph Representation Learning*.

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.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11): 2673–2681.

Sharma, A.; Sharma, A.; Nikashina, P.; Gavrilenko, V.; Tselykh, A.; Bozhenyuk, A.; Masud, M.; and Meshref, H. 2023. A graph neural network (GNN)-based approach for real-time estimation of traffic speed in sustainable smart cities. *Sustainability*, 15(15): 11893.

Sheng, G.; Su, J.; Huang, C.; and Wu, C. 2024. Mspipe: Efficient temporal gnn training via staleness-aware pipeline. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2651–2662.

Shu, K.; Sliva, A.; Wang, S.; Tang, J.; and Liu, H. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1): 22–36.

Souza, A.; Mesquita, D.; Kaski, S.; and Garg, V. 2022. Provably expressive temporal graph networks. *Advances in neural information processing systems*, 35: 32257–32269.

Su, J.; Zou, D.; and Wu, C. 2024. Pres: Toward scalable memory-based dynamic graph neural networks. *arXiv preprint arXiv:2402.04284*.

Tong, Z.; Song, Y.; Wang, J.; and Wang, L. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35: 10078–10093.

Tosi, S. 2009. *Matplotlib for Python developers*, volume 307. Packt Publishing Birmingham, UK.

Trivedi, R.; Farajtabar, M.; Biswal, P.; and Zha, H. 2019. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*.

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.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, L.; Chang, X.; Li, S.; Chu, Y.; Li, H.; Zhang, W.; He, X.; Song, L.; Zhou, J.; and Yang, H. 2021a. Tcl: Transformer-based dynamic graph modelling via contrastive learning. *arXiv preprint arXiv:2105.07944*.

Wang, L.; Huang, B.; Zhao, Z.; Tong, Z.; He, Y.; Wang, Y.; Wang, Y.; and Qiao, Y. 2023. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14549–14560.

Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; Jia, C.; and Yu, J. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*, 1082–1092.

Wang, Y.; Chang, Y.-Y.; Liu, Y.; Leskovec, J.; and Li, P. 2021b. Inductive representation learning in temporal networks via causal anonymous walks. *arXiv preprint arXiv:2101.05974*.

Wang, Y.; Li, K.; Li, X.; Yu, J.; He, Y.; Chen, G.; Pei, B.; Zheng, R.; Wang, Z.; Shi, Y.; et al. 2024. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, 396–416. Springer.

Wei, Y.; Fu, S.; Jiang, W.; Zhang, Z.; Zeng, Z.; Wu, Q.; Kwok, J.; and Zhang, Y. 2024. Gita: Graph to visual and textual integration for vision-language graph reasoning. *Advances in Neural Information Processing Systems*, 37: 44–72.

Wei, Y.; Wang, X.; Zhuang, Z.; Chen, Y.; Chen, S.; Zhang, Y.; Zhang, Y.; and Kwok, J. 2025. Open Your Eyes: Vision Enhances Message Passing Neural Networks in Link Prediction. *arXiv preprint arXiv:2505.08266*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.

Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; and Achan, K. 2020. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*.

You, J.; Du, T.; and Leskovec, J. 2022. ROLAND: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2358–2366.

Yu, L.; Sun, L.; Du, B.; and Lv, W. 2023. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36: 67686–67700.

Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; and Wang, L. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 4741–4753.