Learnable Data Augmentation and Contrastive Pre-training for Temporal Link Prediction

Canghong Jin[1*], **Jiafeng Zhao**[2], **Feng Xu**[1], **Tongya Zheng**[1], **Zemin Liu**[3], **Lina Wei**[1], **Mingli Song**[3]
[1]School of Computer and Computing Science, Hangzhou City University, China
[2]College of Computer Science and Technology, Zhejiang University of Technology, China
[3]College of Computer Science and Technology, Zhejiang University, China
jinch@hzcu.edu.cn,liu.zemin@zju.edu.cn

Abstract

Link prediction is a foundational task in temporal graphs. While temporal graph neural networks exhibit commendable performance, they are often criticized for providing inadequate representations, especially under limited data. Contrastive learning has been introduced as a solution for graph pre-training to mitigate data scarcity. However, the data augmentation techniques on which they depend frequently lead to suboptimal augmentations, manifesting either as over-augmentation or under-augmentation. In light of these challenges, we explore the under-explored domain of contrastive learning of temporal graph transformers and propose a novel model, ContraTGT, which employs a dual-view graph transformer. This transformer is fine-tuned to extract sequences tailored for specific target nodes, encapsulating both spatial and temporal perspectives. To optimize the contrastive learning of this dual-view transformer, we put forth an innovative, learnable data augmentation method. This technique, which involves selective masking of elements within dual-view sequences, generates superior augmentations, thereby amplifying the potency of the contrastive learning approach. Extensive experiments on five public temporal network datasets demonstrate that our model can consistently outperform all baselines, especially in small training set conditions.

1 Introduction

Many real-world networks exhibit dynamic behavior, where interactions or edges carry associated timestamps denoting their occurrence. These are identified as temporal (or dynamic) graphs [Kazemi *et al.*, 2020]. Such dynamism necessitates specialized methods for temporal graph analysis, diverging from traditional graph representation learning techniques [Barros *et al.*, 2021]. Initial endeavors in embedding temporal interaction graphs predominantly yielded static representations [Nguyen *et al.*, 2018; Zhang *et al.*, 2017]. A significant limitation of these approaches is their inability to accommodate newly introduced nodes or edges as graphs progress over time. To address this dynamic element and the need for time-sensitive representations, a new wave of research ushered in the concept of temporal graph neural networks (TGNNs) [Sahili and Awad, 2023], which are crafted to model both temporal nuances and structural dependencies, demonstrating adaptability to emerging nodes and edges [Kumar *et al.*, 2019; Pareja *et al.*, 2020; da Xu *et al.*, 2020; Zhang *et al.*, 2023b; Zhu *et al.*, 2023]. A common theme among these models is the employment of neighborhood aggregation, which dynamically assimilates contextual information essential for node representation learning. Nevertheless, a potential shortcoming is the predominant focus of neighborhood aggregation on immediate contexts and recent history, which may inadvertently sideline the broader global context and invaluable long-term historical insights.

With the advent of graph transformers [Ying *et al.*, 2021; Kreuzer *et al.*, 2021; Dwivedi and Bresson, 2020], a fresh perspective on representation learning in temporal networks has been established. Temporal graph transformers (TGTs) [Cong *et al.*, 2023] have been designed in alignment with traditional practices, primarily utilizing a sequence-based framework, emphasizing the sampling of extensive sequences to encompass both the global context and the historical behaviors associated with a target node. By synergizing these sampled sequences with diverse encodings, TGTs demonstrate a keen ability to identify and interpret prolonged dependencies within the temporal graph. As a testament to their efficacy, TGTs have achieved benchmark performances in a range of temporal graph learning challenges [Wang *et al.*, 2021a; Cong *et al.*, 2023].

The Problem. While Transformer-based approaches have proven effective, they often rely heavily on ample labeled data for model training [Wang *et al.*, 2021a; Cong *et al.*, 2023], which becomes a bottleneck in scenarios with restricted supervision. A promising solution lies in pre-training on graphs, which leverages the intrinsic structures of the graph through a self-supervised approach [Hu *et al.*, 2019; Hu *et al.*, 2020]. Within this context, contrastive learning has emerged as a potent method for pre-training on graphs, particularly for extracting underlying information [Qiu *et al.*, 2020; You *et al.*, 2020; Zhu *et al.*, 2020]. Yet, the application of contrastive learning with graph transformers on temporal graphs

*Corresponding author

remains an under-explored area. This gap is especially pronounced in the realm of temporal graph representation learning under limited supervision. Motivated by this observation, our paper delves into the problem of *contrastive learning of graph transformers on temporal graphs*.

Challenges. Pre-training graph transformers on temporal graphs holds significant promise but remains an under-explored avenue. Within the domain of temporal graph representation learning, the methodology diverges significantly from the commonly explored contrastive learning that utilizes GNN backbones. Unlike these, graph transformers typically operate in a sequence-based fashion. Therefore, a typical practice is to devise a contrastive strategy tailored for sequence-based learning on temporal graphs. Yet, when considering contrastive learning on graphs, prevalent data augmentation techniques [You *et al.*, 2020] like DropEdge [Rong *et al.*, 2020] and DropNode [Feng *et al.*, 2020] often employ random masking of raw data. However, calibrating the degree of such augmentations is complex, potentially leading to excessive (over-augmentation) or insufficient (under-augmentation) alterations. This poses a conundrum: *in the context of contrastive learning on temporal graphs, how to craft an adept contrastive strategy for sequence-based learning that harmoniously integrates an optimal augmentation technique?*

Present work. To address the aforementioned challenges, we introduce a novel framework named ContraTGT, for the Contrastive learning of Temporal Graph Transformers. At its core, ContraTGT leverages a dual-view graph transformer, which is tailored to extract sequences for a given target node, encompassing both spatial and temporal views. Such a design facilitates the assimilation of comprehensive contextual information and traces the historical evolution associated with the target node, thereby refining temporal node representation learning. In tandem with this, to proficiently pre-train the dual-view graph transformer, we architect a sequence-based contrastive strategy, which aligns guarantee with the requirements of contrastive learning tailored for temporal graph transformers. Moreover, we introduce an adaptive, learnable data augmentation technique. By selectively masking elements within the dual-view sequences, this method crafts optimized augmentations, thereby enhancing the efficacy of the contrastive learning process.

We leverage dual-view node sequences sampled from temporal graphs, which are derived from both spatial and temporal perspectives, to formulate our contrastive strategy. Particularly, for each node sequence, we employ node masking—specifically, by dropping nodes within the sampled sequences, to act as the data augmentation. The foundational reason behind this approach is that the sequences, even when subjected to such masking perturbations, retain an approximation of the principal context and historical trends associated with the target node. As a result, these modified sequences can effectively function as augmented counterparts in the contrastive learning process.

While random masking serves its purpose, as highlighted earlier, it lacks precision in controlling the extent of augmentation, which can lead to over-augmentation or under-augmentation, potentially compromising the quality of contrastive learning. Therefore, we introduce a learnable data augmentation mechanism, designed to selectively mask nodes within the dual-view sequences. To steer this learnable data augmentation, we define two essential metrics: *consistency* and *diversity*. The former, consistency, gauges the prediction alignment between augmented and raw data. A higher consistency suggests that the augmentation closely mirrors the key features of the raw data, leading to aligned predictions. Conversely, diversity evaluates the variance between the augmented representation and its raw counterpart. A greater diversity indicates a more desirable augmentation. This rationale stems from the idea that, beyond capturing primary behaviors, diversity encompasses supplementary or secondary data features—essentially, the minor perturbations. These perturbations help to amplify the distinctions between augmented and raw data, aiding in the label propagation from labeled to unlabeled data. Collectively, these metrics allow for meticulous control over the noise introduced, ensuring optimal augmentations tailored for contrastive learning in temporal graphs.

Contributions. In summary, our contributions can be outlined as follows. We tackle the novel research problem of contrastive learning for graph transformers applied to temporal graphs, which represents a significant and under-explored domain. To address this challenge, we introduce a novel model ContraTGT, designed specifically for the contrastive learning of temporal graph transformers. ContraTGT not only samples sequences from both spatial and temporal perspectives but also incorporates a learnable data augmentation mechanism to generate optimized augmentations for these sequences. We conduct extensive experiments on five real-world datasets, showcasing the superior performance of ContraTGT when compared to state-of-the-art baseline methods.

2 Related Work

Graph Representation Learning. The evolution of graph embedding techniques, ranging from early structural mapping [Grover and Leskovec, 2016; Tang *et al.*, 2015] to message-passing paradigms [Hamilton *et al.*, 2017; Veličković *et al.*, 2018], has laid the foundation for modern graph analysis. To capture the evolutionary nature of real-world networks, Dynamic Graph Embeddings (DGEs) extend these concepts into discrete-time snapshots [Pareja *et al.*, 2020; Sankar *et al.*, 2020; Zhang *et al.*, 2023a] and continuous-time event streams [Zhu *et al.*, 2023; Zhang *et al.*, 2023c; Wang *et al.*, 2021c]. While effective, traditional DGEs often struggle to balance fine-grained temporal resolution with global structural dependencies. Recently, Graph Transformers have emerged to capture long-range interactions through self-attention mechanisms [Dwivedi and Bresson, 2020; Kreuzer *et al.*, 2021]. To mitigate the quadratic computational overhead of full attention, sequence-based architectures like TCL [Wang *et al.*, 2021a] and DyFormer [Cong *et al.*, 2023] have been proposed. Building on this, CoLA-Former [Zhao *et al.*, 2025] further refines temporal modeling by integrating to enhance scalability. Parallelly, Graph-Mamba [Wang *et al.*, 2024] introduces the State-Space

Model (SSM) paradigm to the graph domain, leveraging selective scan mechanisms to achieve linear-complexity modeling of long-sequence dependencies, thereby surpassing the efficiency bottlenecks of conventional Transformers.

Self-Supervised Learning and Augmentation. Graph Contrastive Learning (GCL) has become a dominant paradigm for learning perturbation-invariant representations without manual labels [Hu *et al.*, 2019; Qiu *et al.*, 2020; You *et al.*, 2020]. In temporal contexts, TCL [Wang *et al.*, 2021a] utilizes time-depth encodings for link-level contrast, while DyFormer [Cong *et al.*, 2023] employs dual-view sequence contrasting. However, these methods typically rely on random sampling or fixed heuristics, which may overlook underlying temporal trends or lead to suboptimal augmentation—specifically, the issues of over-augmentation or under-augmentation [Bo *et al.*, 2022]. To address these limitations, several advanced augmentation strategies have been explored. For instance, JOAO [You *et al.*, 2021] adopts an adversarial framework to select optimal pairs from a predefined pool, though it often neglects representation consistency. Similarly, GCA [Zhu *et al.*, 2021] introduces adaptive augmentation based on node centrality, yet its dropping mechanism remains static and non-learnable during training. Unlike these approaches, our proposed ContraTGT framework devises a learnable data augmentation strategy guided by both consistency and diversity, specifically optimized for capturing the intricate evolutionary patterns in temporal graphs.

3 Preliminaries

Graph and Temporal Graph. A static graph is defined as $G = \{V, E, \mathbf{X}\}$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, and $\mathbf{X} \in \mathbb{R}^{|V| \times d_X}$ denotes the node feature matrix. A **temporal graph** incorporates temporal dynamics, represented as $G = \{V, E, \mathbf{X}, T, \phi, \varphi\}$. Here, T denotes the time domain, while $\phi : (V, T) \to \{0, 1\}$ and $\varphi : (E, T) \to \{0, 1\}$ serve as indicators for the existence of nodes and edges at a specific timestamp $t \in T$, respectively.

Problem Formulation. Given a node v at timestamp t, we aim to derive a representation $\mathbf{h}_{(v,t)}$ (simplified as $\mathbf{h}_v$) by leveraging its contextual and historical information. We focus on the **dynamic link prediction** task. For an interaction (v, u) at time t, the existence probability $p^t_{(v,u)}$ is calculated as:

$$p^t_{(v,u)} = \text{MLP}(\mathbf{h}_v \,\|\, \mathbf{h}_u), \tag{1}$$

where $\|$ denotes the concatenation operation. The model is optimized using binary cross-entropy loss over a set of training tuples T_{tr}:

$$\mathcal{L}_{\text{tr}} = -\sum_{(v,u,t)\in T_{\text{tr}}} \left(\log p^t_{(v,u)} + \mathbb{E}_{x\sim\mathcal{U}_v} \log(1 - p^t_{(v,x)}) \right), \tag{2}$$

where $\mathcal{U}_v$ denotes the uniform distribution for negative sampling. To address scenarios with *limited supervision*, we employ self-supervised contrastive learning to capture underlying structural nuances.

Graph Transformers. The Transformer architecture [Vaswani *et al.*, 2017] utilizes multi-head self-attention (MSA) to model dependencies within a sequence $\mathbf{H}^{(l-1)} = [\mathbf{h}^{(l-1)}_0, \ldots, \mathbf{h}^{(l-1)}_{n-1}] \in \mathbb{R}^{n \times d_{l-1}}$. In layer l, the input is projected into Query, Key, and Value matrices:

$$\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)} = \mathbf{H}^{(l-1)}\mathbf{W}^{(l)}_Q, \mathbf{H}^{(l-1)}\mathbf{W}^{(l)}_K, \mathbf{H}^{(l-1)}\mathbf{W}^{(l)}_V, \tag{3}$$

where $\mathbf{W}^{(l)}_{Q,K} \in \mathbb{R}^{d_{l-1} \times d^K_l}$ and $\mathbf{W}^{(l)}_V \in \mathbb{R}^{d_{l-1} \times d^V_l}$. The attention output is computed via:

$$\text{Att}(\mathbf{H}^{(l-1)}) = \text{Softmax}(\mathbf{Q}^{(l)}{\mathbf{K}^{(l)}}^\top / \sqrt{d^K_l})\mathbf{V}^{(l)}. \tag{4}$$

The hidden state is updated through a Feed-Forward Network (FFN) with Layer Normalization (LN) and residual connections:

$$\mathbf{H}^{(l)} = \text{LN}\big(\text{FFN}(\text{LN}(\text{Att}(\mathbf{H}^{(l-1)}) + \mathbf{H}^{(l-1)})) + \tilde{\mathbf{H}}^{(l)}\big). \tag{5}$$

Stacked L layers produce the final sequence representation $\mathbf{H}^{(L)}$ for downstream tasks. We denote the parameter set of the Transformer as θ_t.

4 The Proposed Model: ContraTGT

As illustrated in Fig. 1, ContraTGT consists of three main stages: (1) dual-view sequence sampling (Sect. 4.1), (2) contrastive pre-training with learnable data augmentation (Sect. 4.2), and (3) fine-tuning for downstream tasks (Sect. 4.3).

4.1 Dual-View Sequence Sampling

To circumvent the quadratic complexity of global attention on large graphs, we adopt a sequence sampling strategy to capture local context and historical dynamics.

Spatial Sampling. For a target node v, we sample its l_s nearest neighbors based on interaction timestamps. These neighbors are arranged in descending order of time, as recent interactions typically exert more influence. For node v_0 in Fig. 1(b), the spatial sequence is $(v_0, v_1, v_3, v_2, v_4)$. For each node, the time interval Δt is computed relative to the current timestamp t_{now} and transformed via time encoding [da Xu *et al.*, 2020].

Temporal Sampling. Temporal sampling captures global trends preceding the target node's latest interaction. If v's latest interaction occurred at t_{last}, we sample l_t interactions immediately preceding t_{last} in the graph. The involved nodes are chronologically ordered to form a sequence, e.g., $(v_0, v_1, v_4, v_2, v_7)$ in Fig. 1(b). This allows the model to capture historical trajectories beyond the immediate spatial neighborhood.

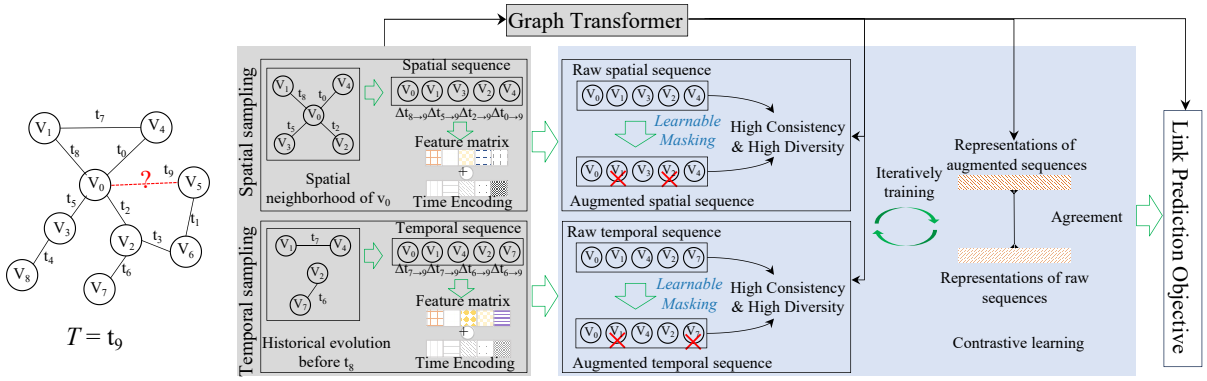
4.2 Pre-Training with Contrastive Learning

We propose a contrastive learning strategy driven by *learnable data augmentation* to avoid the sub-optimal views generated by random masking.

Consistency and Diversity. Guided by [Bo *et al.*, 2022], we optimize augmentations $\hat{x} = \Psi(x; \theta_\Psi)$ using two metrics:

Consistency: Ensures the prediction $f(\mathbf{h}_{\hat{x}})$ remains invariant compared to $f(\mathbf{h}_x)$.

Diversity: Maximizes the representation distance $d(\mathbf{h}_x, \mathbf{h}_{\hat{x}})$ to provide a challenging view.



(a) Input temporal graph (b) Dual-view sequence sampling (c) Pre-training with contrastive learning (d) Supervised loss

Figure 1: The overall framework of ContraTGT. (a) Temporal graph input. (b) Dual-view sequence sampling for target node v_0. (c) Pre-training with learnable masking based on consistency and diversity. (d) Fine-tuning for link prediction between v_0 and v_5.

Learnable Data Augmentation. We implement a learnable masking module Ψ_{lm} that adaptively selects nodes to mask within a sequence. Given a representation sequence S, the augmented features $\hat{s}_x$ are:

$$\hat{s}_x = \text{SharPen}(\text{Atten}(S, v); \rho) \odot s_x, \tag{6}$$

where $\text{Atten}(\cdot)$ calculates attention weights via a parameter matrix $\mathbf{W}$, and $\text{SharPen}(\cdot)$ yields a gradient-preserving binary mask based on the masking rate ρ.

Using a link prediction head $p_{(v,u)}$, we define the augmentation loss $\mathcal{L}_{\text{aug}}$ as a weighted sum of consistency ($\mathcal{L}_{\Psi c}$) and diversity ($\mathcal{L}_{\Psi d}$):

$$\mathcal{L}_{\Psi c} = -\sum(p_{(v,u)} \log p_{(\hat{v},u)} + (1 - p_{(v,u)}) \log(1 - p_{(\hat{v},u)})), \tag{7}$$

$$\mathcal{L}_{\Psi d} = -\sum d(\mathbf{h}_v, \mathbf{h}_{\hat{v}}), \tag{8}$$

$$\mathcal{L}_{\text{aug}} = \mathcal{L}_{\Psi c} + \alpha \mathcal{L}_{\Psi d}. \tag{9}$$

Contrastive Objective. The Transformer is pre-trained by maximizing the agreement between a node v and its augmentation $\hat{v}$, while minimizing it with a negative instance u:

$$\mathcal{L}_{\text{con}} = -\sum \log \frac{\exp(\mathbf{h}_v \cdot \mathbf{h}_{\hat{v}})}{\exp(\mathbf{h}_v \cdot \mathbf{h}_{\hat{v}}) + \exp(\mathbf{h}_v \cdot \mathbf{h}_u)}. \tag{10}$$

Optimization alternates between $\mathcal{L}_{\text{aug}}$ (updating $\theta_{\Psi_{lm}}$) and $\mathcal{L}_{\text{con}}$ (updating θ_t).

4.3 Objective

For downstream tasks, the representations from spatial and temporal views $(\mathbf{H}_s, \mathbf{H}_t)$ are concatenated: $\mathbf{h}_v = \mathbf{H}_s[v] \,\|\, \mathbf{H}_t[v]$. The final model is fine-tuned using supervised binary cross-entropy (Eq. 2).

4.4 Algorithms and Complexity Analysis

Pre-training Paradigm. Algorithm 1 details the iterative training process. The function $\text{NRL}(\cdot)$ (Algorithm 2) handles the dual-view encoding and learnable masking.

Complexity Analysis. Pre-training complexity is dominated by the Transformer encoding, which is $O(B(l_s^2 d + l_t^2 d))$ per batch. Since sequence lengths l_s, l_t are fixed, the model scales linearly with the number of nodes, ensuring efficiency on large-scale temporal graphs.

Algorithm 1 Pre-Training of ContraTGT

Require: $G, l_s, l_t, \alpha, \rho$;
Ensure: $\theta_t, \theta_{\Psi_{lm}}$.
while not converged **do**
  Sample batch $B \in G$;
  for each interaction $(v, u, t) \in B$ **do**
    $\mathbf{h}_v, \mathbf{h}_{\hat{v}} = \text{NRL}(v); \mathbf{h}_u = \text{NRL}(u)$;
    Compute $p_{(v,u)}, p_{(\hat{v},u)}$ via Eq. (1);
    $\mathcal{L}_{\text{aug}} \leftarrow \mathcal{L}_{\text{aug}} + \mathcal{L}_{\Psi c} + \alpha \mathcal{L}_{\Psi d}$; ▷ Eq. (9)
    Sample negative q and compute $\mathbf{h}_q$;
    $\mathcal{L}_{\text{con}} \leftarrow \mathcal{L}_{\text{con}} +$ Eq. (10);
  Update $\theta_{\Psi_{lm}}$ by $\min \mathcal{L}_{\text{aug}}$ (fix θ_t);
  Update θ_t by $\min \mathcal{L}_{\text{con}}$ (fix $\theta_{\Psi_{lm}}$);

Algorithm 2 Node Representation Learning (NRL)

Require: Node v;
Ensure: $\mathbf{h}_v, \mathbf{h}_{\hat{v}}$.
$s_v^s, s_v^t \leftarrow$ Spatial/Temporal Sampling for v;
$S_v^s, S_v^t \leftarrow$ Transformer encoding of s_v^s, s_v^t;
$\hat{s}_v^s, \hat{s}_v^t \leftarrow \Psi_{lm}(S_v^s), \Psi_{lm}(S_v^t)$;
$\mathbf{h}_v = \text{TF}(s_v^s) \| \text{TF}(s_v^t)$;
$\mathbf{h}_{\hat{v}} = \text{TF}(\hat{s}_v^s) \| \text{TF}(\hat{s}_v^t)$;
return $\mathbf{h}_v, \mathbf{h}_{\hat{v}}$.

The SharPen Function. Algorithm 3 implements the gradient-preserving Top-K selection mechanism.

5 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of ContraTGTon five benchmark datasets.

5.1 Experimental Setup

Datasets

We evaluate our approach using five publicly available benchmark temporal networks: **soc-wiki-elec**[1], **ia-slashdot-reply-**

[1]https://networkrepository.com/soc-wiki-elec.php

Algorithm 3 SharPen Function

Require: $\mathbf{h}, K, \epsilon$;
Ensure: $\mathbf{f}$.
$\mathbf{m} \leftarrow \mathbf{0}, \mathbf{f} \leftarrow \mathbf{1}$;
for $i = 1$ to K **do**
  $\mathbf{w} \leftarrow 1 - \text{mask}_{prev}$; $\mathbf{y} \leftarrow \text{SoftMax}(\mathbf{h} + \log(\mathbf{w} + \epsilon))$;
  $\mathbf{m} \leftarrow \mathbf{m} + \mathbf{y} * \mathbf{w}$; $\mathbf{f} \leftarrow \mathbf{f} \cdot \mathbf{m}$;
return $\mathbf{f}$;

	# Nodes	# Interactions	# Repetition
soc-wiki-elec	7,118	107,079	4.2%
ia-slashdot-reply-dir	51,083	140,778	0.2%
soc-sign-bitcoinotc	5,881	35,592	0.0%
Social Evolve 1_month	74	176,088	-
Ubuntu	159,316	964,437	-

Table 1: Summary of datasets.

dir[2], **soc-sign-bitcoinotc**[3], **Social Evolve 1_month** [Madan *et al.*, 2011] and **Ubuntu**[4]. Table 1 provides insights into the ratio of nodes that share multiple interactions (edges), where "-" denotes that the statistics is not provided by the authors.

For every dataset, interactions are randomly divided in a 1:1:8 ratio for training, validation, and testing.

Baselines

To comprehensively evaluate our proposed ContraTGT against state-of-the-art approaches, we consider several state-of-the-art baselines categorized into three main groups[5]: (1) *Static GNN models*: GraphSAGE [Hamilton *et al.*, 2017]; (2) *Temporal GNN models*: JOIDE [Kumar *et al.*, 2019], CAWN [Wang *et al.*, 2021c], TIP-GNN [Zheng *et al.*, 2022], TAP-GNN [Zheng *et al.*, 2023]; (3) *Approaches with advanced techniques*: DyRep [Trivedi *et al.*, 2019], TGAT [da Xu *et al.*, 2020], APAN [Wang *et al.*, 2021b], TGN [Rossi *et al.*, 2020], TIGER [Zhang *et al.*, 2023d], and FTM [Cao *et al.*, 2023]; and (4) *Graph Transformer approaches*: TCL [Wang *et al.*, 2021a] and DyGFormer [Yu *et al.*, 2023], and CoLA-Former [Zhao *et al.*, 2025]; and (5) *State Space Models*: Graph-Mamba [Wang *et al.*, 2024].

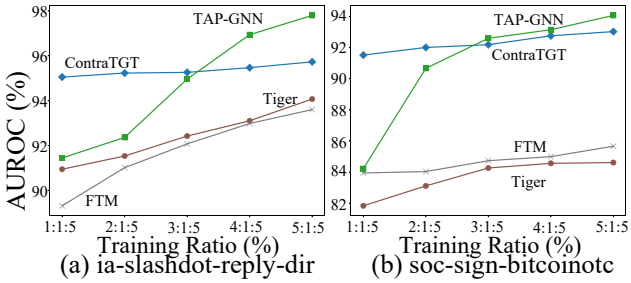
5.2 Performance Evaluation

In this part, we evaluate the performance of ContraTGT on the task of temporal link prediction on temporal graphs.

Evaluation with limited training data

In this experiment, we limit the training data to only 10% of the interactions to simulate scenarios with limited supervision. We conduct the experiments under both transductive and inductive settings. In the transductive setting, the model is evaluated on the entire test dataset, while the inductive setting focuses on predicting interactions involving at least one new node that did not appear in the training data.

[2]https://networkrepository.com/ia-slashdot-reply-dir.php
[3]https://networkrepository.com/soc-sign-bitcoinotc.php
[4]https://snap.stanford.edu/data/sx-askubuntu.html
[5]For the temporal baselines, we only consider continuous-time models.



(a) ia-slashdot-reply-dir (b) soc-sign-bitcoinotc

Figure 2: Evaluation with more training data.

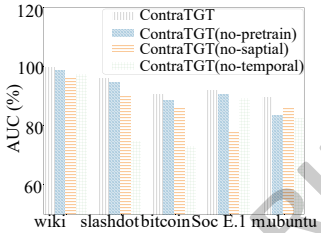


Figure 3: Ablation study.

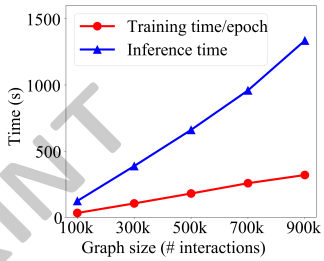


Figure 4: Scalability study.

The performance comparison in Table 2 and Table 3 presents results for the transductive and inductive settings, respectively. Our ContraTGToutperforms all competing approaches on Wiki, Slashdot, and Bitcoin-OTC datasets, while achieving slightly inferior performance to Graph-Mamba and CoLa-Former on Ubuntu dataset. Graph-Mamba benefits from effective long-sequence state modeling, while CoLa-Former leverages semantic consistency through contrastive learning.

Evaluation with more training data

To further assess the efficacy of our proposed model with an increased volume of training data, we executed additional experiments, varying the quantity of training data while keeping the validation and test sets constant at a ratio of n:1:5, where $n = \{1, 2, 3, 4, 5\}$ represents the proportion of the training data. These experiments were carried out on two datasets (ia-slashdot-reply-dir and soc-sign-bitcoinotc), and we selected the most competitive baselines for comparison. The results, with AUC as metric in a transductive setting, are depicted in Fig. 2.

We have the following observations. (1) In scenarios with limited supervision, such as when the proportion of training data is 1 or 2, our proposed model, ContraTGT, surpasses the baselines, underscoring the strength of our model enhanced by pre-training with learnable data augmentation. (2) As the amount of training data increases, the baselines begin to demonstrate comparable, or in some cases, superior performance. This indicates that with ample training data at their disposal, end-to-end models like TAP-GNN, Tiger, and FTM are capable of fully leveraging label information to optimize their models. Typically, these models can surpass those that rely on pre-training when sufficient data is provided, at which point our model still maintains superior performance.

Methods	soc-wiki-elec			ia-slashdot-reply-dir			soc-sign- bitcoinotc			Social E. 1 m.			ubuntu		
	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc
GraphSAGE	0.589±0.017	0.588±0.011	0.556±0.009	0.600±0.008	0.596±0.012	0.576±0.011	0.577±0.005	0.575±0.009	0.533±0.013	0.540±0.009	0.540±0.012	0.516±0.014	0.544±0.013	0.531±0.008	0.515±0.011
JODIE	0.548±0.007	0.543±0.006	0.532±0.007	0.663±0.022	0.659±0.021	0.620±0.018	0.680±0.013	0.690±0.016	0.632±0.017	0.764±0.003	0.692±0.003	0.722±0.005	0.774±0.017	0.810±0.008	0.695±0.006
CAWN	0.788±0.013	0.743±0.019	0.761±0.007	0.892±0.008	0.904±0.010	0.801±0.010	0.831±0.004	0.866±0.003	0.741±0.012	0.871±0.030	0.875±0.003	0.721±0.005	0.774±0.018	0.811±0.008	0.695±0.006
TIP-GNN	0.747±0.009	0.606±0.020	0.63±0.027	0.809±0.031	0.634±0.034	0.651±0.037	0.843±0.010	0.706±0.011	0.775±0.011	0.841±0.003	0.702±0.009	0.773±0.005	0.795±0.010	0.814±0.006	0.807±0.008
TAP-GNN	0.952±0.008	0.952±0.010	0.924±0.022	0.902±0.001	0.877±0.007	0.806±0.021	0.801±0.008	0.774±0.025	0.731±0.016	0.771±0.003	0.673±0.002	0.731±0.003	0.847±0.004	0.846±0.008	0.812±0.005
DyRep	0.538±0.011	0.535±0.009	0.527±0.008	0.659±0.012	0.658±0.019	0.615±0.010	0.688±0.008	0.692±0.011	0.634±0.015	0.719±0.037	0.676±0.038	0.652±0.034	0.763±0.003	0.794±0.003	0.695±0.008
TGAT	0.763±0.014	0.745±0.018	0.681±0.017	0.836±0.004	0.878±0.002	0.787±0.001	0.766±0.005	0.785±0.009	0.704±0.011	0.626±0.010	0.625±0.011	0.579±0.009	0.817±0.024	0.853±0.017	0.756±0.026
APAN	0.832±0.009	0.845±0.010	0.797±0.006	0.798±0.004	0.781±0.011	0.715±0.008	0.767±0.001	0.766±0.004	0.720±0.006	0.745±0.012	0.730±0.007	0.707±0.006	0.731±0.011	0.782±0.004	0.660±0.004
TGN	0.867±0.012	0.881±0.010	0.767±0.012	0.864±0.015	0.878±0.014	0.663±0.008	0.766±0.006	0.801±0.006	0.691±0.006	0.904±0.003	0.882±0.006	0.850±0.003	0.832±0.006	0.865±0.004	0.770±0.004
TIGER	0.832±0.011	0.845±0.014	0.697±0.009	0.942±0.005	0.949±0.002	0.805±0.006	0.763±0.011	0.806±0.008	0.694±0.014	0.925±0.002	0.909±0.004	0.887±0.006	0.831±0.005	0.882±0.003	0.610±0.012
FTM	0.861±0.023	0.876±0.021	0.755±0.019	0.918±0.006	0.933±0.004	0.802±0.004	0.793±0.009	0.820±0.011	0.721±0.005	0.923±0.002	0.904±0.004	0.854±0.002	0.852±0.004	0.881±0.001	0.786±0.001
TCL	0.777±0.032	0.761±0.026	0.684±0.026	0.927±0.009	0.927±0.010	0.732±0.037	0.778±0.004	0.782±0.015	0.703±0.010	0.929±0.004	0.914±0.005	0.885±0.005	0.835±0.003	0.867±0.003	0.786±0.005
DyGFormer	0.706±0.015	0.655±0.019	0.654±0.002	0.885±0.008	0.898±0.004	0.727±0.018	0.822±0.011	0.859±0.009	0.756±0.010	0.930±0.009	0.910±0.006	**0.899**±0.012	0.813±0.010	0.845±0.008	0.810±0.007
Graph-Mamba	0.991±0.001	0.992±0.001	0.979±0.002	0.953±0.006	**0.964**±0.004	0.891±0.034	0.852±0.017	0.885±0.013	0.796±0.017	0.934±0.002	0.917±0.003	0.864±0.014	0.912±0.005	0.937±0.004	0.862±0.007
CoLA-Former	0.987±0.001	0.989±0.001	0.966±0.003	0.930±0.001	0.949±0.001	0.892±0.009	0.870±0.008	0.900±0.008	0.801±0.020	0.890±0.007	0.858±0.010	0.776±0.024	**0.913**±0.002	**0.938**±0.002	**0.864**±0.004
ContraTGT	**0.998**±0.001	**0.998**±0.001	**0.980**±0.001	**0.955**±0.003	0.959±0.003	**0.897**±0.003	**0.902**±0.008	**0.904**±0.011	**0.826**±0.039	**0.934**±0.007	**0.919**±0.009	0.862±0.005	0.861±0.007	0.894±0.009	0.822±0.005

Table 2: Results of small training set(10%) link prediction task (in percent with std; best bolded and runner-up underlined).

5.3 Model Analysis

We analyze ContraTGTfrom four perspectives: ablation, scalability, augmentation comparison, and parameter sensitivity.

Ablation study

To evaluate the contribution of each component in ContraTGT, we conduct an ablation study by comparing with several degenerate variants: (1) *without pretrain*: we use the ContraTGT architecture without the pre-training step; (2) *no spatial*: we remove the spatial-view sequences, by only using the temporal-view sequences for node representation learning. (3)*no temporal*: we remove the temporal-view sequences, by only using the spatial-view sequences for node representation learning.

We show the results of ablation study (using AUC as the metric) in Fig. 3, and make the following observations. Firstly, the performance consistently drops when the model pretraining step is omitted. This finding underscores the efficacy of the proposed contrastive learning strategy, which is significantly enhanced by the learnable data augmentation. Secondly, when the spatial sequences are omitted, ContraTGT exhibits a notable performance decline. This underscores the indispensable nature of spatial information for node representation learning within transformers applied to temporal graphs. Thirdly, the absence of temporal sequences also leads to a general performance decrease. This demonstrates the value of incorporating temporal sampling into ContraTGT. Notably, both spatial and temporal sequences seem to contribute comparably to the overall performance.

Scalability

To assess scalability via experiments, we utilized the largest temporal graph in our study, namely Ubuntu. We sampled several temporal graphs with increasing sizes, specifically 100k, 300k, 500k, 700k, and 900k interactions, to assess their training and inference times. For each temporal graph, we maintained the same data split as in our main experiment, with a 1:1:8 ratio for the training, validation, and testing sets. The training (per epoch) and inference times are reported in Fig. 4. Our observations reveal that both training and inference times for our proposed ContraTGT, scale linearly with the increase in temporal graph size. This linear scalability underscores ContraTGT's capability to be efficiently applied to large-scale temporal graphs, matching the complexity analysis in Section 4.4.

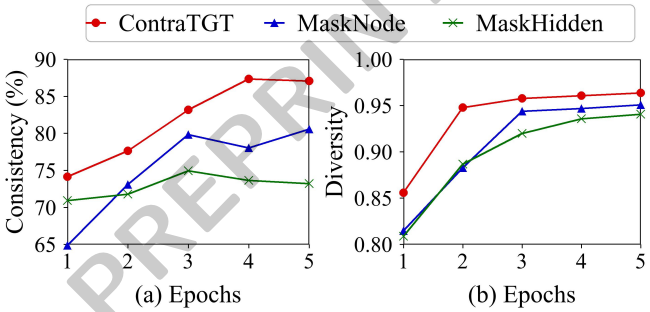


Figure 5: Augmentation Comparison on Consistency and Diversity

Comparison of different augmentations

To assess the effectiveness of our proposed learnable data augmentation method, integrated with the graph transformer featuring dual-view sequence sampling, we introduce two simple yet efficient random masking methods: MaskNode and MaskHidden. For MaskNode, we randomly mask a few nodes in a sampled node sequence according to the predefined masking rate ρ and then compute its representation. Meanwhile, for MaskHidden, we first compute the representation of the entire sequence and then randomly mask a few dimensions with respect to the predefined masking rate ρ, akin to the dropout mechanism. We compare the performance of these three approaches on two representative datasets, presenting the results in Table 4. Note that, for each dataset, we tune the masking rate ρ in the range of [0.1, 0.7] for each method to achieve optimal performance. We observe that our proposed model generally outperforms the other two random masking methods.

We further compare different data augmentation methods using consistency and diversity as evaluation metrics, where higher values indicate better performance. Our learnable data augmentation is compared with MaskNode and MaskHidden on the Bitcoinotc dataset, and the results are shown in Fig. 5. We observe that, as training advances, it becomes ev-

Methods	soc-wiki-elec			ia-slashdot-reply-dir			soc-sign-bitcoinotc			Social E. 1 m.			ubuntu		
	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc
GraphSAGE	0.596±0.012	0.595±0.017	0.569±0.016	0.593±0.012	0.582±0.009	0.574±0.011	0.554±0.011	0.559±0.009	0.536±0.007	0.540±0.009	0.530±0.011	0.509±0.008	0.544±0.007	0.531±0.009	0.509±0.014
JODIE	0.548±0.009	0.550±0.007	0.532±0.010	0.638±0.017	0.641±0.016	0.608±0.012	0.679±0.016	0.692±0.020	0.633±0.022	0.818±0.003	0.755±0.006	0.761±0.005	0.762±0.007	0.796±0.008	0.682±0.005
CAWN	0.961±0.003	0.965±0.005	0.893±0.015	0.845±0.012	0.863±0.012	0.743±0.014	0.748±0.015	0.798±0.009	0.684±0.010	0.903±0.012	0.883±0.024	0.859±0.016	0.796±0.078	0.835±0.065	0.745±0.081
TIP-GNN	0.694±0.007	0.593±0.019	0.631±0.013	0.773±0.005	0.616±0.009	0.637±0.006	0.821±0.012	0.676±0.008	0.737±0.010	0.812±0.011	0.685±0.009	0.743±0.013	0.775±0.007	0.776±0.006	0.757±0.008
DyRep	0.532±0.016	0.538±0.012	0.523±0.011	0.633±0.009	0.639±0.016	0.603±0.008	0.689±0.006	0.697±0.008	0.636±0.020	0.716±0.039	0.687±0.038	0.651±0.042	0.746±0.002	0.780±0.004	0.685±0.010
TGAT	0.785±0.014	0.761±0.019	0.699±0.023	0.823±0.004	0.869±0.002	0.776±0.005	0.790±0.007	0.714±0.016	0.620±0.012	0.630±0.012	0.559±0.004	0.806±0.027	0.805±0.027	0.844±0.018	0.747±0.028
APAN	0.832±0.009	0.844±0.006	0.797±0.006	0.784±0.004	0.775±0.004	0.707±0.007	0.774±0.005	0.768±0.006	0.721±0.020	0.752±0.005	0.738±0.008	0.718±0.007	0.726±0.007	0.776±0.004	0.649±0.008
TGN	0.886±0.008	0.896±0.007	0.802±0.056	0.867±0.017	0.880±0.016	0.692±0.007	0.773±0.006	0.809±0.006	0.701±0.006	0.909±0.007	0.881±0.011	0.852±0.005	0.824±0.007	0.859±0.004	0.763±0.004
TIGER	0.832±0.010	0.845±0.012	0.696±0.008	0.939±0.050	0.947±0.003	0.804±0.007	0.763±0.014	0.806±0.008	0.694±0.016	**0.952**±0.002	**0.938**±0.004	**0.918**±0.007	0.826±0.004	0.878±0.009	0.609±0.014
FTM	0.878±0.017	0.888±0.015	0.760±0.015	0.919±0.006	0.934±0.004	0.807±0.002	0.799±0.011	0.826±0.012	0.728±0.004	0.928±0.003	0.908±0.001	0.849±0.003	0.846±0.004	0.862±0.019	0.779±0.001
CoLA-Former	0.992±0.006	0.992±0.005	0.972±0.009	0.930±0.036	0.950±0.026	**0.902**±0.042	0.863±0.043	0.886±0.037	0.803±0.030	0.928±0.065	0.908±0.087	0.867±0.094	**0.931**±0.028	**0.948**±0.018	**0.867**±0.017
ContraTGT	**0.997**±0.001	**0.998**±0.001	**0.981**±0.001	**0.955**±0.003	**0.960**±0.003	0.895±0.004	**0.899**±0.008	**0.901**±0.012	**0.824**±0.040	0.943±0.007	0.935±0.010	0.877±0.008	0.857±0.007	0.892±0.010	0.820±0.006

Table 3: Inductive learning task results for link prediction.

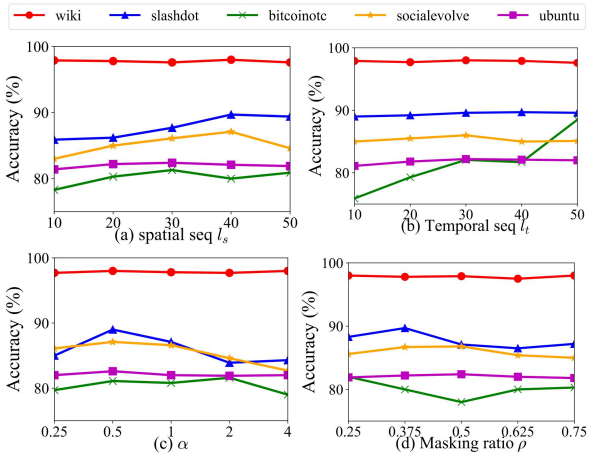


Figure 6: Influence of parameters.

Methods	slashdot			Social E. 1m		
	AUC	AP	Acc	AUC	AP	Acc
MaskNode	0.944±0.003	0.947±0.002	0.866±0.005	0.929±0.003	0.912±0.005	0.853±0.005
MaskHidden	0.937±0.001	0.955±0.006	0.881±0.008	0.914±0.010	0.897±0.006	**0.889**±0.006
ContraTGT	**0.954**±0.003	**0.956**±0.002	**0.891**±0.004	**0.938**±0.004	**0.924**±0.004	0.877±0.006

Table 4: Compare with different augmentations.

ident that our model persistently surpasses the random masking techniques, namely MaskNode and MaskHidden, securing markedly enhanced consistency and diversity. This trend highlights the efficacy of our learnable data augmentation strategy, which evidently produces more advantageous data augmentations when contrasted with random masking methods. The superiority of these data augmentations plays a pivotal role in amplifying the success of our contrastive learning approach. Moreover, the progressive improvement of our proposed model with each epoch is noteworthy. In contrast, the performance trend of the random masking approaches appears more variable, particularly in terms of consistency.

Parameters sensitivity

We evaluate the sensitivity of several important hyperparameters in ContraTGT, and show their impact in Fig. 6. For the length of spatial and temporal sequences, *i.e.*, l_s and l_t, the performance seems to be stable w.r.t.the length, while longer sequences might be slightly better. For the weight of α in Eq. (9), it seems that a smaller α can benefit the performance, showing that consistency is more important in determining the performance. For the masking ratio ρ, the performance seems to be stable across different ρ's. In particular, $\rho \in [0.25, 0.50]$ might be a good choice.

The smooth performance across hyperparameters can be attributed to three factors. (1) The effectiveness of our learnable data augmentation, driven by these metrics, ensures that the model is pre-trained with the most informative augmentations even with different l_s and l_t, contributing to the smooth performance across varying sequence lengths. (2) The smoothness in the masking ratio may be attributed to the effectiveness of the learnable data augmentation mechanism. This mechanism is designed to selectively choose the most informative elements for augmentation, even under extreme masking ratios, leading to promising augmentation generation. (3) For the wiki dataset, the graph Transformer backbone armed with dual-view sequences achieves nearly 100% performance, leaving limited room for significant improvement. As for the Ubuntu dataset, its relatively large size and the utilization of 10% interactions for training may already provide sufficient data for training the graph Transformer.

6 Conclusions

In this work, we introduced a dual-view graph transformer called ContraTGTto extract sequences tailored for specific target nodes, encapsulating both spatial and temporal perspectives. To optimize the contrastive learning of this dual-view transformer, we put forth an innovative, learnable data augmentation method, which involves selective masking of elements within dual-view sequences, generates superior augmentations, thereby amplifying the potency of the contrastive learning approach. Extensive experiments on five public temporal network datasets demonstrate that our model can consistently outperform all baselines, especially in small training set conditions.

Acknowledgements

This research was funded by the Natural Science Foundation of Zhejiang Province of China under Grant (No. LMS26F020043,LHZSD24F020001,LZ25F020012).

References

[Barros *et al.*, 2021] Claudio DT Barros, Matheus RF Mendonça, Alex B Vieira, and Artur Ziviani. A survey

on embedding dynamic graphs. *ACM Computing Surveys (CSUR)*, 55(1):1–37, 2021.

[Bo *et al.*, 2022] Deyu Bo, BinBin Hu, Xiao Wang, Zhiqiang Zhang, Chuan Shi, and Jun Zhou. Regularizing graph neural networks via consistency-diversity graph augmentations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3913–3921, 2022.

[Cao *et al.*, 2023] Bowen Cao, Qichen Ye, Weiyuan Xu, and Yuexian Zou. Ftm: A frame-level timeline modeling method for temporal graph representation learning. *arXiv preprint arXiv:2302.11814*, 2023.

[Cong *et al.*, 2023] Weilin Cong, Yanhong Wu, Yuandong Tian, Mengting Gu, Yinglong Xia, Chun-cheng Jason Chen, and Mehrdad Mahdavi. Dyformer: A scalable dynamic graph transformer with provable benefits on generalization ability. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 442–450. SIAM, 2023.

[da Xu *et al.*, 2020] da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020.

[Dwivedi and Bresson, 2020] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[Feng *et al.*, 2020] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22092–22103. Curran Associates, Inc., 2020.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[Hu *et al.*, 2019] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[Hu *et al.*, 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867, 2020.

[Kazemi *et al.*, 2020] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *The Journal of Machine Learning Research*, 21(1):2648–2720, 2020.

[Kreuzer *et al.*, 2021] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

[Kumar *et al.*, 2019] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1269–1278, 2019.

[Madan *et al.*, 2011] Anmol Madan, Manuel Cebrian, Sai Moturu, Katayoun Farrahi, et al. Sensing the" health state" of a community. *IEEE Pervasive Computing*, 11(4):36–45, 2011.

[Nguyen *et al.*, 2018] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *Companion proceedings of the the web conference 2018*, pages 969–976, 2018.

[Pareja *et al.*, 2020] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5363–5370, 2020.

[Qiu *et al.*, 2020] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1150–1160, 2020.

[Rong *et al.*, 2020] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Rossi *et al.*, 2020] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

[Sahili and Awad, 2023] Zahraa Al Sahili and Mariette Awad. Spatio-temporal graph neural networks: A survey. *arXiv preprint arXiv:2301.10569*, 2023.

[Sankar *et al.*, 2020] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 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*, pages 519–527, 2020.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings*

of the 24th international conference on world wide web, pages 1067–1077, 2015.

[Trivedi *et al.*, 2019] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2021a] Lu Wang, Xiaofu Chang, Shuang Li, Yunfei Chu, Hui Li, Wei Zhang, Xiaofeng He, Le Song, Jingren Zhou, and Hongxia Yang. Tcl: Transformer-based dynamic graph modelling via contrastive learning. *arXiv preprint arXiv:2105.07944*, 2021.

[Wang *et al.*, 2021b] Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 international conference on management of data*, pages 2628–2638, 2021.

[Wang *et al.*, 2021c] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. *arXiv preprint arXiv:2101.05974*, 2021.

[Wang *et al.*, 2024] Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024.

[Ying *et al.*, 2021] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[You *et al.*, 2021] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021.

[Yu *et al.*, 2023] Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *NeurIPS*, 2023.

[Zhang *et al.*, 2017] Yao Zhang, Yun Xiong, Xiangnan Kong, and Yangyong Zhu. Learning node embeddings in interaction graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 397–406, 2017.

[Zhang *et al.*, 2023a] Kaike Zhang, Qi Cao, Gaolin Fang, Bingbing Xu, Hongjian Zou, Huawei Shen, and Xueqi Cheng. Dyted: Disentangled representation learning for discrete-time dynamic graph. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3309–3320, 2023.

[Zhang *et al.*, 2023b] Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Siu Ming Yiu, and Ruihua Han. Spatial-temporal graph learning with adversarial contrastive adaptation. In *International Conference on Machine Learning*, pages 41151–41163. PMLR, 2023.

[Zhang *et al.*, 2023c] Siwei Zhang, Yun Xiong, Yao Zhang, Yiheng Sun, Xi Chen, Yizhu Jiao, and Yangyong Zhu. Rdgsl: Dynamic graph representation learning with structure learning. *arXiv preprint arXiv:2309.02025*, 2023.

[Zhang *et al.*, 2023d] Yao Zhang, Yun Xiong, Yongxiang Liao, Yiheng Sun, Yucheng Jin, Xuehao Zheng, and Yangyong Zhu. Tiger: Temporal interaction graph embedding with restarts. In *Proceedings of the ACM Web Conference 2023*, pages 478–488, 2023.

[Zhao *et al.*, 2025] Zhongying Zhao, Jinyu Zhang, Chuanxu Jia, Chao Li, Yanwei Yu, and Qingtian Zeng. Cola-former: Graph transformer using communal linear attention for lightweight sequential recommendation. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI-25)*, pages 3689–3697, 2025.

[Zheng *et al.*, 2022] Tongya Zheng, Zunlei Feng, Tianli Zhang, Yunzhi Hao, Mingli Song, Xingen Wang, Xinyu Wang, Ji Zhao, and Chun Chen. Transition propagation graph neural networks for temporal networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Zheng *et al.*, 2023] Tongya Zheng, Xinchao Wang, Zunlei Feng, Jie Song, Yunzhi Hao, Mingli Song, Xingen Wang, Xinyu Wang, and Chun Chen. Temporal aggregation and propagation graph neural networks for dynamic representation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

[Zhu *et al.*, 2021] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080, 2021.

[Zhu *et al.*, 2023] Yifan Zhu, Fangpeng Cong, Dan Zhang, Wenwen Gong, Qika Lin, Wenzheng Feng, Yuxiao Dong, and Jie Tang. Wingnn: Dynamic graph neural networks with random gradient aggregation window. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3650–3662, 2023.