Dynamic Graph Transformer with Correlated Spatial-Temporal Positional Encoding

Zhe Wang
Zhejiang University
College of Computer Science
Hangzhou, China
zhewangcs@zju.edu.cn

Sheng Zhou*
Zhejiang University
School of Software Technology
Ningbo, China
zhousheng_zju@zju.edu.cn

Jiawei Chen
Zhejiang University
The State Key Laboratory of
Blockchain and Data Security
Hangzhou, China
sleepyhunt@zju.edu.cn

Zhen Zhang
National University of Singapore
Singapore, Singapore
zhen@nus.edu.sg

Binbin Hu
Ant Group
Hangzhou, China
bin.hbb@antfin.com

Yan Feng
Zhejiang University
The State Key Laboratory of
Blockchain and Data Security
Hangzhou, China
fengyan@zju.edu.cn

Chun Chen
Zhejiang University
The State Key Laboratory of
Blockchain and Data Security
Hangzhou, China
chenc@zju.edu.cn

Can Wang
Zhejiang University
The State Key Laboratory of
Blockchain and Data Security
Hangzhou, China
wcan@zju.edu.cn

ABSTRACT

Learning effective representations for Continuous-Time Dynamic Graphs (CTDGs) has garnered significant research interest, largely due to its powerful capabilities in modeling complex interactions between nodes. A fundamental and crucial requirement for representation learning in CTDGs is the appropriate estimation and preservation of proximity. However, due to the sparse and evolving characteristics of CTDGs, the spatial-temporal properties inherent in high-order proximity remain largely unexplored. Despite its importance, this property presents significant challenges due to the computationally intensive nature of personalized interaction intensity estimation and the dynamic attributes of CTDGs. To this end, we propose a novel Correlated Spatial-Temporal Positional encoding that incorporates a parameter-free personalized interaction intensity estimation under the weak assumption of the Poisson Point Process. Building on this, we introduce the **D**ynamic **G**raph **T**ransformer with **Cor**related Spatial-Temporal Positional Encoding (**CorDGT**), which efficiently retains the evolving spatial-temporal high-order proximity for effective node representation learning in CTDGs. Extensive experiments on seven small and two large-scale datasets demonstrate the superior performance and scalability of the proposed CorDGT. The code is available at: https://github.com/wangz3066/CorDGT.

CCS CONCEPTS

- **Theory of computation → Dynamic graph algorithms**.

KEYWORDS

Continuous-Time Dynamic Graph, Graph Representation Learning, Transformer, Proximity

ACM Reference Format:
Zhe Wang, Sheng Zhou, Jiawei Chen, Zhen Zhang, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. 2025. Dynamic Graph Transformer with Correlated Spatial-Temporal Positional Encoding. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM '25), March 10–14, 2025, Hannover, Germany*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3701551.3703489

1 INTRODUCTION

Graph Neural Networks (GNNs) [6, 10, 29, 31, 41] have become a potent tool for analyzing diverse graph structures due to their effectiveness in learning low-dimensional graph representations. While early GNN research focused on static graphs, many real-world network data exhibit evolving graph structures, such as the World Wide Web and recommendation systems [3, 30, 38]. The increasing prevalence of dynamic graph data has spurred researchers to adapt GNNs to handle dynamic graphs. Dynamic graphs can be categorized into Discrete-Time Dynamic Graphs (DTDGs) and Continuous-Time Dynamic Graphs (CTDGs). Recently, CTDGs have garnered more research attention compared to DTDGs, attributed to their flexible

*Corresponding author.

WSDM '25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03…$15.00
https://doi.org/10.1145/3701551.3703489

generalization capabilities and proficiency in modeling intricate and dynamic node interactions. To learn effective representations for CTDGs, researchers have explored various techniques, including temporal message passing [15, 24, 27, 36, 40] and the incorporation of temporal intervals into random walks [13, 20, 35], etc.

The effectiveness of graph representation fundamentally hinges on the preservation of proximity between nodes. This implies that nodes that are proximate in the graph should also maintain closeness in the low-dimensional space. The first-order proximity, represented by the direct interactions between node pairs observed in the CTDG, can be easily maintained by making the embeddings of adjacent nodes close. However, considering the sparsity and the evolving nature of CTDGs, the preservation of first-order proximity alone does not provide a comprehensive measure of proximity. The high-order proximity [26] in CTDGs, evaluated by the closeness degree between target nodes and auxiliary nodes, embodies the **spatial-temporal duality**. Figure 1 illustrates this spatial-temporal duality of high-order proximity in CTDGs using a social network example. Suppose the model is predicting the interaction between users u and v at time $t = 11$. As there is not direct interaction between the target nodes, their shared neighbors should be taken into consideration. Unlike the static graphs, although users u and v have three shared neighbors in both subgraphs (a) and (b), they have more frequent and recent interactions with their shared neighbors in subgraph (a) than (b), indicating higher probability of connections between node pairs u and v. Although important, most existing methods [24, 27, 40] independently aggregate neighbor information of target nodes without considering dependencies between target nodes. Some existing works [13, 35] encode the appearance positions of auxiliary nodes on the random walks path, which fail to address the spatial-temporal duality property inherent in high-order proximity.

Despite its paramount importance, estimating and preserving comprehensive spatial-temporal proximity in CTDGs poses several challenges for the following reasons: *Firstly*, CTDGs entail multiple interactions occurring between node pairs at varying timestamps. The **interaction intensity**, which represents the count of interactions within a specified time interval, is instrumental in characterizing the degree of connection between node pairs. Consequently, measuring higher-order proximity in CTDGs necessitates a personalized intensity estimation between the target node pair and the auxiliary nodes at any given timestamp, a process that is computationally demanding. Several existing works [37, 50] employ the Hawkes process and learn node representations to estimate pairwise intensity. Nonetheless, integrating these methods to estimate high-order proximity necessitates pre-training of the node embeddings or neural networks, thereby incurring significant computational costs. *Secondly*, the dynamic nature of CTDGs leads to varying proximity between node pairs across different timestamps, influenced by both spatial and temporal factors. This necessitates efficient adaptation in proximity estimation, thereby introducing unique challenges in the modeling of CTDGs.

To address the aforementioned challenges, in this paper, we propose **CorDGT**, a **D**ynamic **G**raph **T**ransformer with **Cor**related Spatial-Temporal Positional Encoding. Inspired by the effectiveness of Poisson process in modeling counting process, we propose a novel Temporal Distance that incorporates a parameter-free interaction intensity estimation by leveraging the weak assumption of the Poisson Point Process. This approach circumvents the computational cost of pretraining, thereby enhancing efficiency. Based on Temporal Distance, we further propose a *Correlated Spatial-Temporal Positional Encoding (STPE-C)* that models the spatial-temporal duality of evolving high-order proximity in CTDGs. Equipped with the STPE-C, we propose a dynamic graph Transformer that adaptively preserves the comprehensive proximity for effective learning of node embeddings in CTDGs, which are subsequently leveraged for downstream tasks. Consistent performance improvement over eight baselines are observed on both link prediction and node classification tasks, demonstrating the superiority of the proposed CorDGT. Additionally, experiments on large-scale datasets demonstrate the superior effectiveness-efficiency trade-off of proposed CorDGT. In summary, the main contributions of this paper are as follows:

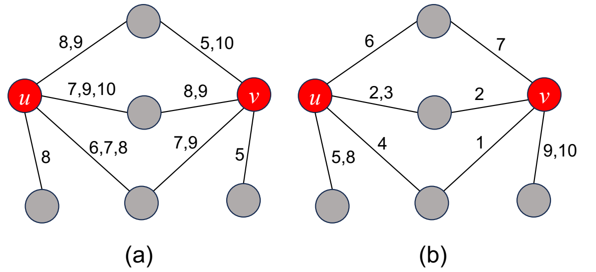


Figure 1: A social network example. The model is expected to predict the existence of the interaction between node u and v at $t = 11$.

- We propose a novel estimation of comprehensive proximity that incorporates an efficient parameter-free personalized intensity to encode the evolving spatial-temporal high-order proximity on CTDGs.
- We propose Dynamic Graph Transformer with Correlated Spatial-Temporal Positional Encoding (CorDGT), which efficiently preserves the evolving comprehensive proximity for effective node representation learning in CTDGs.
- Extensive experiments conducted on seven small and two large-scale datasets demonstrate the superiority and the scalability of the proposed CorDGT.

2 RELATED WORKS

Dynamic Graph Neural Networks. Existing dynamic graphs can be categorized into Discrete-Time Dynamic Graphs (DTDGs) and Continuous-Time Dynamic Graphs (CTDGs) based on whether the timestamps in the dynamic graphs are discrete or continuous. Early works on DTDGs learn the graph representation of each snapshot, which is then fed into a sequential model such as a Recurrent Neural Network or Transformer to learn temporal representation [17, 21, 25, 44, 45]. Some recent works learn the representation of successive snapshots selected by Bernoulli sampling [48] or a sliding window [49]. In contrast, the graph proximity of CTDGs is highly coupled with the time series. Most CTDG models leverage a unified model to learn the node representation via temporal graph neural networks [5, 15, 19, 24, 27, 40], random walks [13, 20, 35], or Transformers [4, 9, 34, 47]. However, due to the sparsity and evolving nature of CTDGs, addressing the spatial-temporal property of high-order proximity is important for proximity measurement, which is overlooked by existing works. Different from existing works, the proposed model addresses the spatial-temporal high-order proximity in CTDGs by incorporating the spatial-temporal distance between the target and the auxiliary nodes.

Temporal Point Process. The Temporal Point Process (TPP) is a mathematical model used to represent a sequence of events in time, which has been employed to model the interaction intensity of Continuous-Time Dynamic Graphs (CTDGs) [2, 27, 37]. These works have utilized a parametric network or temporal node embeddings to model the interaction intensity of TPPs, such as the Hawkes process [11]. However, the pre-training process of these methods can be time-consuming, making them unsuitable for high-order proximity modeling. Instead, we propose a parameter-free intensity estimation method based on the Poisson Point Process.

Graph Transformers. Several studies have explored the application of the pure Transformer model in static graph representation learning, wherein graph-specific information is incorporated as a soft inductive bias via positional encodings such as eigenvectors of the graph Laplacian matrix [7, 14], diagonals of the random walk matrix [8]. pairwise shortest path length [22, 43]. In the context of DTDGs, several works have proposed utilizing the Transformer model to capture the temporal evolution following spatial graph convolution within each graph snapshot [25, 33, 42, 46]. Given the Transformer model's capability to learn long-term dependencies, it has been adopted for learning on CTDGs [9, 47]. APAN [34] employs the Transformer to model asynchronous mail messages from other temporal neighbors. Our proposed model differs from these methods in terms of input tokens, positional encodings, and Transformer architecture.

3 PROBLEM DEFINITION

Problem Formulation. The Continuous-Time Dynamic Graph is defined as a set of interaction events $\mathcal{E} = [(u_i, v_i, t_i, e_i)]_{i=1}^{M}$, where M is the number of events, and the tuple (u_i, v_i, t_i) represents that nodes u_i and v_i interact at time t_i, and $e_i \in \mathbb{R}^{d_e}$ is the feature vector associated with the i-th interaction event. The node feature matrix of the CTDG is denoted as $\mathcal{X} = [x(1), ..., x(N)] \in \mathbb{R}^{N\times d_n}$ where N is the total number of nodes and $x(i)$ is the raw node feature of the i-th node. Two nodes may have multiple interactions at different timestamps in a CTDG. The objective of representation learning on CTDG is to learn a embedding function for each node i at time t: $f_i(t) \in \mathbb{R}^d$. Since most of the CTDG datasets do not have node labels, the CTDG models are typically trained based on the future link prediction task. Future link prediction on CTDGs aims to predict the occurrence probability of the link (u, v, t_{pred}) based on all the historical interactions happening before t_{pred}, which can be categorized into transductive and inductive settings based on whether the testing nodes are visible in the training stage. The network parameters trained based on transductive link prediction can be utilized for other downstream tasks such as node classification.

Temporal Neighbor [24, 40]. Given a node u at time t, the collection of its (1-hop) temporal neighbors is defined as the set of nodes that have interaction with u before t: $\eta^{(1)}(u;t) = \{(w, t')| (u, w, t', \cdot) \in \mathcal{E}, t' < t\}$. The collection of K-hop ($K > 1$) temporal neighbors of the node u at time t can be recursively defined as all the temporal neighbor of its $(K-1)$-hop temporal neighbors, denoted as $\eta^{(K)}(u;t)$. The K-hop temporal neighborhood of a node u at time t is defined as $\mathcal{N}^{(K)}(u;t) = \cup_{i=1}^{K}\eta^{(i)}(u;t)$.

4 MODEL

4.1 Overall Framework

In this section, we introduce the proposed **D**ynamic **G**raph **T**ransformer with **Cor**related Spatial-Temporal Positional Encoding (**CorDGT**). The general framework of CorDGT is presented in Figure 2. Suppose we are predicting the interaction probability of the target nodes u and v at time t_{pred}, CorDGT begins by sampling their *contextual nodes set* from their K-hop temporal neighborhood, denoted as $C(u, t_{pred})$ and $C(v, t_{pred})$, respectively. As stated in Section 1, the spatial-temporal high-order proximity between the target nodes u and v is characterized by their first-order proximity to these contextual nodes $w \in C(u, t_{pred}) \cup C(v, t_{pred})$. Therefore, we first introduce Spatial Distance and Temporal Distance to characterize the first-order proximity on the CTDG. Then, the Correlated Spatial-Temporal Positional Encodings (STPE-C) for each contextual node $w \in C(u, t_{pred}) \cup C(v, t_{pred})$ with respect to the target node pairs u and v at time t_{pred} is proposed to encode the spatial-temporal high-order proximity. Further, the network architecture of CorDGT is presented, which modifies the original Transformer to incorporate structural information of the CTDG.

4.2 Contextual Nodes Sampling

A simple tree-based sampling strategy is adopted to obtain contextual nodes set. Specifically, given the node u at time t_{pred}, its contextual nodes set is initialized as $C(u, t_{pred}) = \{(u, t_{pred})\}$. At the first iteration, we uniformly sample n_1 temporal neighbors of (u, t_{pred}) and add them to $C(u, t_{pred})$. At the k-th iteration ($k > 1$), we uniformly sample n_k temporal neighbors for each nodes sampled at the $(k-1)$ iteration and add them to $C(u, t_{pred})$. Note that the sampling numbers n_i ($i = 1, ..., K$) are predefined hyper-parameters.

4.3 Correlated Spatial-Temporal Positional Encoding

4.3.1 *Spatial Distance.* The spatial distance can be characterized by the shortest path length on the topological structure of the CTDG. However, since the topological structure of the CTDG is continuously evolving as new interaction events happen, online updating the shortest path length between any two nodes on the entire CTDG is time consuming. Alternatively, we use the shortest path length on the sampled K-hop temporal neighborhood as a proxy. Specifically, to compute Spatial Distance (SD) of w to w_0 at time t_{pred}, denoted as $\mathrm{SD}(w; w_0, t_{pred})$, we firstly sample K-hop temporal neighborhood from the root node w_0, then $\mathrm{SD}(w; w_0, t_{pred})$ is

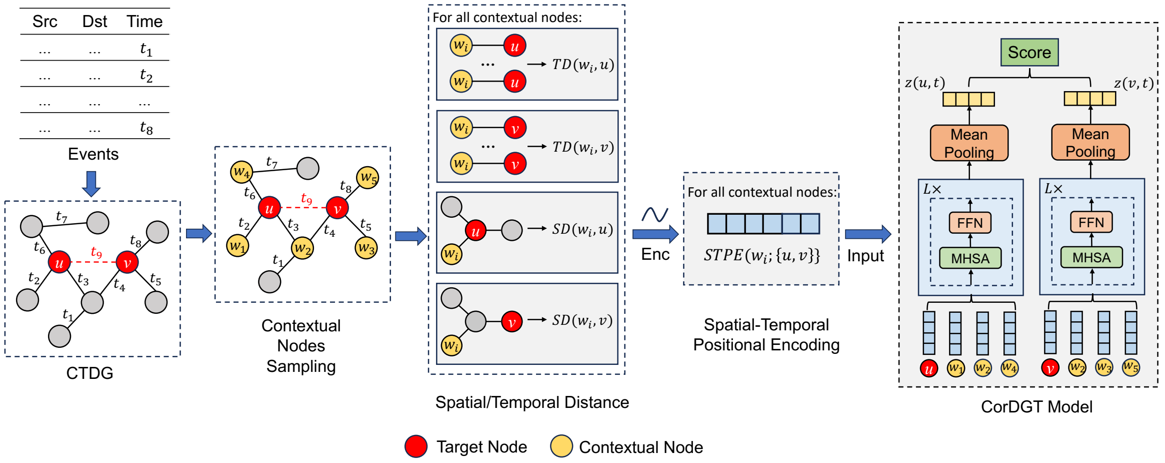


Figure 2: The framework of the proposed CorDGT.

defined as:

$$\mathrm{SD}(w; w_0, t_{pred}) = \min\{hop(w'; C(w_0, t_{pred})| \\ w = w', w' \in C(w_0, t_{pred})\} \tag{1}$$

where *hop* denotes the hop number of w' from the root node w_0. Note that if If $w = w_0$, $\mathrm{SD}(w; w_0, t_{pred})$ is set as 0. The computation of SD is based on the closest w to the root node w_0, since w may occur multiple times in the contextual node set of w_0. If w is not in $C(w_0, t_{pred})$, we set $\mathrm{SD}(w; w_0, t_{pred})$ as infinity.

4.3.2 Temporal Distance. In contrast to static graphs, the node pairs may have multiple interactions at different timestamps in the CTDG, thus the proximity between two nodes is associated with their interaction history. We propose Temporal Distance to characterize the proximity originated from the interaction history between two nodes. Suppose current timestamp is t_{pred}, and the timestamps sequence that nodes w and w_0 interacted prior to t_{pred} is denoted as $T(w, w_0, t_{pred}) = \{t_1, ..., t_n\}$ with $t_{i-1} < t_i (i = 2, ..., n)$ and $t_n < t_{pred}$. Then, the Temporal Distance (TD) between w and w_0 at time t_{pred}, denoted as $\mathrm{TD}(w, w_0, t_{pred})$ is defined as:

$$\mathrm{TD}(w, w_0, t_{pred}) = f(\{t_1, ..., t_n\}, t_{pred}) \tag{2}$$

where f is an arbitrary function which maps the interaction timestamps sequence to a scalar. Moreover, the temporal distance should satisfy following properties: (Recentness) If the most recent interaction between two nodes is closer to current time, then the temporal distance between them should be smaller; (Intensity) If the interaction intensity of these two nodes is higher, then the temporal distance between them should be smaller. The recentness property can be easily characterized by the difference between t_n and t_{pred}. However, estimating the interaction intensity between two nodes is not straightforward since there is no prior knowledge to its distribution and the interactions patterns of different node pairs may be highly divergent. Most existing works adopt Hawkes process in modeling the interaction intensity [37, 50]. However, these methods require to pre-train the node embeddings to obtain the intensity, thus is time-consuming. Instead, we propose a parameter-free approach to estimate the interaction intensity at any time. Specifically, we employ the Poisson point process assumption for its simplicity and generalization ability, which is a commonly used weak assumption for an unknown counting process [16, 35]. We provide following Theorem 1 with the Poisson point process assumption to evaluate maximum likelihood estimation of interaction intensity given the interaction sequence.

Lemma 1. *Suppose the interactions between w and w_0 prior to t_{pred}, denoted as $T(w, w_0, t_{pred}) = \{t_1, ..., t_n\}$ with $t_{i-1} < t_i(i = 2, ..., n)$ and $t_n < t_{pred}$, follow a Poisson point process with intensity λ, then the maximum likelihood estimation of λ is $\frac{n}{t_n}$.*

Proof. With the Poisson Point Process assumption, the probability of the interaction sequence $[t_1, ..., t_n]$ happens is:

$$Pr(N(t_{i-1}, t_i) = 1, i = 1, 2, ..., n) = \prod_{i=1}^{n} \frac{\lambda(t_i - t_{i-1})^1}{1!} e^{-\lambda(t_i - t_{i-1})} \tag{3}$$

where $N(t_{i-1}, t_i)$ denotes the number of interactions within the range (t_{i-1}, t_i) and $t_0 = 0$. Therefore, the likelihood function of the intensity λ can be written as:

$$L(\lambda) = \prod_{i=1}^{n} \lambda(t_i - t_{i-1}) \cdot e^{-\lambda(t_i - t_{i-1})} \tag{4}$$

Thus, the log-likelihood of λ is:

$$l(\lambda) = \sum_{i=1}^{n} \ln(\lambda) + \ln(t_i - t_{i-1}) - \lambda(t_i - t_{i-1}) \tag{5}$$

By setting the derivative $\frac{dl}{d\lambda} = 0$, we get the maximum likelihood estimation of λ:

$$\lambda_{MLE} = \frac{n}{t_n} \tag{6}$$

which concludes the proof. □

Lemma 1 enables us to characterize the interaction intensity by n/t_n. By integrating both the intensity property and recentness property, we specify the temporal distance defined in Eq. (2) as:

$$\text{TD}(w, w_0, t_{pred}) = \alpha * \frac{t_n}{t_{pred} \cdot n} + \beta * \frac{t_{pred} - t_n}{t_{pred}} \tag{7}$$

where $\alpha, \beta > 0$ are hyper-parameters. Note that if w and w_0 do not have interaction before, we set $\text{TD}(w, w_0, t_{pred})$ as a very large value. If $w = w_0$, $\text{TD}(w, w_0, t_{pred})$ is set as 0. For implementation, due to the sparsity of CTDGs, we only need to store the most recent interaction timestamp t_n and the interaction times count n for the node pairs that have interactions before, which leads to the memory complexity significantly less than $O(|\mathcal{E}|)$. To prevent the problem of information leakage, when computing the temporal distance in Eq. (7), we use the recorded t_n and n until the *previous* mini-batch of interactions. After the complete forward propagation of the current mini-batch, we update t_n and n records using the interactions of the current mini-batch.

4.3.3 Spatial-Temporal Positional Encoding. The aforementioned spatial distance and temporal distance encode the direct proximity between two nodes on CTDG in scalars. However, the expressiveness of self-attention will be restricted if we directly use the scalar distance as inputs. In addition, this encoding function should learn the difference of spatial-temporal distance among contextual nodes more effectively. Inspired by [28], we use the sinusoidal function $\text{Enc} : \mathbb{R}^+ \to \mathbb{R}^{2d}$ as the encoding function:

$$\begin{aligned} \text{Enc}(x)[2i] &= \sin\left(\frac{\epsilon x}{10000^{2i/d}}\right) \\ \text{Enc}(x)[2i+1] &= \cos\left(\frac{\epsilon x}{10000^{2i/d}}\right) \end{aligned} \tag{8}$$

where ϵ is used to amplify the influence of x on different positions of the encoding, and we set $\epsilon = 10000$ in this work. Further, the Unitary Spatial-Temporal Positional Encoding (STPE-U) of the contextual node w with respect to the single target node w_0 at time t_{pred} is defined as:

$$\begin{aligned} \text{STPE\_u}(w; w_0, t_{pred}) = &\text{MLP}(\text{Enc}(\text{TD}(w, w_0, t_{pred})))|| \\ &\text{MLP}(\text{Enc}(\text{SD}(w; w_0, t_{pred}))) \end{aligned} \tag{9}$$

where || denotes the concatenation operation, and MLP denotes Multi-Layer Perceptions. The above defined STPE-U can characterize the spatial-temporal first-order proximity of the contextual node to the single target *node.* To address the spatial-temporal high-order proximity between the target node pair (u, v), the contextual nodes $w \in C(u, t_{pred}) \cup C(v, t_{pred})$ are leveraged as the auxiliary nodes for the target node pair (u, v). Specifically, we propose the Correlated Spatial-Temporal Positional Encoding (STPE-C) of the contextual node w as the combination of its STPE-U to both target nodes:

$$\begin{aligned} \text{STPE\_c}(w; (u, v), t_{pred}) = &\text{STPE\_u}(w; u, t_{pred}) + \\ &\text{STPE\_u}(w; v, t_{pred}) \end{aligned} \tag{10}$$

4.4 Network Architecture

In this section, we present the network architecture of CorDGT, which is a Transformer based model incorporating the structural information of CTDG. The input of CorDGT is the node embeddings of contextual nodes $X = [x(w_1), ..., x(w_C)] \in \mathbb{R}^{C \times d_{emb}}$ where $C = |C(w_0, t_{pred})| (w_0 \in \{u, v\})$ is the size of contextual nodes set. The node embedding $x(w_i)$ is the concatenation of raw node feature of w_i and its STPE-C with respect to the target link (u, v) at time t_{pred}. In addition, some CTDG datasets may provide features associated with interaction events, which contains important semantic information about the correlations between the contextual node pairs. Therefore, we modify the self-attention module of the original Transformer to incorporate the event feature. Specifically, the event feature matrix is defined as $E = [e_{ij}]_{1 \le i,j \le C}$, where e_{ij} is the event feature if node i and j are interacted during contextual node sampling period otherwise a zero vector. The input node embedding X is denoted as $H^{(0)}$. Then, the self-attention module of CorDGT (CorDGTAttn) is defined as:

$$\begin{aligned} H_i^{(l)} &= \text{CorDGTAttn}(H^{(l-1)}, E) \\ &= \sum_{j=1}^{L} M_{ij} * \text{Softmax}(A_{ij}^{(l)} (H_j^{(l-1)} W_V^{(l)} + e_{ij} W_{EV}^{(l)}) \\ \text{Where} \quad A_{ij}^{(l)} &= \frac{h_i^{(l-1)} W_Q^{(l)} (h_j^{(l-1)} W_K^{(l)} + e_{ij} W_{EK}^{(l)})^T}{\sqrt{d_k}} \end{aligned} \tag{11}$$

where $H_i^{(l)}$ denotes the i-th row of the matrix $H^{(l)}$. $W_Q^{(l)}, W_K^{(l)} \in \mathbb{R}^{d \times d_K}$, $W_V^{(l)} \in \mathbb{R}^{d \times d_V}$, $W_{EK} \in \mathbb{R}^{d_e \times d_K}$, $W_{EV}^{(l)} \in \mathbb{R}^{d_e \times d_V}$ are weight matrices. For simplicity, we set $d_K = d_V = d$ for the intermediate layers. In addition, $M \in \mathbb{R}^{C \times C}$ in Eq. (11) is a masking matrix defined as follow:

$$M_{ij} = \begin{cases} 1, & t_i < t_j \quad \text{and} \quad hop_i \ge hop_j \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where t and hop are the timestamps and hop numbers obtained in contextual node sampling period. This masking matrix ensures that messages can only pass from the history to future, and from farther temporal neighbors to the closer temporal neighbors.

Following the common practice of Transformer models, we adopt the the Layer Normalization (LN) [1] and residual connection [12] in our CorDGT layer. For easier optimization, we adopt a Pre-Norm architecture [39] where the Layer Normalization is applied before CorDGTAttn and Feed-Forwad Networks (FFN). Formally, the CorDGT layer is defined as follows:

$$\begin{aligned} H'^{(l)} &= \text{CorDGTAttn}(\text{LN}(H^{(l-1)}), E) + H^{(l-1)} \\ H^{(l)} &= \text{FFN}(\text{LN}(H'^{(l)})) + H'^{(l)} \end{aligned} \tag{13}$$

where L is the total number of layers. Multi-head self attention [28] can also be adopted to further enhance the expressive power of CorDGT. The output of CorDGT layer $H^{(L)} \in \mathbb{R}^{C \times d}$ are the embeddings of contextual nodes. The embedding of the root node $w_0 (w_0 \in \{u, v\})$ at time t_{pred}, denoted as $z(w_0, t_{pred})$, is obtained by applying mean pooling on the node embeddings of its associated contextual nodes $C(w_0, t_{pred})$:

$$z(w_0, t_{pred}) = \frac{1}{C} \sum_{i=1}^{C} H^{(L)}(i, :) \tag{14}$$

4.5 Training Objective

Given the target link (u, v, t_{pred}), the node embeddings $z(u, t_{pred})$ and $z(v, t_{pred})$ can be computed using Eq. (14). Then, the predicted score of (u, v, t_{pred}) is computed as:

$$S(u, v, t_{pred}) = \sigma(\text{MLP}(z(u, t_{pred}), z(v, t_{pred}))) \tag{15}$$

where σ denotes the Sigmoid function.Finally, the Binary Cross Entropy (BCE) loss is adopted to train CorDGT:

$$\begin{aligned}\mathcal{L} = \sum_{(u,v,t)} & - \log S(u, v, t_{pred}) \\ & - \mathbb{E}_{r \sim Unif(\mathcal{V} \backslash \{u,v\})} \log(1 - S(u, r, t_{pred}))\end{aligned} \tag{16}$$

where $Unif(\cdot)$ denotes a uniform sampling distribution on the node set. The overall training pipeline is in Appendix.

4.6 Complexity Analysis

In this section, we analyze the time and spatial complexity of the proposed CorDGT model. For time complexity, given a mini-batch of interactions of size B, sampling the contextual nodes requires to binary-search the insertion point of the timestamp and costs $O(B \log(\bar{d}))$, where $\bar{d}$ is the average degree of nodes. Computing the Temporal Distance costs $O(BC)$ complexity, where C is the number of contextual nodes. Computing the Spatial Distance costs $O(K)$ complexity where K is the maximum hop of contextual nodes. Forwarding the model costs $O(C^2HD)$ due to self-attention operation, where H is the number of attention heads and D is the hidden dimension of weights. Therefore, the time complexity of training CorDGT is $O(B(\log(\bar{d}) + C) + C^2HD)$. For spatial complexity, storing the statistics of interactions cost $O(|\mathcal{E}|)$, where $|\mathcal{E}|$ is the total number of interactions. This spatial complexity is inevitable for learning CTDG models due to the CTDG data loading. The complexity comparison with other methods is presented in Appendix.

5 EXPERIMENTS

5.1 Experimental setup

Datasets. We evaluate the proposed model on nine Continuous-Time Dynamic Graph (CTDG) datasets: Reddit, Wikipedia, LastFM, UCI, Enron, Social Evolution, Flights, Gowalla-Food, and Gowalla-Outdoors. Among these, Reddit and Wikipedia constitute bipartite networks abundant in node/edge attributes, while LastFM represents a bipartite network devoid of node features. UCI, Enron, Social Evolution, and Flights are non-bipartite communication networks, also lacking attributes. Gowalla-Food and Gowalla-Outdoors are two large-scale datasets derived from the primary Gowalla dataset. For datasets without meaningful node features (i.e., LastFM, UCI, and Enron), we employ zero vectors as node features. Further details regarding these datasets are provided in Appendix.

Baselines. We compare the proposed CorDGT with three types of CTDG models: (1) GNN-based: DyRep [27], TGAT [40], TGN [24] and Graphmixer [5]. (2) Random walk based: CTDNE [20] and CAW [35]. (3) Transformer based: TCL [32] and TGSRec [9]. More introductions about the baseline methods and tuned hyper-parameters are presented in Appendix.

Evaluation Protocols. Our evaluation protocols closely follow [24]. In specific, we adopt transductive/inductive link prediction and dynamic node classification tasks for evaluation. For *Transductive* link prediction task, we split the total time range $[0, T]$ into three seperate intervals $[0, T_{train})$, $[T_{train}, T_{val}]$ and $[T_{val}, T)$ with $T_{train}/T = 0.7$ and $T_{val}/T = 0.85$ fixed. Then, we allocate the interactions happening within each interval to generate the training, validation and testing set. The inductive link prediction task follows the same splitting protocol as the transductive experiments. However, we randomly select 10% of the nodes as "masking nodes", excluding any links associated with them in the training set, and removing any links not associated with them in the validation and testing sets.

Training Configurations. We train all the models for 50 epochs and adopt the early stopping strategies. We adopt Adam optimizer and learning rate of 0.001 for all the tasks. Early stopping strategy is adopted. The batch size is set as 100. We sample 2-hop temporal neighbors for all datasets. More details about hyper-parameter of CorDGT and other baselines are presented in Appendix.

5.2 Results and Discussion

The Average Precision (AP) scores of transductive and inductive link prediction experiments are presented in Table 1. The Area Under the receiver operating Characteristic (AUC) results are presented in Appendix. As can be seen from Table 1, our CorDGT achieves the best AP performance on all the datasets for both transductive and inductive settings. Specifically, the transductive AP and inductive AP of CorDGT show an average improvement of 1.95% and 3.21%, respectively, demonstrating the effectiveness of CorDGT.

In addition, we make the following observations: (1) Our proposed model demonstrates robust performance across both attributed networks (Reddit and Wikipedia) and non-attributed datasets (LastFM, UCI, MOOC, and UCI). In contrast, the performance of several baseline methods, which lack node encodings designed for the evolving laws of CTDGs (such as Jodie, DyRep, and TGAT), significantly decreases on non-attributed datasets. This suggests the efficacy of our proposed STPE-C in modeling the evolving nature of CTDGs. (2) On the LastFM and Social Evolution datasets, our model improves the inductive Average Precision (AP) by 6.71% and 9.52% over the strongest baseline, respectively. This may be attributed to the significantly higher average interaction intensity of the Social Evolution ($\lambda = 2.73 \times 10^{-3}$) and LastFM ($\lambda = 5.04 \times 10^{-4}$) datasets compared to others. As such, the intensity may play a more crucial role in proximity estimation. Our proposed model excels at capturing long-term interaction intensity via temporal distance. (3) When compared to the parametric intensity-based method (DyRep), our proposed model displays significantly improved performance. This improvement can be attributed to the incorporation of parametric-free intensity into high-order proximity encoding. The results of node classification is presented in Appendix.

5.3 Ablation Studies

In this subsection, we conduct ablation studies to evaluate the effectiveness of different modules of CorDGT . The inductive AP and AUC results are shown in Table 2. The results are analyzed as follows: (1) In Ablations 1 and 2, we eliminate the recentness term and intensity term from the temporal distance calculation, respectively. As evidenced by the results, the removal of either term leads to a performance degradation across all datasets, especially on LastFM. This demonstrate the significance of both recentness and intensity in computing temporal distance. (2) In Ablations 3

Table 1: The Average Precision (AP) results of transductive/inductive link prediction are reported. The values are multiplied by 100. The results of the best and second best performing models are highlighted in bold and underlined, respectively.

	Model	Reddit	Wikipedia	LastFM	UCI	Enron	Social Evo.	Flights
Transductive	JODIE	97.18±0.2	94.09±0.5	70.89±0.8	85.91±0.9	74.73±4.6	88.24±0.5	95.23±1.6
	DyRep	98.09±0.1	94.54±0.3	68.45±3.2	52.94±0.8	69.36±3.4	87.93±0.3	94.97±0.6
	TGAT	98.15±0.1	94.69±0.1	54.42±0.9	77.75±0.2	58.91±0.3	92.15±0.2	94.76±0.2
	TGN	98.70±0.0	98.49±0.1	72.43±2.9	85.57±4.5	77.45±3.3	93.27±0.1	97.23±0.1
	CAWN	98.72±0.0	98.79±0.1	85.52±0.3	92.51±0.1	89.32±0.1	86.39±0.1	98.12±0.3
	TGSRec	88.56±2.4	85.68±1.0	67.60±3.9	76.00±0.5	69.46±2.8	74.51±0.7	94.18±0.3
	TCL	97.68±0.2	96.82±0.3	70.88±1.5	89.43±1.0	80.84±0.4	93.25±0.1	91.38±0.3
	Graphmixer	97.42±0.0	97.25±0.1	78.26±0.4	93.46±0.3	83.28±0.2	92.93±0.1	91.37±0.6
	CorDGT	**99.18±0.0**	**99.07±0.1**	**92.23±0.1**	**96.03±0.2**	**91.76±0.5**	**93.81±0.2**	**98.81±0.0**
	Improve	0.46	0.28	6.71	2.57	2.44	0.54	0.69
Inductive	JODIE	94.46±0.1	92.98±0.1	82.88±1.3	72.93±0.7	72.97±2.0	91.49±0.5	95.17±0.4
	DyRep	95.81±0.3	92.01±0.4	80.43±2.2	49.42±0.8	57.54±3.3	89.82±0.5	93.58±0.7
	TGAT	97.18±0.6	93.50±0.1	55.63±1.7	70.96±0.7	58.44±2.4	90.74±0.4	89.46±0.3
	TGN	97.54±0.1	97.83±0.1	79.16±2.7	82.99±1.2	72.08±2.8	90.46±0.6	95.63±0.3
	CAWN	97.19±0.6	97.33±0.4	83.51±0.6	91.95±0.6	86.28±2.2	79.14±0.2	96.80±0.2
	TGSRec	82.08±5.4	78.39±0.3	68.42±3.8	65.46±0.7	68.83±3.8	68.85±1.6	87.64±0.3
	TCL	93.89±0.5	97.08±0.2	72.56±1.7	88.37±1.8	77.28±0.8	91.67±0.1	83.41±0.1
	Graphmixer	97.38±0.0	97.10±0.0	83.11±0.4	92.10±0.6	75.84±1.2	92.03±0.1	83.03±0.1
	CorDGT	**98.82±0.1**	**98.48±0.2**	**93.03±1.8**	**94.70±1.0**	**91.65±1.3**	**94.44±0.8**	**97.46±0.4**
	Improve	1.28	0.65	9.52	2.60	5.37	2.41	0.66

Table 2: Ablation studies results. Results of inductive link prediction are reported. The values are multiplied by 100. The best performance is marked in bold.

Ablations	Reddit		Wikipedia		LastFM		UCI	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Full Model	**98.67±0.1**	**98.82±0.1**	**98.26±0.4**	**98.48±0.2**	**93.01±2.0**	**93.03±1.8**	**93.20±1.8**	**94.70±1.0**
1. Set $\alpha = 0$ in Eq. (7)	98.09±0.0	98.24±0.1	98.17±0.0	98.28±0.3	89.10±0.6	92.07±0.4	91.30±0.3	91.52±0.3
2. Set $\beta = 0$ in Eq. (7)	98.10±0.0	98.19±0.1	97.87±0.0	98.18±0.2	89.55±0.6	90.80±1.7	90.46±0.3	94.28±0.8
3. Remove temporal distance	97.91±0.1	98.33±0.3	97.81±0.1	97.11±0.1	88.43±0.1	86.68±0.4	92.19±0.1	91.20±0.3
4. Remove spatial distance	97.81±0.0	98.02±0.1	97.62±0.0	97.36±0.4	92.47±0.1	91.43±0.5	90.15±0.1	92.37±1.0
5. Replace STPE-C with STPE-U	96.99±0.2	93.87±1.1	89.22±2.2	75.38±5.1	64.82±0.7	50.03±2.9	78.59±2.2	64.32±4.6
6. Remove mask in Eq. (11)	98.14±0.0	98.19±0.0	98.08±0.1	98.19±0.1	88.95±0.1	85.81±1.4	90.31±0.1	92.05±0.3
7. Use recent sampling	97.49±0.1	97.36±0.1	96.44±0.3	96.93±0.2	87.23±0.8	85.91±0.1	86.25±1.6	87.15±1.5

and 4, we exclude the spatial distance and temporal distance from the STPE-C component, respectively. The results indicate that the performance across all datasets, particularly non-attributed ones, is compromised when either spatial distance or temporal distance is removed. This highlights the importance of modeling both spatial and temporal proximity in learning on CTDGs. (3) In Ablation 5, we substitute the binary STPE-C with STPE-U as defined in Eq. 8. In this scenario, the high-order proximity between the target nodes fails to be captured. We observe a significant drop in performance across all datasets, underscoring the critical role of modeling spatial-temporal high-order proximity in learning the evolving patterns of CTDGs. (4) In Ablation 6, we remove the masking matrix utilized in the Transformer model, which also leads to a performance decline across all datasets. (5) In Ablation 7, we replace the uniform contextual nodes sampling with the most recent sampling (i.e., the most recent interacted neighbors are sampled as the contextual nodes). The performance of CorDGT drops on all four datasets. The reason may be that the diversity of neighbors decrease when replaced with most recent sampling strategy, thus the high-order proximity estimation may be less accurate.

5.4 Scalability Analysis

In this section, we evaluate the performance and efficiency of the proposed CorDGT on large-scale datasets. We adopt *Outdoors* and *Food* subsets from the large Gowalla [18] dataset for evaluation. The Outdoors dataset contains around 0.22M nodes and 1.19M edges. The Food dataset has around 0.67M nodes and 2.71M edges. We run all the models for one epoch and compare the performance. The inductive AP metrics and the training speed per epoch of the Food dataset are presented in Figure 3. The results of Outdoors are presented in Appendix.

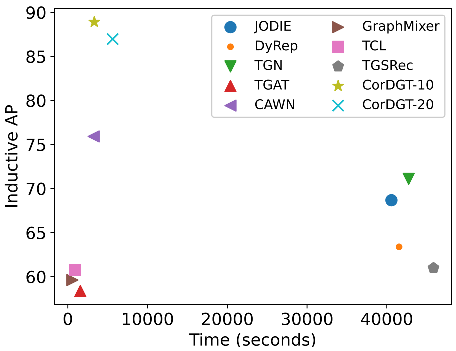


Figure 3: The inductive AP metrics and the training time per epoch on Food datasets. The closer to the upper left corner, the better performance. CorDGT-10 and CorDGT-20 denote the CorDGT with 10 and 20 contextual nodes, respectively.

As can be seen from Figure 3, both two configurations of CorDGT can achieve consistently outperform the baselines in terms of inductive AP. In addition, training one epoch of CorDGTwith 10 contextual neighbors takes 3319 seconds, which is significantly faster than memory-based models (JODIE, DyRep and TGN). It is because the memory-based models need to store the memory state for each node. Given the vast number of nodes in large-scale datasets, the process of storing and managing the memory state incurs a considerable computational cost. Our proposed CorDGT is marginally slower than CAWN (3234 seconds), but has significant better inductive AP performance.

5.5 Visualization

One motivation of the proposed CorDGT is to capture the spatial-temporal high-order proximity in CTDGs by considering the distance of the contextual nodes to both ends of target nodes. To further demonstrate the interpretability of the proposed CorDGT, we visualize the link prediction score by the contextual nodes with different Temporal Distance (TD) and Spatial Distance (SD) to the target nodes. Specifically, we replace Eq. (15) as:

$$S(u, v, t_{pred}) = \sigma(\Phi^T(z(u, t_{pred}) + z(v, t_{pred}))) \quad (17)$$

where $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^1$ is a trainable linear projector, and train the model. In this way, the contribution of each contextual node w to the link prediction score can be decomposed as $\Phi^T(h_w)$. After training, we randomly select a mini-batch of contextual nodes. To see the influence of temporal distance to the predicted score, we evenly split the range of TD to target node pairs in this mini-batch to 5 groups, which formulates total 25 buckets. Then, we allocate the contextual nodes into these buckets based on their TD to target node pairs. Similarly, we can allocate the contextual nodes into different buckets based on their SD to target node pairs. Finally, we compute the average prediction score of the contextual nodes in each bucket. The visualization results on UCI and Enron are shown in Figure 4. As can be seen from Figure 4, the contextual nodes have closer temporal distance (closer to top-left corner) to both ends of the target link will give higher prediction score of the link (closer to yellow). This indicates that the proposed CorDGT will give higher prediction if the contextual nodes have smaller temporal distance to target node pairs. Similar observations are also seen from spatial distance. Therefore, the proposed CorDGT may be capable to capture the spatial-temporal high-order proximity.

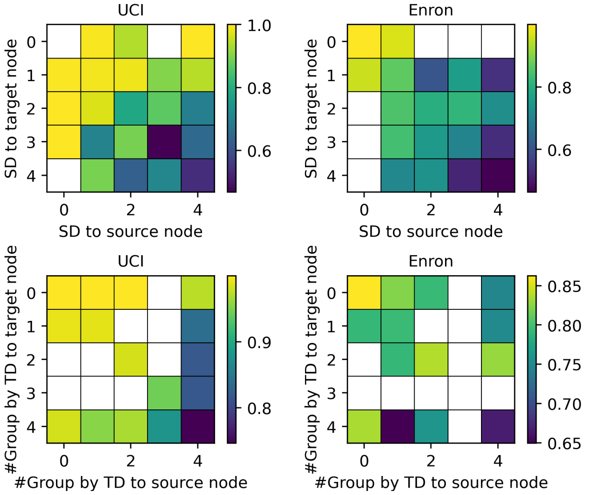


Figure 4: Heatmap values indicate the confidence score on positive source/target node pairs predicted by different groups of contextual nodes. Left to right: UCI and Enron datasets. Top to bottom: the contextual nodes are grouped according to their Spatial Distance and Temporal Distance to source/target nodes. The blank cells indicate that no data is allocated to this group. Best viewed in color.

6 CONCLUSIONS

This paper introduces the Dynamic Graph Transformer with Correlated Spatial-Temporal Positional Encoding (CorDGT), a novel approach for representation learning on Continuous-Time Dynamic Graphs (CTDGs). We employ the Poisson Point Process assumption and sampled temporal neighborhood to achieve comprehensive proximity estimation on CTDGs. Subsequently, we propose Correlated Spatial-Temporal Positional Encodings (STPE-C), which utilizes the comprehensive proximity to capture spatial-temporal high-order proximity. Extensive experiments conducted on seven small and two large-scale datasets demonstrate the performance superiority and scalability of the proposed CorDGT model. A potential future direction for this work could involve designing more sophisticated spatial-temporal distances for improved proximity estimation and preservation.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (62476244), Zhejiang Provincial Natural Science Foundation of China (Grant No: LTGG23F030005), National Natural Science Foundation of China (62372399,62476245) and the advanced computing resources provided by the Supercomputing Center of Hangzhou City University.

ETHICAL CONSIDERATIONS

The proposed CorDGT is used to learn the temporal embeddings which can be leveraged for downstream tasks such as link prediction and node classification. The direct negative societal effects of this research, encompassing fairness, privacy, and security considerations, are minimal. Nevertheless, akin to other predictive models, a few erroneous predictions by the model could impact system functionality. Despite extensive experimental validation of the model's efficacy, occasional inaccurate predictions, particularly on outlier data, remain plausible. Therefore, enhancing data quality through measures such as data cleaning prior to model application is advised.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Xiaofu Chang, Xuqin Liu, Jianfeng Wen, Shuang Li, Yanming Fang, Le Song, and Yuan Qi. 2020. Continuous-time dynamic graph learning via neural interaction processes. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 145–154.

[3] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[4] Weilin Cong, Yanhong Wu, Yuandong Tian, Mengting Gu, Yinglong Xia, Mehrdad Mahdavi, and Chun-cheng Jason Chen. 2021. Dynamic graph representation learning via graph transformer networks. *CoRR abs/2111.10447* (2021).

[5] Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. 2023. Do We Really Need Complicated Model Architectures For Temporal Networks? *International Conference on Learning Representations* (2023).

[6] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. 2021. On the equivalence of decoupled graph convolution network and label propagation. In *Proceedings of the Web Conference 2021*. 3651–3662.

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

[8] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2021. Graph neural networks with learnable structural and positional representations. *International conference on learning representations* (2021).

[9] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 433–442.

[10] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Neural Information Processing Systems*.

[11] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Ming Jin, Yuan-Fang Li, and Shirui Pan. 2022. Neural Temporal Walks: Motif-Aware Representation Learning on Continuous-Time Dynamic Graphs. In *Thirty-Sixth Conference on Neural Information Processing Systems*.

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

[15] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 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.

[16] Stephen Lavenberg. 1983. *Computer performance modeling handbook*. Elsevier.

[17] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[18] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 739–748.

[19] Yuhong Luo and Pan Li. 2022. Neighborhood-aware scalable temporal network representation learning. In *Learning on Graphs Conference*. PMLR, 1–1.

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

[21] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5363–5370.

[22] Wonpyo Park, Woong-Gi Chang, Donggeon Lee, Juntae Kim, et al. 2022. Grpe: Relative positional encoding for graph transformer. In *ICLR2022 Machine Learning for Drug Discovery*.

[23] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

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

[25] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 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.

[26] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[27] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*.

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

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).

[30] Bohao Wang, Jiawei Chen, Changdong Li, Sheng Zhou, Qihao Shi, Yang Gao, Yan Feng, Chun Chen, and Can Wang. 2024. Distributionally Robust Graph-based Recommendation System. In *Proceedings of the ACM on Web Conference 2024*. 3777–3788.

[31] Can Wang, Zhe Wang, Defang Chen, Sheng Zhou, Yan Feng, and Chun Chen. 2024. Online adversarial knowledge distillation for graph neural networks. *Expert Systems with Applications* 237 (2024), 121671.

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

[33] Tian Wang, Jiahui Chen, Jinhu Lü, Kexin Liu, Aichun Zhu, Hichem Snoussi, and Baochang Zhang. 2022. Synchronous spatiotemporal graph transformer: A new framework for traffic data prediction. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[34] Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. 2021. APAN: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 International Conference on Management of Data*. 2628–2638.

[35] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. 2021. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. In *International Conference on Learning Representations*.

[36] Zhe Wang, Tianjian Zhao, Zhen Zhang, Jiawei Chen, Sheng Zhou, Yan Feng, Chun Chen, and Can Wang. 2024. Towards Dynamic Graph Neural Networks with Provably High-Order Expressive Power. *arXiv preprint arXiv:2410.01367* (2024).

[37] Zhihao Wen and Yuan Fang. 2022. Trend: Temporal event and node dynamics for graph representation learning. In *Proceedings of the ACM Web Conference 2022*. 1159–1169.

[38] Jiancan Wu, Xiangnan He, Xiang Wang, Qifan Wang, Weijian Chen, Jianxun Lian, and Xing Xie. 2022. Graph convolution machine for context-aware recommender system. *Frontiers of Computer Science* 16, 6 (2022), 166614.

[39] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*. PMLR, 10524–10533.

[40] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*.

[41] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.

[42] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908* (2020).

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

[44] Jiaxuan You, Tianyu Du, and Jure Leskovec. 2022. ROLAND: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2358–2366.

[45] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.

[46] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. 2020. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*. Springer, 507–523.

[47] Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems* 36 (2023), 67686–67700.

[48] Kaike Zhang, Qi Cao, Gaolin Fang, Bingbing Xu, Hongjian Zou, Huawei Shen, and Xueqi Cheng. 2023. DyTed: Disentangled Representation Learning for Discrete-time Dynamic Graph. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3309–3320.

[49] Yifan Zhu, Fangpeng Cong, Dan Zhang, Wenwen Gong, Qika Lin, Wenzheng Feng, Yuxiao Dong, and Jie Tang. 2023. WinGNN: Dynamic Graph Neural Networks with Random Gradient Aggregation Window. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3650–3662.

[50] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. 2018. Embedding temporal network via neighborhood formation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2857–2866.

A OVERALL ALGORITHM OF CORDGT

Algorithm 1 CorDGT Training

```
Input: Training data $\mathcal{E} = [(u_i, v_i, t_i, e_i)]_{i=1}^{M}$, node feature $\mathcal{X} = [x(i)]_{i=1}^{N}$, hop number $K$, sampling number of contextual nodes $\{n_i\}_{i=1}^{K}$, $\alpha$, $\beta$.
Initialize sparse matrices $\mathcal{M}_n$ and $\mathcal{M}_t$.
$loss \leftarrow 0$
for $(u, v, e, t) \in \mathcal{E}$ do
    $r \leftarrow$ Sampled negative node;
    Sample the contextual node sets $C(u,t), C(v,t), C(r,t)$.
    for $(w, t') \in C(u,t) \cup C(v,t)$ do
        Read the interaction counts $cnt(u,w)$ and $cnt(v,w)$ from $\mathcal{M}_n$; Read the latest interaction timestamps $t_n(u,w)$ and $t_n(v,w)$ from $\mathcal{M}_t$.
        Compute STPE_c($w, (u,v), t_{pred}$) using Eq. (10).
        $h^{(0)}(w) \leftarrow$ STPE_c($w, (u,v), t_{pred}$)$||x(w)$.
    end for
    $H_{pos}^{(0)}(u) \leftarrow$ stack($\{h^{(0)}(w))|w \in C(u,t)\}$), $H_{pos}^{(0)}(v) \leftarrow$ stack($\{h^{(0)}(w))|w \in C(v,t)\}$)
    Forward $H_{pos}^{(0)}(u)$ and $H_{pos}^{(0)}(v)$ using Eq. (13-14) to obtain $z_{pos}(u,t)$ and $z_{pos}(v,t)$.
    Forward $z_{pos}(u,t)$ and $z_{pos}(v,t)$ using Eq. (15) to obtain score $S(u,v,t)$.
    for $(w, t') \in C(u,t) \cup C(r,t)$ do
        Read the interaction counts $cnt(u,w)$ and $cnt(v,w)$ from $\mathcal{M}_n$; Read the latest interaction timestamps $t_n(u,w)$ and $t_n(v,w)$ from $\mathcal{M}_t$.
        Compute STPE_c($w, (u,r), t_{pred}$) using Eq. (10).
        $h^{(0)}(w) \leftarrow$ STPE_c($w, (u,r), t_{pred}$)$||x(w)$.
    end for
    $H_{neg}^{(0)}(u) \leftarrow$ stack($\{h^{(0)}(w))|w \in C(u,t)\}$), $H_{neg}^{(0)}(r) \leftarrow$ stack($\{h^{(0)}(w))|w \in C(r,t)\}$)
    Forward $H_{neg}^{(0)}(u)$ and $H_{neg}^{(0)}(r)$ using Eq. (13-14) to obtain $z_{neg}(u,t)$ and $z_{neg}(r,t)$.
    Forward $z_{neg}(u,t)$ and $z_{neg}(r,t)$ using Eq. (15) to obtain score $S(u,r,t)$.
    $loss \leftarrow loss + BCE(S(u,v,t), S(u,r,t))$
    Update $\mathcal{M}_n(u,v) \leftarrow \mathcal{M}_n(u,v) + 1$, $\mathcal{M}_t(u,v) \leftarrow t$
end for
return $loss$
```

B MORE DISCUSSION ON OTHER STOCHASTIC PROCESS

The intensity is the number of happening times within a certain time interval of a counting process. Therefore, non-counting processes, such as Gaussian process and Wiener process, are not applicable for intensity estimation. Poisson process is the most well-known and commonly used counting process. It assumes the intensity is a constant, which provides great mathematical tractability and computation efficiency. Most of other counting processes are generalized from Poisson processes, such as non-homogeneous Poisson process, Hawkes process and Markovian arrival process,

Table 3: The time and space complexity of different models.

	Space Complexity	Time Complexity
TGAT	$O(B\log(\bar{d})+CHD)$	$O(P)$
TGN	$O(B\log(\bar{d})+CHD)$	$O(P+ND')$
CorDGT	$O(B(\log(\bar{d})+C)+C^2HD)$	$O(P+\|\mathcal{E}\|)$

which includes more undetermined parameters for flexibility. Estimating the parameters of these process requires significantly higher computation budget than Poisson process.

C COMPLEXITY COMPARISON WITH EXISTING MODELS

In Section 4, we analyze the computation complexity of CorDGT . In this section, we also analyze the computation complexity of two baseline methods, i.e., TGAT [40] and TGN [24]. For these models, given a mini-batch of interactions of size B, the historical neighbor sampling process costs $O(B\log(\bar{d}))$, where $\bar{d}$ represents the average degree of nodes. The forward process, costs $O(CHD)$ where C, H, D denote the number of sampled neighbors, attention heads, and hidden dimensions, respectively. Thus, the overall time complexity of TGAT and TGN is $(Blog(\bar{d})+CHD)$. The size of network parameters is denoted as P. In addition, TGN requires a memory and costs the spatial complexity of $O(ND')$ where N and D' denote the number of nodes and hidden dimension of the memory. We summarize the time and space complexity in Table 3.

D EXPERIMENTAL SETTING

D.1 Datasets

Our experiments section includes seven public datasets: Reddit [1], Wikipedia[2], UCI[3], LastFM[4], Enron[5], Social Evolution[6] and Flights[7]. Reddit network is an user action datasets which consists of subreddits posted by different users in one month on Reddit website. It is a bipartite dataset consisting of 10000 most active users and 984 subreddits with rich interaction feature provided. Wikipedia network records the clicking actions on wikipedia pages by different users. It is a bipartite network consisted by clicking actions on 1000 pages in one month made by users with rich interaction feature provided. UCI network is non-bipartite network which contains sent messages between the users of an online community of students from the University of California, Irvine. The nodes represent students and the edges represent the communicated messages among them. Enron is a non-bipartite dataset which consists of approximately 0.5M emails that were exchanged between employees of the Enron energy company over a span of three years. Social Evolution is a mobile phone proximity network which tracks the everyday life of a whole undergraduate dormitory from October 2008 to May 2009. Flights is a directed dynamic flight network illustrating the development of the air traffic during the COVID-19 pandemic, which was extracted and cleaned for the purpose of this study. Each node represents an airport and each edge is a tracked flight. The edge weights specify the number of flights between two given airports in a day. In addition, we select the large-scale Gowalla [18] dataset for scalability evaluation. Gowalla is a social network for users check-ins at various locations, containing about 36 million check-ins made by 0.32 million users over 2.8 million locations. These check-in records are in the time span of Jan 2009 - June 2011. The locations are grouped into 7 main fields. We select a subset of Outdoors and Food field for experiments. Specifically, we choose the part of the Outdoors data from Jan. 2009 to Dec. 2010 and the part of the Food data from Jan. 2011 to June 2011. Detailed statistics of aforementioned datasets are presented in Table 4.

D.2 Baselines

The brief introduction of baseline methods in the Experiments section are as follows:

CTDNE [20]. CTDNE extends DeepWalk [23] to dynamic graphs which leverages a SkipGram model on the temporal random walk sequence and learn the node embeddings.

JODIE [15]. Jodie updates node embeddings in an interaction via two coupled RNNs, which are leveraged for future link prediction via a temporal projector.

DyRep [27]. DyRep updates the node embeddings involved in an interaction by a recurrent model considering the messages from 2-hop temporal neighbors.

TGAT [40]. TGAT extends GAT [29] and GraphSAGE [10] to dynamic graphs, which samples and recursively aggregates the messages of k-hop temporal neighbors. The temporal representation is obtained by Fourier transformation on the time interval.

TGN [24]. TGN proposes a generalized message-passing networks by extending Jodie and TGAT with a per-node memory mechanism for long-time interactions.

TGSRec [9]. TGSRec proposes a Temporal Collaborative Transformer which simultaneously captures the collaborative signals from users and items as well as temporal dynamics.

TCL [32]. proposes a dynamic-graph-topology-aware Transformer with a two-stream encoder for semantic inter-dependency modeling. Contrastive learning is adopted to maximize mutual information between future interaction nodes.

CAWN [35]. CAWN samples temporal random walks and anonymize the node identities via Causal Anonymous Walks (CAW). The node encodings on temporal random walks are learned via a sequential model.

Graphmixer [5]. Graphmixer proposes a conceptually simple architecture that leverage MLP and mean-pooling to aggregate the temporal information and node features of K most recently interacted neighbors. Note that the original Graphmixer leverages

[1] http://snap.stanford.edu/jodie/reddit.csv
[2] http://snap.stanford.edu/jodie/wikipedia.csv
[3] http://konect.cc/networks/opsahl-ucsocial/
[4] http://snap.stanford.edu/jodie/lastfm.csv
[5] https://www.cs.cmu.edu/~enron/
[6] http://realitycommons.media.mit.edu/socialevolution.html
[7] https://zenodo.org/records/3974209#.Yf62HepKguU

Table 4: Dataset statistics. Average Interaction Intensity $\lambda = 2|\mathcal{E}|/|\mathcal{V}|\mathcal{T}$ [35], where $|\mathcal{E}|$ and $|\mathcal{V}|$ denote the number of interactions and nodes, respectively. $\mathcal{T}$ denotes the total duration (seconds).

	Reddit	Wikipedia	LastFM	UCI	Enron	Social Evolution	Flights	Outdoors	Food
# Nodes	10,984	9,227	1,980	1,899	184	74	13,169	223,777	673,858
# Links	672,447	157,474	1,293,103	59,835	125,235	2,099,519	1,927,145	1,192,397	2,708,688
# Nodes attributes	172	172	0	0	0	0	0	0	0
# Link attributes	172	172	0	0	0	2	1	0	0
λ	4.57×10^{-5}	1.27×10^{-5}	5.04×10^{-4}	3.79×10^{-6}	1.20×10^{-5}	2.73×10^{-3}	1.41×10^{-5}	1.71×10^{-7}	2.58×10^{-7}
Is bipartite?	True	True	True	False	False	False	False	True	True

one-hot node encoding as input, thus can not be applied for inductive experiment. In this work, we replace the one-hot encodings as the node encodings used by other baselines for fair comparison.

D.3 Hyper-parameters Tuning

For all the baselines, we set the dimension of time encoding and hidden unit as 100 and 172, respectively. For Jodie, DyRep and TGN, we adopt the implementation [8] for evaluation. In specific, we set the memory dimension as 32 (for UCI) or 172 (for Reddit, Wikipedia and LastFM). We adopt a one-layer model with 10 temporal neighbors being sampled. The official implementations of TGAT [9] and TGSRec [10] are adopted. The number of layers, attention heads and the sampled temporal neighbors are set as 2, 2 and 20, respective.

For CAWN, we adopt its official implementation [11]. We grid search following hyper-parameters: the time scaling factor is set as $\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$, the random walk length in $\{2, 3, 4, 5\}$ and the number of walks in $\{16, 32, 64, 128\}$.

For Graphmixer, we adopt the original implementation [12]. The time gap is set as 2000. The number of MLP-Mixer layers is set as 2. For TCL, we adopt the implementation of DyGLib [13]. The number of layers, attention heads and the sampled temporal neighbors are set as 2, 2 and 20, respective.

D.4 Implementation details of CorDGT

For fair comparison, we train all the models for 50 epochs with early stopping executed if there is no improvement on validation AP for 3 epochs. In addition, we use the batch size of 100 for all models. We repeat the methods for 3 runs and report the mean and standard deviation of statistics. For the proposed CorDGT, we set the learning rate as 0.001 and optimizer as Adam for all datasets. The attention heads, layer number and hidden dimension of main CorDGT encoder is set as 6, 2, 64, respectively, for all datasets. We sample 2-hop temporal neighbors for all datasets. For LastFM and UCI, the sampling numbers of contextual neighbors are {32,1}, and for other datasets, the sampling numbers of contextual neighbors are {20,1}. We set $\alpha = 0.1$ and $\beta = 1.0$ for UCI and $\alpha = 1$ and $\beta = 0.1$ for LastFM. We set $\alpha = 1$ and $\beta = 10$ for other datasets. We set the dimension of Correlated Spatial-Temporal Positional Encoding as 200, where the dimension of Spatial Distance encoding and Temporal distance is set as 100 and 100, respectively. All the experiments are run on a Linux Ubuntu 18.04 Server with a NVIDIA RTX2080Ti GPU.

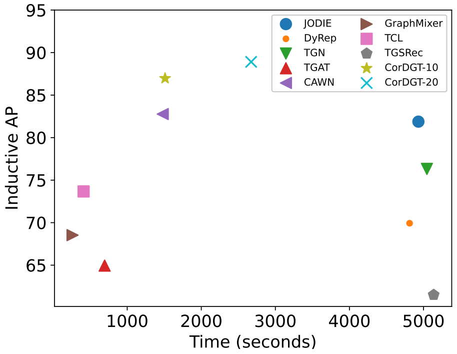


Figure 5: The inductive AP metrics and the training time per epoch on Outdoors datasets. The closer to the upper left corner, the better performance. CorDGT-10 and CorDGT-20 denote the CorDGT with 10 and 20 contextual nodes, respectively.

E ADDITIONAL EXPERIMENTAL RESULTS

E.1 Node classification results

The experiments settings of node classification are as follows. We initially train a model using the transductive link prediction task. Subsequently, we load the trained model and freeze its parameters, then append a classifier on top of it for the purpose of classification. The AUC results of dynamic node classification are presented in Table 6. Note that we replace STPE-C with STPE-U as the positional encodings, since the node classification task only considers single node rather than interaction. Our CorDGT achieves the best performance on both Wikipedia and Reddit datasets compared with other baselines.

E.2 AUC results of link prediction

The AUC results of link prediction are presented in Table 5. As can be seen, our proposed CorDGT consistently outperforms other baselines on both transductive and inductive experiments.

[8] https://github.com/twitter-research/tgn
[9] https://github.com/StatsDLMathsRecomSys/Inductive-representation-learning-on-temporal-graphs
[10] https://github.com/DyGRec/TGSRec/
[11] https://github.com/snap-stanford/CAW
[12] https://github.com/CongWeilin/GraphMixer
[13] https://github.com/yule-BUAA/DyGLib

Table 5: Area Under ROC curve (AUC) results of transductive/inductive link prediction. The values are multiplied by 100. The results of the best and second best performing models are highlighted in bold and underlined, respectively.

	Model	Reddit	Wikipedia	LastFM	UCI	Enron	Social Evo.	Flights
Transductive	JODIE	97.51±0.2	94.67±0.1	70.43±0.6	88.67±0.2	79.65±2.6	91.74±0.5	96.17±1.2
	DyRep	98.03±0.1	94.05±0.4	68.77±2.9	55.97±1.7	73.21±2.2	91.02±0.2	96.84±0.5
	TGAT	98.04±0.1	94.42±0.1	51.81±1.2	79.73±0.1	59.25±0.2	94.37±0.2	94.24±0.2
	TGN-attn	98.67±0.0	98.42±0.1	72.48±2.8	86.35±3.1	79.54±2.6	94.93±0.2	98.06±0.1
	CAWN-attn	98.63±0.0	98.63±0.1	82.36±0.5	89.86±0.3	90.28±0.2	88.21±0.1	98.30±0.0
	TGSRec	88.96±1.5	85.28±0.7	66.67±3.7	71.37±0.6	71.85±2.1	77.47±0.0	95.04±0.3
	TCL	97.67±0.0	96.07±0.2	65.23±1.2	87.85±1.4	76.47±0.8	93.87±0.2	91.37±0.5
	Graphmixer	97.53±0.2	97.14±0.1	75.28±0.1	90.63±0.2	82.76±0.4	94.61±0.1	91.32±0.0
	CorDGT	**99.18±0.1**	**98.84±0.1**	**92.37±0.2**	**95.12±0.4**	**93.28±0.4**	**95.48±0.1**	**98.84±0.0**
	Improve	0.51	0.21	10.01	4.84	3.00	0.55	0.54
Inductive	JODIE	95.02±0.2	92.65±0.3	82.18±1.2	74.74±1.5	75.65±2.6	93.78±0.3	95.36±0.3
	DyRep	95.81±0.3	91.15±0.7	79.81±2.0	48.63±1.3	56.93±2.0	91.43±0.5	93.64±0.9
	TGAT	97.06±0.7	93.08±0.1	53.50±2.4	70.89±0.4	58.18±3.0	93.26±0.6	89.35±0.4
	TGN-attn	97.45±0.1	97.74±0.1	78.61±2.9	82.08±2.7	72.15±2.3	93.54±0.6	96.03±0.4
	CAWN-attn	97.23±0.4	96.43±0.4	88.67±0.6	88.91±0.8	87.24±0.5	84.57±0.3	96.64±0.0
	TGSRec	81.99±4.2	79.30±0.7	67.93±3.0	61.66±1.5	72.32±2.2	63.57±1.5	90.31±0.1
	TCL	93.84±0.1	95.36±0.2	71.32±1.0	92.14±0.1	73.28±1.1	93.85±0.2	84.54±2.0
	Graphmixer	94.65±0.0	97.38±0.0	81.65±0.1	85.79±0.1	76.38±0.7	94.13±0.1	81.98±1.7
	CorDGT	**98.67±0.8**	**98.26±0.3**	**93.01±1.8**	**93.20±0.9**	**92.64±1.9**	**95.69±1.0**	**98.12±0.4**
	Improve	1.24	0.52	4.34	1.06	5.40	1.56	1.48

Table 6: The AUC results of node classification. The values are multiplied by 100. The baseline results are taken from [24]. The best results are highlighted in bold.

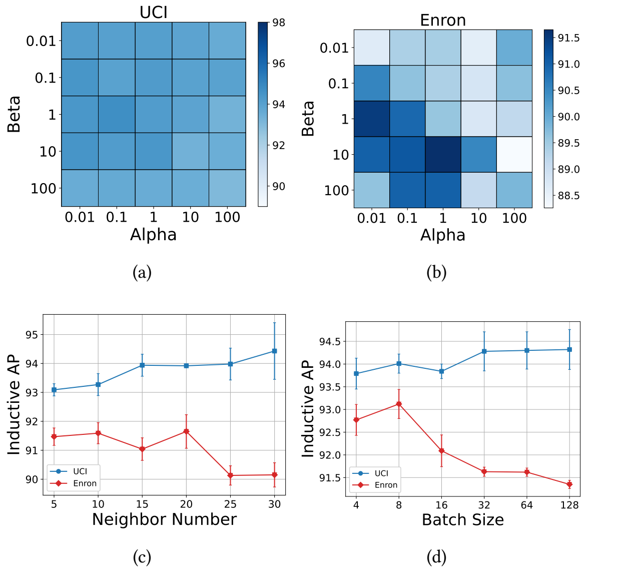
Method	Wikipedia	Reddit
CTDNE	75.89±0.5	59.43±0.6
JODIE	84.84±1.2	61.83±2.7
DyRep	84.59±2.2	62.91±2.4
TGAT	83.69±0.7	65.56±0.7
TGN	87.81±0.3	67.06±0.9
CorDGT	**88.16±0.3**	**70.33±0.6**

E.3 Scalability on Gowalla Outdoors

We further evaluate the scalability of the proposed CorDGT on Gowalla-Food datasets. We train all the models one epoch. The inductive AP and training time per epoch results are presented in Figure 5. We observe that CorDGT with 10 or 20 contextual nodes can obtain the highest inductive AP among baselines. In addition, the training speed of CorDGT with 10 contextual nodes (1511 seconds per epoch) is significantly faster than TGN (5046 seconds per epoch) on Food datasets. This result demonstrate the scalability of the proposed CorDGT .

E.4 Parameters Sensitivity

In this subsection, we evaluate the sensitivity of the proposed CorDGT with respect to some key hyperparameters. The results



(a)

(b)

(c)

(d)

Figure 6: Parameters sensitivity analysis. The inductive AP values are described in all figures. The performance of the model changes with (a) different α and β on UCI dataset, (b) different α and β on Enron dataset, (c) different 1-hop neighbor number, (d) different batch size.

are illustrated in Figure 6, and the observations are as follows: (1) Equation (7) employs coefficients α and β to balance recentness and intensity. We evaluate the influence of α and β on UCI and Wikipedia. As shown in Figure 6(a), for UCI dataset, varying settings of α and β minimally influence the performance of CorDGT. On the other hand, for Enron dataset (Figure 6(b)), the smaller α (≤ 1.0) and larger β (≥ 1.0) enhance the performance of CorDGT. (2) We also investigate the influence of sampled 1-hop contextual node numbers on UCI and Enron datasets, as depicted in Figure 6(c). On the UCI dataset, the model's performance improves with an increasing number of sampled contextual neighbors. In contrast, on the Enron dataset, the model achieves optimal performance with 20 sampled neighbors, possibly due to Enron's smaller node number. (3) Furthermore, we explore the impact of different batch sizes on UCI and Enron datasets, as shown in Figure 6(d). The model's performance remains stable on UCI. However, on the Enron dataset, performance generally deteriorates with increasing sampled neighbor numbers. This discrepancy may stem from Enron's higher average interaction intensity, suggesting that more frequent updates on recent interactions and counts could benefit proximity evaluation.