Tide: A Time-Wise Causal Debiasing Framework for Generative Dynamic Link Prediction

Xin Zhang*
National Key Laboratory of Information Systems Engineering, National University of Defense Technology
Changsha, China
zhangxin16@nudt.edu.cn

Jianming Zheng*
Laboratory for Advanced Computing and Intelligence Engineering
Wuxi, China
zhengjianming12@nudt.edu.cn

Fei Cai†
Zhiqiang Pan
caifei08@nudt.edu.cn
panzhiqiang@nudt.edu.cn
National University of Defense Technology
Changsha, China

Wanyu Chen
College of Electronic Countermeasures, National University of Defense Technology
Hefei, China
wanyuchen@nudt.edu.cn

Chonghao Chen
Honghui Chen†
chenchonghao@nudt.edu.cn
chenhonghui@nudt.edu.cn
National University of Defense Technology
Changsha, China

Abstract

Dynamic link prediction aims to predict the future links in dynamic graphs. Existing generative dynamic link prediction studies utilize the global degree distribution for mitigating the over-estimation problem, which can model the time-invariant features while neglecting the time-varying features, resulting in capturing inaccurate evolution patterns. However, such time related features are intrinsically coupled, which makes simultaneously and independently modeling both features infeasible. Motivated by these issues, we propose a Time-wise causal debiasing framework (**Tide**) for generative dynamic link prediction, which does not resort to any extra trainable modules. Instead, to obtain the time-invariant features, we first utilize a time-invariant deconfounded learning mechanism for decoupling the prediction score with the degree distribution. To leverage the time-varying features, we intervene in the model during the inference stage by a predicted future degree distribution, aiming to make the accurate predictions for dynamic graphs. Experiments conducted on four public datasets under both inductive and transductive settings present that our Tide enhanced models can outperform their corresponding vanilla versions by up to **21.42**% and **27.73**% in terms of NDCG and Jaccard, respectively. [1]

CCS Concepts

• **Information systems** → **Web mining**.

Keywords

Dynamic link prediction, Generative model, Causal inference

ACM Reference Format:
Xin Zhang, Jianming Zheng, Fei Cai, Zhiqiang Pan, Wanyu Chen, Chonghao Chen, and Honghui Chen. 2025. Tide: A Time-Wise Causal Debiasing Framework for Generative Dynamic Link Prediction . In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761182

1 Introduction

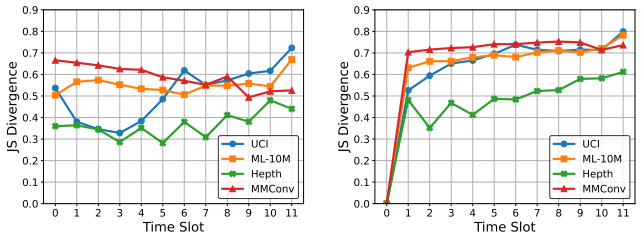
The graph can be utilized for modeling various fields of real-world data, such as social networks [6, 12, 22], citation networks [31, 44] and recommendation systems [10, 33], where the contained entities are considered as nodes and the relationships between them are regarded as edges, respectively [8]. Furthermore, the above graphs usually evolve over time continuously [7], demonstrating the characteristics of dynamic changes. Therefore, dynamic graph link prediction, also called dynamic link prediction, is a widely investigated fundamental task [9, 21], which aims to capture the evolution pattern of temporal graphs and predict future edges.

Current studies for dynamic link prediction can be broadly categorized into two paradigms, i.e., the discriminative approaches and generative approaches. The former always learns a binary classification model to discriminate whether an edge exists between the central node and the given nodes [3, 21, 27]. Inspired by the position encoding and sequential modeling capability of Transformer models [19, 24], the latter typically introduces the Transformer-based framework to carry out dynamic link prediction research, which aims to predict the next neighbor of the central node [28, 37]. Owing to the autoregressive architecture where the historical graph is serialized as the precondition, the Transformer-based generative

*Both authors contributed equally to this research.
†Corresponding Author
[1]Our code is available at: https://github.com/SwaggyZhang/Tide

CIKM '25, Seoul, Republic of Korea.

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761182



(a) Global degree distribution vs. each time slot one. **(b) Initial time slot vs. subsequent time slots.**

Figure 1: Time-varying degree distributions.

approaches have the capability to effectively capture long-term temporal dependencies of the dynamic graphs, thus achieving the state-of-the-art performance. While offering significant advantages, *the autoregressive nature of Transformers inherently exhibits a prediction bias toward high-degree nodes, as it favors those that frequently appear in the historical sequences,* a.k.a. , **node Mathew effect**.

Motivations. Recent improvements in Transformer-based dynamic link prediction models attempt to mitigate the over-estimation of high-degree nodes. For instance, Zhang et al. [37] directly integrate the global degree distribution as a structural prior, explicitly modeling the mutual information between temporal node sequences and current candidate selections. As shown in Fig. 1(a), however, the Jensen-Shannon divergence [4] between the global and temporal degree distribution maintains at the $0.3 - 0.7$ level, demonstrating the unreliability of global-distribution dependencies. Furthermore, the ever-increasing JS divergence between the initial time slot and subsequent slots in Fig. 1(b) further displays the degree distribution dynamics. This implies that the time-varying degree distribution is an intrinsic characteristic of dynamic graphs. Apart from the time-varying features, empirical analysis of real-world dynamic graphs reveals the widespread existence of persistent structural invariants, such as stable community structures in social media and users' long-term preferences for certain-types of items in recommendation systems. In this light, we argue that the **time-varying** and **time-invariant** features should be collaboratively leveraged to construct the dynamic link prediction model. The time-varying features reflect the evolution pattern of a dynamic graph, while the time-invariant features can mitigate the node degree's Matthew effect by maintaining the essence of a dynamic graph.

Challenges. Theoretically, the time-varying and time-invariant features are conceptually orthogonal, which cannot be jointly modeled directly. However, the real-world graph structures often exhibit complex couplings between these two properties. On the one hand, the time-invariant long-term features may influence the time-varying short-term patterns, e.g., the celebrity follow on social media can trigger the emergence of new celebrities. On the other hand, the time-varying features may evolve into the long-term time-invariant attributes. For instance, the trending items in recommendation systems can become perennial bestsellers. This creates a modeling paradox: *how to rigorously preserve conceptual orthogonality in representation learning from the empirical dataset with coupled time features.*

Proposals and Contributions. To overcome the aforementioned challenges, we regard time as a new dimension and propose a Time-wise causal debiasing framework (**Tide**) for generative dynamic link prediction. Specifically, we first construct the structural causal model containing node features, historical sequences, time-varying degree distributions and the prediction results, and then employ *do-calculus* to block the path between node features and time-varying degree distributions in the causal model, which can decouple time-varying degree distributions from time-invariant features, thereby eliminating their confounding negative effects and dealing with the modeling paradox. Subsequently, to enable the model to capture time-varying features in dynamic graphs, we leverage the observed historical degree distributions to predict future distributions. During the inference stage, these predicted distributions are employed to intervene in model outputs, thereby enhancing future link prediction accuracy in dynamic graphs. Experimental results present that our framework improves the NDCG and Jaccard metrics against the state-of-the-art (SOTA) models by up to 21.42% and 27.73%, respectively. The main contributions of our work can be condensed into the following aspects:

(1) To the best of our knowledge, we are the first to investigate the generative dynamic link prediction from the causal inference view, mitigating the node Matthew effect in the Transformer architecture. It should be noted that Tide can be a flexible plug-in for any generative dynamic link prediction work.
(2) We abstract two key factors influencing the dynamic link prediction from an empirical study, namely, **time-invariant** and **time-varying** features, and decouple these two features by introducing the deconfounded learning mechanism.
(3) We intervene in the model to predict the future neighbors with the forecasted time-varying features during inference, the improvements over the SOTA baselines show the effectiveness.

2 Related Work

2.1 Dynamic link prediction

In real-world scenarios, dynamic graphs find widespread applications across various fields, including but not limited to social media [25], event modeling [42, 43] and recommendation systems [35]. This widespread adoption has spurred substantial research efforts toward improving dynamic graph neural networks, particularly in architecture design [40] and temporal modeling [3]. These networks explicitly model graph dynamics by jointly capturing structural evolution and feature transitions over time. A lot of approaches have been proposed to tackle the unique challenges arising from dynamic graph structures. Existing approaches for dynamic graphs can be categorized into Graph Neural Network-based (GNN) and Non-GNN-based models. The GNN-based methods usually leverage GNN to model the structural characteristic of dynamic graphs with the message passing mechanism, and they design a module to capture the pattern of temporal evolution [15, 23, 26, 27]. Motivated by Transformers' exceptional ability for long sequence modeling, researchers have proposed Transformer-based architectures [28, 37] that perform dynamic link prediction without relying on GNN frameworks. Specifically, Wu et al. [28] propose to reformulate the dynamic graph link prediction task into generating the next token of a sequence, and incorporate temporal alignment into the input

sequences. Subsequently, Zhang et al. [37] fuse the topology of subgraphs into the mask matrices of sequences to help the model understand the relationships between neighbor nodes.

2.2 Causal Inference

Causal inference [18] constitutes a field of study dedicated to quantifying the causal relationships among variables, which is widely used for mitigating the negative bias in various real-world applications [15], such as recommendation systems [34, 38], graph analysis [32, 41], and natural language processing [14].

For example, Zhang et al. [38] leverage do-calculus to remove the popularity bias during training and introduce the predicted popularity for inference. Additionally, Yoo et al. [34] propose a disentanglement method that integrates temporal popularity dynamics. They introduce a new temporal-aware embedding design for both users and items, based on which they propose popularity coarsening and bootstrapping to enhance generalization. Zhao et al. [41] enhances the model's generalization on out-of-distribution data by eliminating environmental confounding factors and learning invariant graph representations. However, these methods only focus on debiasing for the GNN-based models or collaborative filtering-based models. Inspired by these approaches, we develop a causal inference-enhanced framework for dynamic graph link prediction, specifically designed to mitigate the negative effects of imbalanced degree distributions on autoregressive models.

3 Preliminary

3.1 Traditional dynamic link prediction

Formally, a dynamic graph can be denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$. Here, $\mathcal{V} = \{v_i \mid i = 1, 2, \ldots, |\mathcal{V}|\}$ is the node set, $\mathcal{E} = \{e_{ij} \mid e_{ij} = (v_i, v_j, t_\tau), \tau \in [1, 2, \ldots, \mathcal{T}]\}$ is the edge set, where each element (v_i, v_j, t_τ) denotes that an edge e_{ij} connects nodes v_i and v_j at the timestamp t_τ. In addition, $\mathcal{F}$ denotes the feature matrix, which may represent the node/edge features in each dataset.

Given a dynamic graph $\mathcal{G}$, the aim of dynamic link prediction task is to train a model $f(\mathcal{G}; \theta)$ with learnable parameters θ, capturing the temporal evolution pattern of $\mathcal{G}$ so that to predict the new edge set $\mathcal{E}[\cdot;\cdot;t_{\mathcal{T}+1}]$ at the future timestamp $t_{\mathcal{T}+1}$:

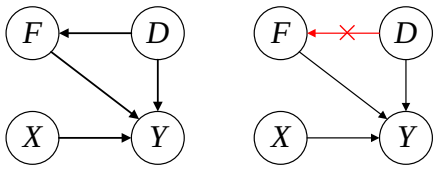
$$\begin{cases} \theta^* = \arg\min_{\theta \in \Theta} \sum_{\tau=2}^{\mathcal{T}} \ell(\mathcal{E}[\cdot;\cdot;t_\tau], f(\mathcal{G}_{t_\tau}; \theta)) \\ \mathcal{E}[\cdot;\cdot;t_{\mathcal{T}+1}] = f(\mathcal{G}_{t_\mathcal{T}}; \theta^*) \end{cases}, \tag{1}$$

where θ^* is the parameters that minimize the loss $\ell(\cdot,\cdot)$ on the training set, Θ denotes the parameter space, $\mathcal{E}[\cdot;\cdot;t_\tau]$ and $\mathcal{G}_{t_\tau}$ denote the edge set and the graph until the timestamp t_τ, respectively.

3.2 Generative dynamic link prediction

In addition, to capture the inherent temporal evolution patterns within dynamic graphs and streamline the modeling process of their evolution, dynamic link prediction is re-conceptualized as a sequence generation task, which consists of two main stages: dynamic graph serialization and generative link prediction.

3.2.1 Dynamic graph serialization. To enable generative link prediction for dynamic graphs, the main adopted solution is segmenting the whole original dynamic graph into the 1-hop subgraphs of each node, and transforming the subgraph into a token sequence according to the chronological order of interactions. For instance, given the center node v_i and its 1-hop subgraph $\mathcal{G}^{(i)}$, the serialized format of $\mathcal{G}^{(i)}$ can be represented as:



(a) Conventional methods (b) Removing degree bias

Figure 2: Structural causal models for generative dynamic link prediction.

$$\begin{aligned} x^{(i)} =& [\langle hist \rangle, v_i, \langle time_1 \rangle, S_i^1, \ldots, \langle time_m \rangle, S_i^m, \ldots, \\ & \langle time_{(T-1)} \rangle, S_i^{T-1}, \langle endofhis \rangle] \\ y^{(i)} =& [\langle pred \rangle, \langle time_T \rangle, S_i^T, \langle endofpre \rangle], \end{aligned} \tag{2}$$

where $x^{(i)}$ is the input sequence, $y^{(i)}$ is the ground truth, $\langle\cdot\rangle$ denotes a special token, $\langle time_m \rangle$ marks the start of time slot m, $\langle hist \rangle$ and $\langle endofhis \rangle$ denote the start and end of the known historical sequence, respectively. Similarly, $\langle pred \rangle$ and $\langle endofpre \rangle$ denote the start and end of the ground truth sequence, respectively. Moreover, T denotes the number of uniformly partitioned time slots across the entire temporal span $\mathcal{T}$. In addition, S_i^m, the subsequence of v_i's neighbors in the m-th time shot, can be formulated as:

$$S_i^m = [v_i^{t_{m,1}}, v_i^{t_{m,2}}, \ldots, v_i^{t_{m,|S_i^m|}}], m \in \{1, 2, \ldots, T\}. \tag{3}$$

3.2.2 Generative link prediction. For each subgraph $\mathcal{G}^{(i)}$, we can abstract the input sequence $x^{(i)}$ in Eq. (2) into the following form:

$$\mathbf{x}^{(i)} = [h_1^{(i)}, h_2^{(i)}, \ldots, h_k^{(i)}, \ldots h_{n_i}^{(i)}] \in \mathbb{R}^{n_i}, \tag{4}$$

where $\mathbf{x}^{(i)}$ consists of n_i token ids, comprising both regular node tokens and special tokens introduced in Sec. 3.2.1, and $h_k^{(i)}$ denotes the k-th token in the sequence $\mathbf{x}^{(i)}$. We use the notation $\mathbf{h} \in \mathbb{R}^{dim}$ to represent the dim-dimensional embedding of token h in Eq. (4). Thus, $\mathbf{x}^{(i)}$ can be converted into a hidden state matrix $\mathbf{H}^{(i)}$:

$$\mathbf{H}^{(i)} = \left[\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \ldots, \mathbf{h}_{n_i}^{(i)}\right] \in \mathbb{R}^{n_i \times dim}. \tag{5}$$

Subsequently, given the input sequences $[x_1, \ldots, x_{|\mathcal{V}|}]$, the goal of generative dynamic link prediction is to accurately generate the next node tokens. Thus, the optimization goal in the training stage can be defined as:

$$\begin{cases} \mathcal{L} = -\sum_{i=1}^{|\mathcal{V}|} \sum_{k=1}^{|\mathbf{x}^{(i)}|} \log P_\theta\left(h_k^{(i)} | \mathbf{x}_{<k}^{(i)}\right) \\ P_\theta\left(h_k^{(i)} | \mathbf{x}_{<k}^{(i)}\right) = \textsc{Softmax}\left(\text{LN}\left(\mathbf{H}_{<k}^{(i)}\right) \mathbf{W}_{\text{token}}\right) \end{cases}, \tag{6}$$

where $\mathbf{x}_{<k}^{(i)}$ denotes the sub-sequence of $\mathbf{x}^{(i)}$ containing the first $(k-1)$ tokens, $\mathbf{H}_{<k}^{(i)}$ denotes the embedding of $\mathbf{x}_{<k}^{(i)}$, P_θ represents the probability of generating the k-th token from the learnable

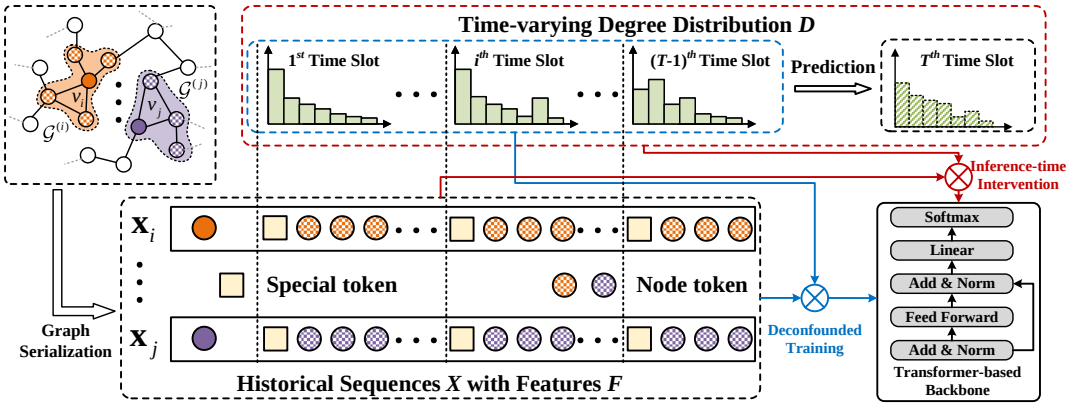


Figure 3: The pipeline of Tide framework. The blue and red arrows represent the training and inference stages, respectively.

matrix $\mathbf{W}_{\text{token}} \in \mathbb{R}^{dim \times |\text{token}|}$. Moreover, $|\text{token}| = |\mathcal{V}| + |\text{spt}|$ denotes the total number of tokens, which equals the sum of node count $|\mathcal{V}|$ and special token count $|\text{spt}|$.

4 Approach

4.1 Overview

In this section, we first analyze the degree bias in Sec. 4.2, then we detail the Tide framework, which consists of two main mechanisms, i.e., time-invariant deconfounded learning (see Sec. 4.3) and time-varying intervened inference (see Sec. 4.4). The pipeline of Tide is shown in Fig. 3. First, we preprocess the original graph following the procedure in Sec. 3.2.1. Subsequently, we compute the node distributions within each time segment. On this basis, our key innovation lies in establishing a causality-informed processing pipeline: (1) During the training stage, we decouple the historical node distributions from the model prediction scores to facilitate the learning of time-invariant features; (2) At the inference stage, we leverage the historical distributions to predict future temporal patterns, then integrate these time-varying characteristics into the input sequences to generate future neighbor accurately.

4.2 A causal view of degree bias

4.2.1 Analyze degree bias with causal graphs. To investigate the bias induced by the degree distributions in dynamic link prediction, we employ causal graphs [18], which represents the variables as nodes and their relationships as edges. Specifically, we construct the conventional model discarding the degree distributions as well as our proposed debiased model, which are shown in Fig. 2(a) and Fig. 2(b), respectively. Four variables in both models presented in Fig. 2 are described as follows:

- Node F denotes the **f**eatures of nodes.
- Node D denotes the time-varying **d**egree distributions of nodes in the dynamic graph.
- Node X denotes the historical sequence x defined in Eq. (2) at the current timestamp.
- Node Y denotes the next predicted node.

Moreover, Fig. 2 reveals two main causal relationships that influence the next-node prediction probability in conventional generative dynamic link prediction models:

- The paths $\{D, X, F\} \rightarrow Y$ represent that the next node is determined by the combination of three factors, i.e., the node features F, the time-varying degree distributions D, and the historical sequence X.
- The path $D \rightarrow F \rightarrow Y$ represents that the node features F are affected by the time-varying degree distributions D and then influence the prediction result Y.

Furthermore, there exist two causal paths from the degree distributions D to the prediction result Y, i.e., $D \rightarrow Y$ and $D \rightarrow F \rightarrow Y$ in the causal graph. The first path illustrates that the time-varying degree distributions influence the prediction probability, since the degree can reflect the importance of the node to some extent [1]. The second path indicates that the time-varying degree distributions D affect the node occurrence frequencies in each dataset, which subsequently bias the learning of node features F, amplifying the auto-regressive model's propensity to predict the high-degree nodes [30]. Obviously, the amplification should be avoided since a link prediction model should capture the evolution pattern reliably and is immune to the node degrees. Therefore, the second path $D \rightarrow F \rightarrow Y$ affects the estimation of the time-varying degree distributions' impact on the prediction result Y.

4.2.2 Analysis on conventional methods. Existing methods suffer from the time-varying degree distributions D when learning the dynamic graph evolution pattern $\{X, F\} \rightarrow Y$. Specifically, the conditional probability $P(Y|X,F)$ in conventional generative methods can be modeled as the following form:

$$\begin{aligned} P(Y|X,F) &\overset{(1)}{=} \sum_d P(Y,d|X,F) \\ &\overset{(2)}{=} \sum_d P(Y|X,F,d)P(d|X,F) \\ &\overset{(3)}{=} \sum_d P(Y|X,F,d)P(d|F) \\ &\overset{(4)}{\propto} \sum_d P(Y|X,F,d)P(F|d)P(d). \end{aligned} \tag{7}$$

Below explains the derivation step by step:

- (1) is the definition of margin distribution;
- (2) is because of Bayes' theorem;

- (3) is because X is independent of D in Fig. 2(a);
- (4) is because of Bayes' theorem.

From Eq. (7), we can observe a term $P(F|d)$, which illustrates the influence of the time-varying degree distributions D on the node features F, and thus affecting the predicted next node Y.

4.3 Time-invariant deconfounded learning

We now consider how to obtain a model that is immune to the impact of $D \rightarrow F$. To achieve our goal of removing the degree bias D while learning $\{X, F\} \rightarrow Y$, we adopt the *do*-calculus [18] in causal science. Specifically, the do-calculus is used to eliminate the impact of node D (the parent of node F). Thus, we formulate the predictive model as $P(Y|do(X, F))$ rather than $P(Y|X, F)$ modeled by Eq. (7). Let the causal graph of conventional methods shown in Fig. 2(a) be G, and the intervened causal graph shown in Fig. 2(b) be G'. Then, performing *do*-calculus on G leads to:

$$\begin{aligned} P(Y|do(X,F)) &\overset{(1)}{=} P_{G'}(Y|X,F) \\ &\overset{(2)}{=} \sum_d P_{G'}(Y|X,F,d)P_{G'}(d|X,F) \\ &\overset{(3)}{=} \sum_d P_{G'}(Z|X,F,d)P_{G'}(d) \\ &\overset{(4)}{=} \sum_d P(Y|X,F,d)P(d), \end{aligned} \tag{8}$$

where $P_{G'}$ denotes the probability evaluated on G'. The derivation is explained step by step below:

- (1) is because of the backdoor criterion [18] since the backdoor path of $F \leftarrow D \rightarrow Y$ in causal graph G is blocked by $do(X, F)$;
- (2) is because of Bayes' theorem;
- (3) is because that $\{X, F\}$ are independent of D in G';
- (4) $P_{G'}(Y|X, F, d) = P(Y|X, F, d)$ is because that the causal mechanism $\{X, F, D\} \rightarrow Z$ is not changed when cutting of the edge $D \rightarrow Y$, and $P(d) = P_{G'}(d)$ since the degree distribution D has the same prior on the two causal graphs.

Comparing Eq. (8) with Eq. (7), we can find that the term $P(F|d)$ in Eq. (7) disappears. Such a phenomenon demonstrates that by applying the *do*-calculus on the variable node F, we can obtain the debiased prediction probability as $P(Y|do(X, F)) = \sum_d P(Y|X, F, d)P(d)$. Specifically, this eliminates the impact of the bias of degree distributions, i.e., $\{D \rightarrow Y\}$, thus learning the **time-invariant** features. Then, we aim at estimating $P(Y|do(X, F))$ from the training data, including two steps, i.e., estimating $P(Y|D, X, F)$ and $\sum_d P(Y|X, F, d)P(d)$, respectively.

Step 1: *Estimating* $P(Y|D, X, F)$. This conditional probability function quantifies the probability $P\left(y|x, f, d_{[:(T-1)]}\right)$ of generating y as the next token of sequence x, given an input sequence $X = x$, the node features $F = f$, and the degree distributions $D = d_{[:,(T-1)]}$ up to the $(T-1)$-th time slot of x. This can be implemented by the generative dynamic link prediction model with the learnable parameters θ introduced in Sec. 3.2.2, where the parameters θ can be optimized by $\mathcal{L}$ in Eq. (6).

Furthermore, given the degree distribution as d, to learn the **time-invariant** features, we propose to decouple the degree distribution d from the modeling processing of predicting Y, i.e., $P\left(y|x, f, d_{[:(T-1)]}\right)$. Since the embedding matrix $\mathbf{H}$ of sequence x is derived from features f, Eq. (6) allows the transformation of the conditional probability $P_\theta\left(y|x, f, d_{[:(T-1)]}\right)$ into $P_\theta\left(y|\mathbf{x}, d_{[:(T-1)]}\right)$. Thus, we design a decoupled prediction function by improving the conditional probability in Eq. (6) as follows:

$$\begin{cases} P_\theta\left(y|\mathbf{x}_{<k,m}, d_{[:m]}\right) = \sigma\left(score_\theta\left(\mathbf{x}_{<k,m}\right)\right) \times {d'_m}^\gamma \\ score_\theta\left(\mathbf{x}_{<k,m}\right) = \left(\mathrm{LN}\left(\mathbf{H}^{(i)}_{<k}\right)\mathbf{W}_{\text{token}}\right) \in \mathbb{R}^{dim \times |\text{token}|} \\ d'_m = \textsc{Concat}(d_m, C \cdot \mathbf{1}) \end{cases}, \tag{9}$$

where m denotes the last time slot of sequence $\mathbf{x}_{<k}$, $d_m \in \mathbb{R}^{|\mathcal{V}|}$ denotes the node degree distribution in the m−th time slot. Subsequently, since our primary objective is predicting the neighbor nodes, we uniformly set the frequency of special tokens to a small constant C, whose distribution represented in vector form is $C \cdot \mathbf{1} \in \mathbb{R}^{|\text{spt}|}$. Moreover, $d'_m \in \mathbb{R}^{|\text{token}|}$ is the vector representing the token distribution across all tokens in the m-th time slot, formed by the concatenation of vectors d_m and $C \cdot \mathbf{1}$. Additionally, we choose the state-of-the-art generative dynamic link prediction model TopDyG [37] as our backbone to implement the $score_\theta(\cdot)$.

Besides, the hyper-parameter γ is used for smoothing the degree distribution: a hyper-parameter γ controls the smoothing of the degree distribution: when $\gamma = 0$, the Eq. (9) collapses to the naive conditional probability that disregards the degree distribution, i.e., $P_\theta\left(y|\mathbf{x}_{<k}\right)$. The function $\sigma(\cdot)$ denotes the activation function $ELU'(\cdot)$ [2, 38] with range $(0, +\infty)$, thereby ensuring strict monotonicity of $P_\theta\left(y|\mathbf{x}_{<k,m}, d_{[:m]}\right)$ since all elements in d are positive. Specifically, taking the degree distribution in the m−th time slot d_m as an instance, it can be formally expressed as:

$$\begin{aligned} d_m &= [p_m(v_1), p_m(v_2), \ldots, p_m(v_i), \ldots, p_m(v_{|\mathcal{V}|})], \\ p_m(v_i) &= \begin{cases} \dfrac{\textsc{Degree}_m(v_i)}{\sum_{k=1}^{|\mathcal{V}|} \textsc{Degree}_m(v_k)}, \textsc{Degree}_m(v_i) > 0 \\ C \quad , \text{otherwise} \end{cases}, \end{aligned} \tag{10}$$

where $\textsc{Degree}_m(v_i)$ and $p_m(v_i)$ denote the degree and the relative frequency of node v_i in the m−th time slot, respectively. Note that we omit the normalization of $P_\theta\left(y|\mathbf{x}_{<k,m}, d_{[:m]}\right)$, though this normalization choice does not affect the candidate token ranking during the future neighbor generation.

Step 2: *Estimating* $\sum_d P(Y|X, F, d)P(d)$. Then, we concentrate on estimating the interventional probability $P(Y|do(X, F))$. However, considering that the value space of the random variable D is too large, exact summation for prediction is computationally intractable. Fortunately, we can leverage the following reduction to eliminate the need for this summation:

$$\begin{aligned} P(Y|do(X,F)) &\overset{(1)}{=} \sum_d P(Y|X,F,d)P(d) \\ &\overset{(2)}{=} \sum_d \left(\sigma\left(score_\theta(\mathbf{x})\right) \times d^\gamma\right) \times P(d) \\ &\overset{(3)}{=} \sigma\left(score_\theta(\mathbf{x})\right) \sum_d \left(d^\gamma \times P(d)\right) \\ &\overset{(4)}{=} \sigma\left(score_\theta(\mathbf{x})\right) \mathbb{E}\left(d^\gamma\right) \end{aligned} \tag{11}$$

where $\mathbb{E}(d^Y)$ denotes the expectation of d^Y, which is a constant. Considering that $P(Y|do(X,F))$ is used for choosing the next token from the vocabulary according to the predicted scores, the value of $\mathbb{E}(d^Y)$ does not change the ranking of the candidate tokens. Thus, we can adopt $\sigma(score_\theta(\mathbf{x}))$ to estimate $P(Y|do(X,F))$.

To sum up, we train the model with parameters θ by fitting the historical sequences in each dynamic graph to capture the **time-invariant** features, i.e., $\{X,F\} \rightarrow Y$ by removing the negative impact of time-varying degree distributions, i.e., $D \rightarrow Y$.

4.4 Time-varying intervened inference

By estimating the probability $P(Y|do(X,F))$, we eliminate the negative effect of the degree distributions bias, thus accurately capturing the **time-invariant** features from the historical data. Then, after training on historical data from the previous $T-1$ time slots, we propose to leverage the time-varying degree distributions bias to intervene in the inference for the T−th slot, as this inherent **time-varying** characteristics of dynamic graphs can guide the model toward more accurate predictions. To achieve the target, we only need perform the intervention $D = \hat{d}_T$ on the model during inference stage as:

$$P(Y|do(X=x,F=f),do(D=\hat{d}_T)) = P(Y=y|x,f,\hat{d}_T), \quad (12)$$

where $\hat{d}_T$ denotes the degree distribution in T−th time slot. Such an intervention in probability directly equals the conditional probability since we have removed the backdoor path between D and Y in the causal graph. Furthermore, since predicting future node degree distributions is not our primary research focus, we only employ a lightweight time-series forecasting method to compute $\hat{d}$ as:

$$\hat{d}_T = d_{(T-1)} + \beta\left(d_{(T-1)} - d_{(T-2)}\right), \quad (13)$$

where $d_{(T-1)}$ is the degree distribution in the $(T-1)$−th time slot, i.e., the nearest slot to the time slot to be inferred, and β is the hyper-parameter to balance the contribution of the distributions drift in the prediction. Therefore, during the inference phase, the conditional probability after incorporating the predicted future degree distribution via Eq. (12) is given by:

$$P\left(Y=y|x,f,\hat{d}\right) = \sigma\left(score_\theta\left(y|\mathbf{x},d_{[:(T-1)]}\right)\right) \times \hat{d}_T^{\hat{\gamma}}, \quad (14)$$

where $\hat{\gamma}$ denotes the smoothing hyper-parameter used for prediction. Note that for the implementation convenience, we set $\hat{\gamma}$ equal to the value in the training phase ($\hat{\gamma} = \gamma$), which also yields competitive performance. When $\hat{\gamma} = 0$, the model degenerates to the time-invariant deconfounded learning-only configuration without the predicted degree distribution intervention.

5 Experiment

We validate the effectiveness of our Tide framework by answering the following research questions (RQs):

- **RQ1** Can the Tide framework improve the performance of generative models on the dynamic link prediction task?
- **RQ2** What is the contribution of different contained components in Tide to the performance?
- **RQ3** How do different construction methods of time-varying features affect dynamic link prediction performance?
- **RQ4** How do different debiasing methods affect generative dynamic link prediction performance?
- **RQ5** How is the performance of generative models affected by different weights of the time-varying characteristic?

Table 1: Statistics of the datasets used in our experiments.

Dataset	UCI	ML-10M	Hepth	MMConv
Bipartite	✗	✗	✗	✓
Learning setting	Transductive	Transductive	Inductive	Transductive
# Nodes	1,781	15,841	4,737	7,415
# Edges	16,743	48,561	14,831	91,986
# Time slots	13	13	12	16

5.1 Experiment setup

5.1.1 Evaluation datasets. Following the previous work [28, 37], we adopt the following four public datasets, including UCI [16], ML-10M [5], Hepth [11] and MMConv [13], to examine the models' performance. The statistics of the discussed datasets are shown in Table 1. To guarantee the fairness of comparative experiments, we adopt graph serialization and temporal alignment methods as described in SimpleDyG [28] (the first work on generative dynamic link prediction), and we choose the same evaluation metrics as those used in SimpleDyG.

5.1.2 Model configurations. Following the previous research [28, 37], we adopt the decoder-only language model GPT-2 [19] as the generative model. The learning rate, number of model layers, and hidden layer dimensions are configured identically to SimpleDyG's experimental setup. For all four datasets, we consistently model the original dynamic graphs as undirected graphs for input, regardless of their inherent directionality. Furthermore, we adhere to the data partitioning protocol in SimpleDyG. Specifically, given a dataset uniformly divided into T temporal segments: (1) Training utilizes historical data from the first $(T-2)$ segments; (2) Validation takes the training set as input with the $(T-1)$-th segment as the ground truth for model selection; (3) Testing utilizes the historical $(T-1)$ segments as input while evaluating against the T-th segment, i.e., ground truth. All experiments are conducted using PyTorch 1.13.1 on an Ubuntu 22.04 server with an NVIDIA GeForce RTX 3090 GPU (24 GB memory).

5.1.3 Model summary. Since our work focuses on dynamic graphs, we explicitly exclude methods designed for static graphs from our comparative analysis, such as SEAL [36]. Additionally, although LLM-based approaches have emerged [39], we exclude them from our discussion due to their prohibitive computational requirements. Therefore, the models discussed in this paper are as follows: (1) Two discrete-time GNN-based models, i.e., DySAT [21] and EvolveGCN [17]; (2) Six continuous-time GNN-based models, i.e., DyRep [23], JODIE [9], TGAT [29], TGN [20], TREND [27] and GraphMixer [3]; (3) Two Transformer-based generative models, i.e., SimpleDyG [28] and TopDyG [37].

5.2 Overall performance (RQ1)

To answer **RQ1**, we examine the dynamic graph link prediction performance of our proposals as well as the baseline models in Table 2, from which we can obtain the following observations:

Table 2: Performance of dynamic link prediction of our proposal and the baselines on four datasets. (In each column, the best result is bolded and the runner-up is underlined. All the performance data of baselines is obtained from previous work [28, 37].)

	UCI		ML-10M		Hepth		MMConv	
	NDCG@5	Jaccard	NDCG@5	Jaccard	NDCG@5	Jaccard	NDCG@5	Jaccard
DySAT	0.010±0.003	0.010±0.001	0.058±0.073	0.050±0.068	0.007±0.002	0.005±0.001	0.102±0.085	0.095±0.080
EvolveGCN	0.064±0.045	0.032±0.026	0.097±0.071	0.092±0.067	0.009±0.004	0.007±0.002	0.051±0.021	0.032±0.017
DyRep	0.011±0.018	0.010±0.005	0.064±0.036	0.038±0.001	0.031±0.024	0.010±0.006	0.140±0.057	0.067±0.025
JODIE	0.022±0.023	0.012±0.009	0.059±0.016	0.020±0.004	0.031±0.021	0.011±0.008	0.041±0.016	0.032±0.022
TGAT	0.061±0.007	0.020±0.002	0.066±0.035	0.021±0.007	0.034±0.023	0.011±0.006	0.089±0.033	0.058±0.021
TGN	0.041±0.017	0.011±0.003	0.071±0.029	0.023±0.001	0.030±0.012	0.008±0.001	0.096±0.068	0.066±0.038
TREND	0.067±0.010	0.039±0.020	0.079±0.028	0.024±0.003	0.031±0.003	0.010±0.002	0.116±0.020	0.060±0.018
GraphMixer	0.104±0.013	0.042±0.005	0.081±0.033	0.043±0.022	0.011±0.008	0.010±0.003	0.172±0.029	0.085±0.016
SimpleDyG	0.104±0.010	0.092±0.014	0.138±0.009	0.131±0.008	0.035±0.014	0.013±0.006	0.184±0.012	0.169±0.010
SimpleDyG$_{[\text{Tide}]}$	0.120±0.011	0.098±0.004	<u>0.147±0.001</u>	**0.139±0.001**	0.041±0.005	0.014±0.001	0.220±0.008	0.203±0.010
TopDyG	<u>0.149±0.007</u>	<u>0.101±0.006</u>	0.144±0.002	<u>0.135±0.002</u>	<u>0.042±0.004</u>	<u>0.017±0.002</u>	<u>0.242±0.040</u>	<u>0.218±0.030</u>
TopDyG$_{[\text{Tide}]}$	**0.150±0.010**	**0.129±0.006**	**0.148±0.001**	**0.139±0.001**	**0.051±0.005**	**0.019±0.002**	**0.266±0.009**	**0.233±0.005**

Table 3: Ablation studies of Tide for the link prediction tasks. The biggest drop in each column is marked with ↓.

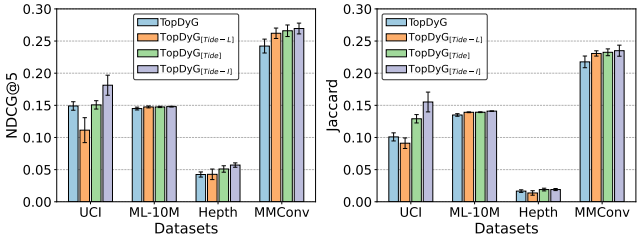
	UCI		ML-10M		Hepth		MMConv	
	NDCG@5	Jaccard	NDCG@5	Jaccard	NDCG@5	Jaccard	NDCG@5	Jaccard
TopDyG$_{[\text{Tide}]}$	**0.150±0.010**	**0.129±0.006**	**0.148±0.001**	**0.139±0.001**	0.051±0.005	0.019±0.002	**0.266±0.009**	**0.233±0.005**
w/o d.l.	0.147±0.023	0.084±0.002	0.136±0.012	0.124±0.015	**0.056±0.011**	**0.020±0.004**	0.243±0.045↓	0.216±0.033↓
w/o int	0.091±0.012↓	0.079±0.009↓	0.100±0.029↓	0.083±0.036↓	0.025±0.003↓	0.008±0.001↓	0.247±0.045	0.217±0.030
TopDyG	0.149±0.007	0.101±0.006	0.144±0.002	0.135±0.002	0.042±0.004	0.017±0.002	0.242±0.040	0.218±0.030
SimpleDyG$_{[\text{Tide}]}$	0.120±0.011	**0.098±0.004**	0.147±0.001	**0.139±0.001**	0.041±0.005	0.014±0.001	**0.220±0.008**	**0.203±0.010**
w.o d.l.	**0.133±0.019**	0.085±0.006	**0.148±0.001**	0.128±0.014	**0.052±0.006**	**0.018±0.003**	0.190±0.025↓	0.176±0.022↓
w/o int	0.079±0.009↓	0.068±0.005↓	0.039±0.013↓	0.035±0.011↓	0.025±0.001↓	0.009±0.001↓	0.215±0.010	0.197±0.012
SimpleDyG	0.104±0.010	0.092±0.014	0.138±0.009	0.131±0.008	0.035±0.014	0.013±0.006	0.184±0.012	0.169±0.010

- **Tide effectively improves the generative models' accuracy on the dynamic link prediction task.** Specifically, the models SimpleDyG$_{[\text{Tide}]}$ and TopDyG$_{[\text{Tide}]}$ are the variants of SimpleDyG and TopDyG running under the Tide framework, respectively. Both variants demonstrate obvious performance improvements in terms of NDCG and Jaccard metrics. Tide framework helps the models achieve up to a 27.73% performance improvement over the original baseline models on the three transductive datasets. Even on the Hepth dataset, the Tide enables the two models to achieve performance improvements up to 21.42%.
- **Tide enhances the diversity of prediction outputs in generative models.** TopDyG$_{[\text{Tide}]}$ demonstrates greater performance gains in terms of Jaccard metric compared to NDCG on the UCI dataset, and this pattern is consistently observed across ML-10M and MMConv datasets. Additionally, SimpleDyG$_{[\text{Tide}]}$ shows the same tendency on MMConv dataset. This phenomenon indicates that the Tide enables models to generate more diverse outputs by mitigating the adverse effects of high-frequency tokens.
- **Tide framework enhances the stability of generative link prediction.** Specifically, SimpleDyG$_{[\text{Tide}]}$ model demonstrates obviously reduced standard deviations of 88% (NDCG) and 87.5% (Jaccard) compared to the original SimpleDyG when tested on the ML-10M dataset. For the inductive Hepth dataset, SimpleDyG$_{[\text{Tide}]}$ achieves 64% and 83% reductions in standard deviation for NDCG and Jaccard metrics respectively, relative to the original SimpleDyG. For TopDyG$_{[\text{Tide}]}$, it shows obviously reduced standard deviations of 77.5% (NDCG) and 83.3% (Jaccard) compared to the original TopDyG when examined on the MMConv dataset.

5.3 Ablation study (RQ2)

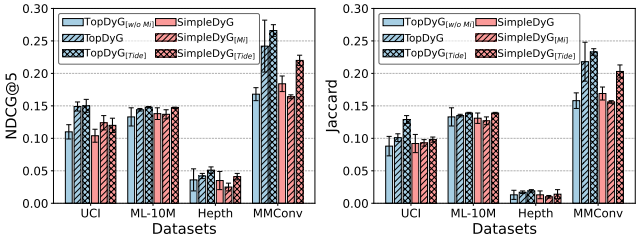
To answer **RQ2**, we conduct an ablation study to thoroughly investigate the contribution of each component of our proposed Tide framework. Specifically, we compare the Tide-enhanced models with their two variants: (1) w/o d.l., which replaces the time-invariant deconfounded learning mechanism with the vanilla training method; (2) w/o int., which removes the time-varying intervened inference mechanism.

As shown in Table 3, "w/o d.l." leads to the biggest drop of both NDCG and Jaccard on MMConv dataset. MMConv exhibits a bipartite graph structure, in which the nodes are strictly partitioned into two disjoint sets. The time-invariant deconfounded learning mechanism can help the model to decouple the confounded time-varying features. On the other hand, though the variant "w/o d.l." still retains the time-varying intervened inference strategy, the bias caused by the degree distribution is difficult to correct. Surprisingly, the removal of time-invariant deconfounded learning ("w/o d.l.") yields better results than complete Tide frameworks for both backbones on Hepth. The anomalous result suggests that in the inductive



(a) NDCG@5 (b) Jaccard

Figure 4: Performance comparison to examine the impact of intervention techniques.



(a) NDCG@5 (b) Jaccard

Figure 5: Performance comparison to examine the impact of debiasing techniques.

setting, the debiased representations learned via time-invariant deconfounded learning strategy may be redundant, which ensures the stability while partially constraining generalization capability on unseen nodes. Additionally, for SimpleDyG$_{[\text{Tide}]}$, although its variant "w/o d.l." obtains higher NDCG scores on UCI and ML-10M datasets than the complete version, the Jaccard scores declined, indicating that the important role of time-invariant deconfounded learning in mitigating the negative effect of the degree bias.

In addition, for both TopDyG$_{[\text{Tide}]}$ and SimpleDyG$_{[\text{Tide}]}$, "w/o int." causes the largest drop of both NDCG and Jaccard on UCI, ML-10M and Hepth datasets, and the performance degradation is particularly pronounced on the Hepth dataset. Both variants perform even worse than their respective original versions. The phenomenon indicates that the time-varying intervened inference mechanism indeed helps the model capture the time-varying features of the datasets. Moreover, in the Hepth dataset's strict inductive setting, central nodes in test stage are unseen during training, which means that their historical sequences do not exist. Thus, predicting their neighbors requires not only features of the unseen nodes but also time-varying characteristics of the seen nodes.

5.4 Impact of intervention method (RQ3)

To investigate the effect of different types of time-varying intervened inference methods, we modify the future time-varying features in Eq. (14), i.e., $\hat{d}_T$. The results are shown in Fig. 4.

We leverage three methods to predict the degree distributions of the testing time slot. Specifically, TopDyG$_{[\text{Tide-L}]}$ takes the degree distribution in the last seen time slot as the future degree distribution; TopDyG$_{[\text{Tide}]}$ uses Eq. (13) to predict the future distribution, and TopDyG$_{[\text{Tide-I}]}$ denotes the ideal condition, which leverages the true degree distribution in the next time slot to guide the model to generate future neighbors. As clearly demonstrated in Fig 4, TopDyG$_{[\text{Tide-I}]}$ consistently outperforms all baselines across all datasets in terms of both NDCG and Jaccard metrics, which is under the ideal condition. TopDyG$_{[\text{Tide-L}]}$ demonstrates comparable performance to the original TopDyG, with no significant disparity observed between the two variants in general, although it outperforms the original version on MMConv but underperforms on UCI dataset. For TopDyG$_{[\text{Tide}]}$, it predicts the future with the distributions of the last two known time slots. Although there is still a gap between TopDyG$_{[\text{Tide}]}$ and ideal condition TopDyG$_{[\text{Tide-I}]}$, TopDyG$_{[\text{Tide}]}$ has outperformed the vanilla TopDyG and the variant TopDyG$_{[\text{Tide-L}]}$. The superior performance of TopDyG$_{[\text{Tide}]}$ and its ideal variant TopDyG$_{[\text{Tide-I}]}$ over TopDyG (which directly uses global degree distributions) further validates the efficacy of the time-varying intervened inference mechanism.

5.5 Effect of debiasing methods (RQ4)

To answer **RQ4**, we conduct experiments with three variants of each of the two backbones: (1) no debiasing, (2) mutual information-based (Mi) debiasing [37], and (3) our Tide framework.

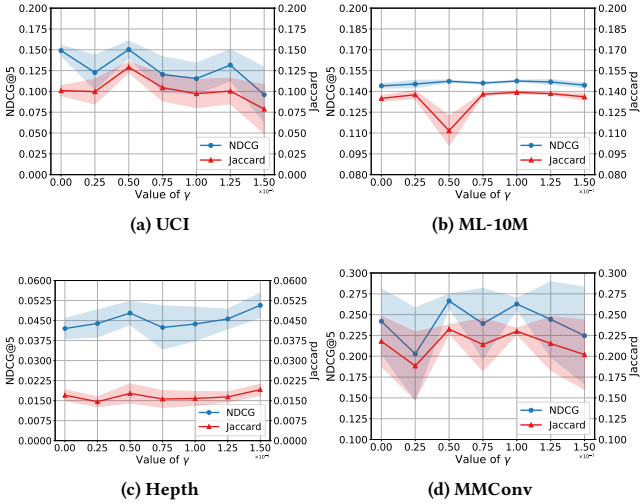
As shown in Fig. 5, distinct patterns represent different debiasing methods, while different colors indicate different backbones. Since the vanilla TopDyG contains the Mi debiasing module, the variant TopDyG$_{[\text{w/o Mi}]}$ refers to the no debiasing variant of TopDyG. We can observe that for each backbone, the Tide-enhanced variants are always better than the others. Although the Mi debiasing method proves effective when using TopDyG as the backbone architecture, SimpleDyG$_{[\text{Mi}]}$ underperforms the corresponding vanilla model. The observed performance degradation on the inductive dataset Hepth specifically demonstrates that debiasing approaches relying on global degree distribution priors exhibit instability in inductive scenarios, due to the inability to estimate the future degree distribution during the inference stage.

5.6 Analysis on hyper-parameter γ (RQ5)

To answer **RQ5**, we examine the performance of TopDyG$_{[\text{Tide}]}$ on the UCI, ML-10M and Hepth datasets by varying γ from 0 to 0.15 with a step size of 0.025. For the MMConv dataset, we set γ to range from 0 to 0.0015 with an increment step of 2.5×10^{-4} o. The results are shown in Fig. 6.

First, the value of γ on MMConv dataset is much smaller than those for the other three non-bipartite graph datasets. Combining the observations in Sec. 5.3 with the bipartiteness of the MMConv dataset, we can explain this phenomenon by the fact that although it is bipartite, its link prediction still selects nodes from the complete node set. When nodes from different categories serve as subgraph centers, their corresponding candidate nodes sets are different in the ideal conditions, making predictions more dependent on the sequence-learned time-invariant features. Consequently, large γ values cause time-varying features to dominate over time-invariant features, resulting in the predictions being primarily guided by the estimated future degree distributions.

Next we zoom in on the trend of the curves. The metrics, i.e., NDCG and Jaccard reach the peaks when γ is between 0.05 and 0.125, showing an overall upward trend from 0 to 0.05, and a general



(a) UCI (b) ML-10M

(c) Hepth (d) MMConv

Figure 6: Performance of TopDyG$_{[\text{Tide}]}$ with the parameter γ in Eq. (9) changing. The dark line represents the mean, and the light shaded area indicates the standard deviation.

downward trend from 0.125 to 0.15 on the UCI and ML-10M dataset. For the Hepth dataset, both NDCG and Jaccard metrics exhibit an overall positive correlation with γ values. Building upon our discussion of time-varying intervened inference mechanisms in Sec. 5.3, we interpret this phenomenon as further evidence of time-varying features' critical role in dynamic link prediction under inductive settings. For the MMConv dataset, NDCG and Jaccard scores reach the peaks when γ is between 0.5×10^{-3} and 1.0×10^{-3}, showing an overall upward trend from 0 to 0.5×10^{-3}, and a general downward trend from 1.0×10^{-3} to 1.5×10^{-3}. These observations indicate that the optimal γ value is inherently related to graph characteristics.

5.7 Case study

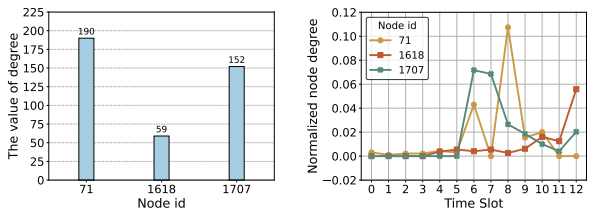
To clearly demonstrate the effectiveness of the Tide framework, we select two representative samples from the UCI dataset for detailed analysis. Fig. 7(a) shows the details. Specifically, the key "Node id" denotes the central node of the 1-hop subgraph, and "Input" denotes the serialized subgraph incorporating temporal identifiers, i.e., special tokens representing the start of time slots. While the baseline TopDyG model incorrectly predicts node 71 for both central nodes 447 and 494, TopDyG$_{[\text{Tide}]}$ generates more accurate predictions for each respective central node than the baseline. We attribute these observed phenomena to two primary factors:

- **TopDyG's errors predominantly stem from the influence caused by global node degree distributions.** Specifically, as shown in Fig. 7(b), throughout the entire time span, node 71 maintains a higher degree than nodes 1618 and 1707. Consequently, TopDyG fails to capture the temporal evolution of degree distributions since it is influenced by the global domination of the high-degree nodes, such as node 71.
- **Tide's effectiveness derives from its ability to capture time-varying characteristics in dynamic graph, i.e., the time-varying degree distributions.** Specifically, as shown in Fig. 7(c),

```
Node id: "447"
Input: "<|endoftext|> <|history|> 447 <|time0|> 448 456 438 465 <|time1|> 448 198
89 492 130 380 164 284 518 242 529 389 456 131 396 2 345 544 292 387 193 298 440
313 608 35 25 645 272 542 385 629 688 445 183 <|time2|> 385 193 183 <|time3|> 1239
<|time4|> 1306 <|time5|> <|time6|> <|time7|> 1618 486 193 <|endofhistory|>"
Ground Truth: "1125" ,"1618";
Predicted by TopDyG: "71"
Predicted by TopDyG[Tide]: "1618"

Node id: "494"
Input: "<|endoftext|> <|history|> 494 <|time0|> 253 <|time1|> 17 89 58 <|time2|>
<|time3|> 58 389 1310 89 1210 17 731 608 88 596 629 518 244 115 262 296 183 342
<|time4|> <|time5|> 596 <|endofhistory|>"
Ground Truth: "1310", "1707", "1556", "1750";
Predicted by TopDyG: "71"
Predicted by TopDyG[Tide]: "1707"
```

(a) Two instances extracted from UCI dataset



(b) The total degree of each node. (c) Normalized degree across time.

Figure 7: Performance comparison to examine the impact of intervention techniques.

beginning at the 8-th time slot, degree of node 71 exhibits a pronounced decline, while degrees of nodes 1618 and 1707 demonstrate opposing upward trends. Intuitively, the future prediction probability for node 71 should decrease while those for nodes 1618 and 1707 should correspondingly increase. The Tide framework's time-invariant deconfounded learning effectively diminishes the influence of global degree distributions, whereas its time-varying intervened inference mechanism successfully captures the evolving patterns of time-sensitive degree distributions.

6 Conclusion and future work

In this paper, we propose Time-wise causal debiasing framework (**Tide**) for generative dynamic link prediction, which consists of time-invariant deconfounded learning and time-varying intervened inference. Specifically, time-invariant deconfounded learning aims to help the model capture the time-invariant features in the graph by decoupling the model prediction score from the node degree distribution. Additionally, time-varying intervened inference regards the time-varying degree distributions as an intrinsic feature of dynamic graphs to guide the models to generate accurate results. Comprehensive experiments on four public datasets of dynamic graphs validate the effectiveness of Tide framework, with consistent and steady performance improvements.

In future research, we intend to generalize the causal intervention approach to applications integrating large language models with dynamic graph networks, with the objective of advancing LLMs' capability to model temporal graph dynamics.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under No. 62372458.

GenAI Usage Disclosure

- We only used GenAI for grammar checking and polishing.
- We did not use GenAI to generate any content, including tables, graphs, images, code, and texts.

References

[1] Barabasi and Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–12.

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *ICLR (Poster)*.

[3] Weilin Cong, Si Zhang, Jian Kang, and et al. 2023. Do We Really Need Complicated Model Architectures For Temporal Networks?. In *ICLR*. OpenReview.net.

[4] Dominik Maria Endres and Johannes E. Schindelin. 2003. A new metric for probability distributions. *IEEE Trans. Inf. Theory* 49, 7 (2003), 1858–1860.

[5] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.

[6] Keke Huang, Ruize Gao, Bogdan Cautis, and Xiaokui Xiao. 2024. Scalable Continuous-time Diffusion Framework for Network Inference and Influence Estimation. In *WWW*. ACM, 2660–2671.

[7] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, and et al. 2020. Representation Learning for Dynamic Graphs: A Survey. *J. Mach. Learn. Res.* 21 (2020), 70:1–70:73.

[8] Shima Khoshraftar and Aijun An. 2024. A Survey on Graph Representation Learning Methods. *ACM Trans. Intell. Syst. Technol.* 15, 1 (2024), 19:1–19:55.

[9] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *KDD*. ACM, 1269–1278.

[10] Seunghan Lee, Geonwoo Ko, Hyun-Je Song, and Jinhong Jung. 2024. MuLe: Multi-Grained Graph Learning for Multi-Behavior Recommendation. In *CIKM*. ACM, 1163–1173.

[11] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*. ACM, 177–187.

[12] Yilin Li, Jiaqi Zhu, Congcong Zhang, and et al. 2023. THGNN: An Embedding-based Model for Anomaly Detection in Dynamic Heterogeneous Social Networks. In *CIKM*. ACM, 1368–1378.

[13] Lizi Liao, Le Hong Long, Zheng Zhang, and et al. 2021. MMConv: An Environment for Multimodal Conversational Search across Multiple Domains. In *SIGIR*. ACM, 675–684.

[14] Yulei Niu, Kaihua Tang, Hanwang Zhang, and et al. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *CVPR*. Computer Vision Foundation / IEEE, 12700–12710.

[15] Zhiqiang Pan, Chen Gao, Fei Cai, and et al. 2025. On the Cross-Graph Transferability of Dynamic Link Prediction. In *WWW*. ACM, 4101–4110.

[16] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. 2009. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *J. Assoc. Inf. Sci. Technol.* 60, 5 (2009), 911–932.

[17] Aldo Pareja, Giacomo Domeniconi, Jie Chen, and et al. 2020. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In *AAAI*. AAAI Press, 5363–5370.

[18] Judea Pearl. 2009. *Causality*. Cambridge University Press.

[19] Alec Radford, Jeff Wu, Rewon Child, and et al. 2019. Language Models are Unsupervised Multitask Learners. (2019).

[20] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, and et al. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. In *ICML 2020 Workshop on Graph Representation Learning*.

[21] Aravind Sankar, Yanhong Wu, Liang Gou, and et al. 2020. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In *WSDM*. ACM, 519–527.

[22] Chengyu Song, Linru Ma, Jianming Zheng, Jinzhi Liao, Hongyu Kuang, and Lin Yang. 2024. Audit-LLM: Multi-Agent Collaboration for Log-based Insider Threat Detection. *CoRR* abs/2408.08902 (2024).

[23] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. DyRep: Learning Representations over Dynamic Graphs. In *ICLR (Poster)*. OpenReview.net.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, and et al. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[25] Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. 2020. Streaming Graph Neural Networks via Continual Learning. In *CIKM*. ACM, 1515–1524.

[26] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, and et al. 2021. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. In *ICLR*. OpenReview.net.

[27] Zhihao Wen and Yuan Fang. 2022. TREND: TempoRal Event and Node Dynamics for Graph Representation Learning. In *WWW*. ACM, 1159–1169.

[28] Yuxia Wu, Yuan Fang, and Lizi Liao. 2024. On the Feasibility of Simple Transformer for Dynamic Graph Modeling. In *WWW*. ACM, 870–880.

[29] Da Xu, Chuanwei Ruan, Evren Körpeoglu, and et al. 2020. Inductive representation learning on temporal graphs. In *ICLR*. OpenReview.net.

[30] Jin Xu, Xiaojiang Liu, Jianhao Yan, and et al. 2022. Learning to Break the Loop: Analyzing and Mitigating Repetitions for Neural Text Generation. In *NeurIPS*.

[31] Pengwei Yan, Yangyang Kang, Zhuoren Jiang, and et al. 2024. Modeling Scholarly Collaboration and Temporal Dynamics in Citation Networks for Impact Prediction. In *SIGIR*. ACM, 2522–2526.

[32] Chenxiao Yang, Qitian Wu, Qingsong Wen, and et al. 2022. Towards Out-of-Distribution Sequential Event Prediction: A Causal Treatment. In *NeurIPS*.

[33] Yuhao Yang, Lianghao Xia, Da Luo, and et al. 2024. GraphPro: Graph Pre-training and Prompt Learning for Recommendation. In *WWW*. ACM, 3690–3699.

[34] Hyunsik Yoo, Ruizhong Qiu, Charlie Xu, and et al. 2025. Generalizable Recommender System During Temporal Popularity Distribution Shifts. In *KDD (1)*. ACM, 1833–1843.

[35] Jiaxuan You, Yichen Wang, Aditya Pal, and et al. 2019. Hierarchical Temporal Convolutional Networks for Dynamic Recommender Systems. In *WWW*. ACM, 2236–2246.

[36] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *NeurIPS*. 5165–5175.

[37] Xin Zhang, Fei Cai, Jianming Zheng, Zhiqiang Pan, and et al. 2025. Triangle Matters! TopDyG: Topology-aware Transformer for Link Prediction on Dynamic Graphs. In *WWW*. ACM, 3607–3617.

[38] Yang Zhang, Fuli Feng, Xiangnan He, and et al. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*. ACM, 11–20.

[39] Zeyang Zhang, Xin Wang, Ziwei Zhang, and et al. 2024. LLM4DyG: Can Large Language Models Solve Spatial-Temporal Problems on Dynamic Graphs?. In *KDD*. ACM, 4350–4361.

[40] Zeyang Zhang, Ziwei Zhang, Xin Wang, and et al. 2023. Dynamic Heterogeneous Graph Attention Neural Architecture Search. In *AAAI*. AAAI Press, 11307–11315.

[41] Chu Zhao, Enneng Yang, Yuliang Liang, and et al. 2025. Graph Representation Learning via Causal Diffusion for Out-of-Distribution Recommendation. In *WWW*. ACM, 334–346.

[42] Jianming Zheng, Fei Cai, Yanxiang Ling, and Honghui Chen. 2020. Heterogeneous Graph Neural Networks to Predict What Happen Next. In *COLING*. International Committee on Computational Linguistics, 328–338.

[43] Jianming Zheng, Fei Cai, Jun Liu, Yanxiang Ling, and Honghui Chen. 2024. Multistructure Contrastive Learning for Pretraining Event Representation. *IEEE Trans. Neural Networks Learn. Syst.* 35, 1 (2024), 842–854.

[44] Xiaoshi Zhong and Huizhi Liang. 2024. On the Scale-Free Property of Citation Networks: An Empirical Study. In *WWW (Companion Volume)*. ACM, 541–544.