

Invariant Graph Representations for Continuous-Time Dynamic Graphs Under Distribution Shifts

Lanting Fang¹, Yulian Yang², Yawei Zhang¹, Shanshan Feng^{3*}, Kaiyu Feng¹ and Hanning Yuan¹

¹Beijing Institute of Technology

²Southeast University

³Wuhan University

ltfang@bit.edu.cn, 406674524@qq.com, z15334025555@163.com, {fengky, yhn6}@bit.edu.cn

Abstract

Continuous-Time Dynamic Graphs (CTDGs) enable fine-grained modeling of evolving relational systems. However, most existing CTDG representation learning methods are tailored to in-distribution settings and exhibit limited robustness under out-of-distribution (OOD) shifts. Although recent causal approaches learn invariant representations via interventions, they are primarily designed for static or discrete-time graphs and become computationally prohibitive for CTDGs due to the combinatorial explosion of structural and temporal variations. To address these challenges, we propose CIR, a framework grounded in a novel structural causal model termed the ICCM. To avoid exhaustive interventions, we leverage the Normalized Weighted Geometric Mean (NWGM) to efficiently approximate interventional predictions. We further instantiate ICCM within a practical deep learning architecture that jointly captures invariant structural and temporal patterns through dedicated subgraph extractors, and maintains an environment memory bank to model distribution shifts across evolving contexts. Extensive experiments demonstrate that CIR consistently outperforms existing methods under diverse OOD scenarios.

Therefore, this paper focuses on invariant representation learning for CTDGs under OOD scenarios. While techniques such as subgraph extraction [Deng and Shen, 2024; Li *et al.*, 2024a] and attention mechanisms [Ranjan *et al.*, 2020] can partially alleviate distribution shifts, they remain vulnerable to exploiting spurious correlations [Chang *et al.*, 2020]. For example, in the link prediction setting illustrated in Figure 1, a triadic closure pattern (highlighted in red) constitutes the invariant subgraph underlying the formation of the target link (u, v) . However, because a bridging link (x, u) (highlighted in blue), also appears before (u, v) in most training graphs, the model may incorrectly treat (x, u) as the invariant subgraph for (u, v) . As a result, at test time, the model may falsely predict the existence of (u, v) based solely on (x, u) , ignoring the absent of the triadic closure pattern.

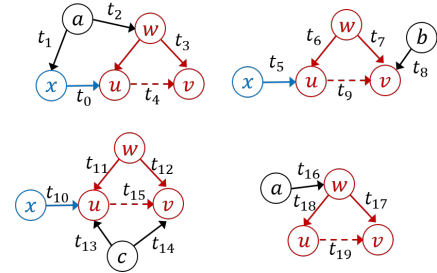


Figure 1: Examples of training data.

1 Introduction

Many real-world systems exhibit continuously evolving relational structures, which are naturally modeled as dynamic graphs [Zhang *et al.*, 2025]. Existing dynamic graph models can be broadly categorized into Discrete-Time Dynamic Graphs (DTDGs) [Ahmed and Chen, 2016; Ma *et al.*, 2019], which represent evolution as a sequence of temporal snapshots, and Continuous-Time Dynamic Graphs (CTDGs) [Peng *et al.*, 2025; Yuan *et al.*, 2025a; Li *et al.*, 2017; De Winter *et al.*, 2018], which capture fine-grained structural changes via continuous node and edge updates. Although existing methods have achieved significant success in modeling temporal dynamics, they predominantly overlook robustness under out-of-distribution (OOD) scenarios.

Recent structural causal model (SCM) based methods [Li *et al.*, 2025a; Sui *et al.*, 2022; Wang *et al.*, 2025; Fan *et al.*, 2022] have been proposed to mitigate spurious correlations by identifying and intervening on invariant subgraphs, either across multiple environments or through sampling-based interventions. However, these approaches implicitly assume a fixed or bounded graph structure, where the set of nodes and candidate edges remains stable over time, as in static graphs or DTDGs. Under this assumption, causal invariance can be meaningfully defined and estimated over a shared structural support. In contrast, CTDGs allow nodes and edges to be added or removed at arbitrary time points, resulting in a non-stationary and exponentially expanding space of graph topologies. This fundamentally challenges existing SCM-based intervention mechanisms.

*Shanshan Feng is the corresponding author.

In this paper, we propose *Continuous-time Invariant Representation (CIR)*, a novel method that captures invariant representation and mitigates OOD effects from a new SCM perspective. To effectively address OOD generalization in CTDGs, CIR is designed to satisfy three essential properties: (1) robust mitigation of OOD distribution shifts, which is fundamental for generalization across unseen environments; (2) simultaneous capture of invariant subgraphs across both structural and temporal dimensions, which is crucial in CTDGs where graph topology and event timing evolve continuously; and (3) computationally efficient causal interventions, necessitated by the exponentially large space of possible structural-temporal subgraphs.

CIR is theoretically grounded in a structural causal model termed the Independent and Confounded Causal Model (ICCM), which explicitly models both in-distribution and OOD environments. Directly intervening on all possible structural and temporal subgraphs is computationally infeasible. To address this challenge, we leverage the Normalized Weighted Geometric Mean (NWGM) [Xu *et al.*, 2015] to efficiently approximate the expected interventional outcome, substantially reducing computational cost. We instantiate ICCM within a practical deep learning framework. The architecture first employs dual subgraph extractors to capture causal structural and temporal patterns. An environment memory bank is then maintained to learn environment-specific representations, which are used to condition the prediction process. To approximate the expected outcome of causal interventions defined in ICCM, we aggregate environment-conditioned predictions using the NWGM. Finally, the causal structural and temporal patterns are encoded into invariant representations, and the model performs predictions based on the learned invariant representations together with the environment representations. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to investigate invariant representation learning problem for CTDGs under OOD shifts.
- We introduce ICCM, a novel structural causal model that characterizes invariant causal subgraphs in CTDGs, and develop an efficient intervention strategy based on the NWGM to approximate interventional predictions, avoiding exhaustive enumeration of structural and temporal subgraphs.
- We design a framework named CIR that instantiate ICCM, enabling the joint learning of invariant structural and temporal representations along with environment-specific features.

2 Related Work

In this section, we review two areas closely related to our study: dynamic graph neural networks and invariant graph representation.

Dynamic Graph Neural Networks. Dynamic graph neural networks encompass two primary classifications: Discrete-Time Dynamic Graphs (DTDGs) [Ma *et al.*, 2019; Qin

et al., 2023] and Continuous-Time Dynamic Graphs (CTDGs) [Li *et al.*, 2017; De Winter *et al.*, 2018; Li *et al.*, 2023]. DTDGs comprise a sequence of static graph snapshots captured at regular time intervals [Sankar *et al.*, 2020; Yu *et al.*, 2018; Mahdavi *et al.*, 2018]. CTDGs capture the evolution of graphs by considering modifications on the graph that occur continuously rather than discretely at pre-defined time steps [Xu *et al.*, 2020; Trivedi *et al.*, 2019; Feng *et al.*, 2023]. However, these GNNs focus primarily on modeling graph dynamics while overlooking robustness to OOD scenarios.

Invariant Graph Representation. Invariant learning enhances the robustness of GNNs by identifying stable, environment-invariant patterns. Existing approaches fall broadly into two categories: disentangled learning and causal learning. Disentangled learning [Yuan *et al.*, 2025b; Ning *et al.*, 2025; Li *et al.*, 2024b] aims to learn representations in which latent factors are separated and interpretable, capturing distinct variations in the data. Such representations have been shown to improve robustness to adversarial attacks and enhance generalization. However, these methods are limited in their ability to identify causal variables that remain invariant in the presence of shortcuts or spurious correlations. Causal inference methods [Li *et al.*, 2025b; Wu *et al.*, 2022] focus on unveiling and comprehending the true causal variables that drive observed phenomena. On real-world graphs, uncovering these causal variables becomes an act of explanation, revealing the “why” behind intricate relationships. However, most of these methods are largely restricted to static settings. To extend causal inference into dynamic scenarios, DIDA [Zhang *et al.*, 2022] has been proposed as an invariant rational discovery framework for DTDGs. However, DIDA requires constructing an intervention set for each snapshot, making it computationally intensive when applied to CTDGs. EAGLE [Yuan *et al.*, 2023] employs a VAE-based model as an environment sample generation network. However, when the number of environments is large and the available sample size is limited, the generated samples may not adequately represent the underlying environments, thereby impairing generalization performance. In this work, we focus on the more realistic and challenging setting of CTDGs, where interactions evolve continuously rather than at discrete time intervals. Existing causal inference methods are generally inapplicable to this setting because they predominantly rely on snapshot-based assumptions, which fail to capture the fine-grained temporal dynamics inherent in CTDGs.

3 Problem Formulation

Definition 1 (CTDG) A Continuous Time Dynamic Graph (CTDG) $G = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ consists of a set of vertices $\mathcal{V}$, a set of edges $\mathcal{E}$, and a continuous time domain $\mathcal{T}$. The graph evolves over $t \in \mathcal{T}$ through edge additions, removals, or changing characteristics. We represent G as a sequence of time-stamped edges $G = \langle e_{ij}(t_k) \rangle$, where $e_{ij}(t_k)$ denotes an interaction between node v_i and v_j at time t_k with feature vector $x_{ij}^e(t_k)$. Each node v_i has a feature vector x_i^n .

Given a CTDG $G = (\mathcal{V}, \mathcal{E})$, the task is to predict whether

an interaction $e_{ij}(t)$ will occur at a future time t , based on the historical sequence of nodes, edges, and their features. To evaluate OOD robustness, the model is trained on one environment and tested on CTGDs from other environments, where structural and temporal distributions may differ.

4 Independent and Confounded Causal Model

In this section, we introduce a novel structural causal model framework, termed Independent and Confounded Causal Model (ICCM).

In dynamic graphs, the existence of an edge can be influenced by both the structural topology (e.g., triadic closure) and temporal dynamics (e.g., temporal burst) [Cong *et al.*, 2023]. These mechanisms may act independently or become entangled through latent confounders, and their influence can vary across environments. Existing causal formulations typically conflate structural and temporal effects, limiting their ability to disentangle spurious correlations under distribution shifts.

To explicitly capture both independent causal effects and environment-dependent confounding, ICCM models structural and temporal causal processes as distinct but potentially confounded components. We formalize ICCM by inspecting the causalities among nine variables: the structural causal subgraph $C_s \subseteq G$, the temporal causal subgraph $C_t \subseteq G$, the hidden representations H^T and H^S , the structural confounder $\mathcal{D}_s$, the temporal confounder $\mathcal{D}_t$, the edge existence probability in the in-distribution environment Y^I , the edge existence probability in OOD environments Y^O , and the final prediction result Y . Assumption 4.1 summarizes the core principles underlying ICCM:

Assumption 4.1 (ICCM)

$$\begin{aligned} H^S &= f_s(C_s), \quad H^T = f_t(C_t), \quad Y^I = f_o^I(H^S, H^T) \\ Y^C &= f_o^C(H^S, H^T, \mathcal{D}_s, \mathcal{D}_t), \quad \text{s.t.} \quad C_s \perp \mathcal{D}_s, \quad C_t \perp \mathcal{D}_t, \\ Y &= f_o(Y^I, Y^C). \end{aligned}$$

In this assumption, C_s and C_t are causal structural and temporal subgraph. These subgraphs are encoded into latent representations H^S and H^T via $f_s(\cdot)$ and $f_t(\cdot)$. The in-distribution prediction Y^I is determined by the outcome function $f_o^I(H^S, H^T)$:

$$Y^I = f_o^I(H^S, H^T) = \sigma(W^I(W_1^I H^S + W_2^I H^T)). \quad (1)$$

In OOD environments, however, structural and temporal confounders introduce spurious correlations that degrade model generalization. Specifically, structural confounders establish backdoor paths $C_s \leftarrow C_b^s \rightarrow Y$ and $C_s \leftarrow U_s \rightarrow Y$, where $C_b^s = G \setminus C_s$, and U_s represent unobserved variables. These paths induce spurious correlations between C_s and Y^C . Temporal confounders exert a similar influence. To mitigate these spurious correlations, we employ the *do-operation* introduced by Pearl [Pearl *et al.*, 2016]. The *do-operation* allows us to block backdoor paths, thereby eliminating spurious correlations and isolating the invariant causal effect. Using

the *do-operation*, the prediction Y^C can be formulated as:

$$\begin{aligned} Y^C &= f_o^C(H^S, H^T, \mathcal{D}_s, \mathcal{D}_t) \quad \text{s.t.} \quad C_s \perp \mathcal{D}_s, \quad C_t \perp \mathcal{D}_t, \\ &= \mathbb{E}_{u \in \mathcal{D}_s, v \in \mathcal{D}_t} [f_o^C(H^S, H^T, u, v)] \end{aligned} \quad (2)$$

where $\mathcal{D}_s = \{C_b^s, U_s\}$ and $\mathcal{D}_t = \{C_b^t, U_t\}$, respectively, f_o^C is output network:

$$\begin{aligned} f_o^C(H^S, H^T, u, v) &= \sigma(W_1^c f_y^s(H^S) + W_2^c f_y^t(H^T) \\ &\quad + W_3^c f_y^u(u) + W_4^c f_y^v(v)). \end{aligned} \quad (3)$$

where W^* denotes the model parameters, $f_y^*(\cdot)$ denotes linear network, σ denotes the activation function. However, we observe Equation 2 is computationally intractable. It necessitates enumerating all possible pairs of invariant causal subgraphs across the space of structural confounders $\mathcal{D}_s$ and temporal confounders $\mathcal{D}_t$. In the context of large-scale CTGDs, this combinatorial explosion of potential subgraph configurations leads to prohibitive computational costs, making exhaustive intervention infeasible.

To address this issue, we propose an efficient intervention process by leveraging the NWGM approximation [Xu *et al.*, 2015]. The NWGM provides a computationally efficient way to estimate the expectation of a softmax or sigmoid output by moving the expectation inside the non-linearity:

$$\mathbb{E}_{d \sim \mathcal{D}} [\sigma(f(x))] \approx \sigma(\mathbb{E}_{d \sim \mathcal{D}} [f(x)]). \quad (4)$$

By applying the NWGM approximation, the interventional prediction in Equation 2 can be reformulated as:

$$\begin{aligned} Y^C &\approx \sigma(\mathbb{E}_{u \in \mathcal{D}_s, v \in \mathcal{D}_t} [W_1^c f_y^s(H^S) + W_2^c f_y^t(H^T) \\ &\quad + W_3^c f_y^u(u) + W_4^c f_y^v(v)]), \\ &= \sigma(W_1^c f_y^s(H^S) + W_2^c f_y^t(H^T) + W_3^c \mathbb{E}_{u \sim \mathcal{D}_s} [f_y^u(u)] \\ &\quad + W_4^c \mathbb{E}_{v \sim \mathcal{D}_t} [f_y^v(v)]). \end{aligned} \quad (5)$$

Note that $\mathcal{D}_s$ and $\mathcal{D}_t$ denote abstract latent confounders, which may themselves correspond to complex subgraph patterns. While the confounder space can be large, Equation 5 does not require explicit enumeration of all possible confounders. Instead, it only requires expectation over a representative set of confounding modes, which can be approximated via confounder prototypes (Explained in Section 5.2).

5 Neural Instantiation of ICCM

In this section, we describe how the proposed ICCM is instantiated within a neural architecture by mapping latent causal variables to learnable components. The complete architecture is illustrated in Figure 2.

While the theoretical ICCM assumes access to causal subgraphs C_s and C_t , these variables are typically latent and unobserved in real-world scenarios. To address this, we employ causal subgraph extractors to infer structural and temporal subgraphs directly from the input data. The extractors identify invariant structural and temporal patterns and produce hidden representations H^S and H^T .

For the in-distribution perspective, these representations are passed to the outcome function f_o^I to predict the label

y^I . For the out-of-distribution (OOD) perspective, as described in Equation 5, the interventional prediction requires the expected representations of structural and temporal confounders: $\mathbb{E}_{u \sim \mathcal{D}_S}[f_y^u(u)]$ and $\mathbb{E}_{v \sim \mathcal{D}_T}[f_y^v(v)]$.

To estimate these expectations without exhaustive enumeration, we maintain structural and temporal memory banks that approximate the latent confounder distributions $\mathcal{D}_S$ and $\mathcal{D}_T$. A multi-head attention-based fusion module then aggregates the memory entries to compute the expected environmental context, which approximates the expectation over latent confounder effects. Finally, the invariant representations H^S and H^T are integrated with the computed expectations from the memory banks and fed into the outcome function f_o^C to produce the OOD prediction y^C . We term this framework Continuous Invariant Representation (CIR).

5.1 Extracting and Encoding Causal/Non-causal Subgraphs

Temporal extraction and encoding. Given a dynamic graph G and a target node pair (u, v) for prediction, our goal is to identify historical temporal interactions that causally contribute to the potential existence of an edge between u and v . Unlike heuristic time-window approaches, we aim to extract temporally relevant evidence that is both **time-aware** and **pair-specific**, aligning with the causal assumptions in ICCM. We first extract the temporal edges associated with nodes u and v denoted by S_G^u and S_G^v , respectively. Each edge $e_{ui}(t_k) \in S_G^u$ is encoded using a continuous-time representation following [Cong *et al.*, 2023]: $\cos((t_0 - t_k)\omega)$, where t_0 is the prediction timestamp, $\omega = \{\alpha^{-(i-1)/\beta}\}_{i=1}^d$ defines a set of multi-scale temporal frequencies, with α and β representing hyperparameters. This encoding is combined with its corresponding edge features as $[\cos((t_0 - t_k)\omega) \parallel \mathbf{x}_{ui}^e(t_k)]$.

A 1-layer MLP-mixer [Tolstikhin *et al.*, 2021] is employed to encode the edge features,

$$F_u = \text{MLPmixer}(\text{stack}(S_G^u)), \quad (6)$$

where $\text{stack}(S_G^u)$ indicates the stacking of the edge features in sequence S_G^u . The MLP-Mixer enables global mixing across temporal edges, allowing the representation of higher-order temporal dependencies that are crucial for causal reasoning.

Intuitively, a historical interaction of v is considered temporally causal for predicting (u, v) only if it helps explain u 's behavior, and vice versa. Therefore, we introduce a cross-node attention mechanism. The subgraph is generated by:

$$\begin{aligned} q_u &= W_1^m \text{Mean}(F_u), & K_v &= W_2^m(F_v) \\ q_v &= W_1^m \text{Mean}(F_v), & K_u &= W_2^m(F_u) \\ M_v^e &= \text{Softmax}\left(\frac{q_u^T K_v}{\sqrt{d}}\right), & M_u^e &= \text{Softmax}\left(\frac{q_v^T K_u}{\sqrt{d}}\right). \end{aligned} \quad (7)$$

Here, d denotes a specific hyperparameter, and $M_*^e[k]$ represents the importance score assigned to the k -th edge within S_* . This attention mechanism filters temporal evidence based on mutual relevance, rather than local frequency or recency alone. Consequently, the highest top- k scores in M_v^e and M_u^e are selected to construct the temporal causal subgraphs C_t ,

while the remaining edges form the temporal non-causal subgraphs $\hat{B}_t$. Finally, the temporal representations of the causal and non-causal graphs are encoded by:

$$\begin{aligned} h_u^t &= \text{MLPmixer}(\text{stack}(S_{C_t}^u)), & h_v^t &= \text{MLPmixer}(\text{stack}(S_{C_t}^v)), \\ x_u^t &= \text{MLPmixer}(\text{stack}(S_{\hat{B}_t}^u)), & x_v^t &= \text{MLPmixer}(\text{stack}(S_{\hat{B}_t}^v)), \end{aligned} \quad (8)$$

The final representations are obtained by concatenation:

$$H^T = [h_u^t \parallel h_v^t], \quad X^T = [x_u^t \parallel x_v^t]. \quad (9)$$

Causal temporal evidence is disentangled from confounding effects, supporting intervention-based learning.

Structural extraction and encoding. Our objective is to identify structural neighbors that causally support the interaction between u and v , while filtering out background nodes that may introduce spurious correlations. We first encode each node based on its n -hop neighborhood within a recent time window $[t_0 - T, t_0]$: $z_u = x_u^n + \text{Mean}(\{x_v^n \mid v \in \mathcal{N}^n(u; t_0 - T, t_0)\})$. Here, $\mathcal{N}^n(u; t_0 - T, t_0)$ denotes the n -hop neighbors of node u with valid temporal edges, where T controls the structural temporal scope. The node mask matrices are computed through the equations:

$$M_v^n = \text{Softmax}\left(\frac{z_u^T Z_v}{\sqrt{d}}\right), \quad M_u^n = \text{Softmax}\left(\frac{z_v^T Z_u}{\sqrt{d}}\right). \quad (10)$$

Here, Z_u and Z_v is the stack of the encoded node features of all nodes in $\mathcal{N}^n(u; t_0 - T, t_0)$ and $\mathcal{N}^n(v; t_0 - T, t_0)$, respectively. The nodes with the highest top- k scores in M_u^n and M_v^n are chosen to form the structural causal subgraph, while the remaining edges form the temporal non-causal subgraphs $\hat{B}_s$. The final structural representation H^S is computed by:

$$\begin{aligned} h_u^s &= x_u^n + \text{Mean}(\{x_i^n \mid i \in \mathcal{N}_S^n(u)\}), \\ h_v^s &= x_v^n + \text{Mean}(\{x_i^n \mid i \in \mathcal{N}_S^n(v)\}), \\ H^S &= [h_u^s \parallel h_v^s], \end{aligned} \quad (11)$$

where x_v^n represents the node feature of v , $\mathcal{N}_S^n(u)$ represents the n -hop neighbors of u in C_s . The final structural representation X^S for the non-causal subgraph $\hat{B}_s$ is obtained using the same encoding method applied to C_s .

5.2 Memory Banks

The memory bank is designed to capture all confounders, including both non-causal subgraphs and unobserved factors. Since unobserved factors are not directly observable, we approximate them using the representations of observable non-causal subgraphs, based on the insight that unobserved influences are often implicitly reflected in the structure of these subgraphs.

As the dynamic graph evolves, the number of non-causal subgraphs can grow exponentially, making it infeasible to consider each instance individually. To address this, each entry in the memory bank approximates a cluster of non-causal subgraphs, capturing the shared influence of similar confounders. Specifically, the memory bank consists of K prototype vectors stored in the $\hat{\mathcal{D}}_* \in \mathbb{R}^{K \times d}$, where K denotes

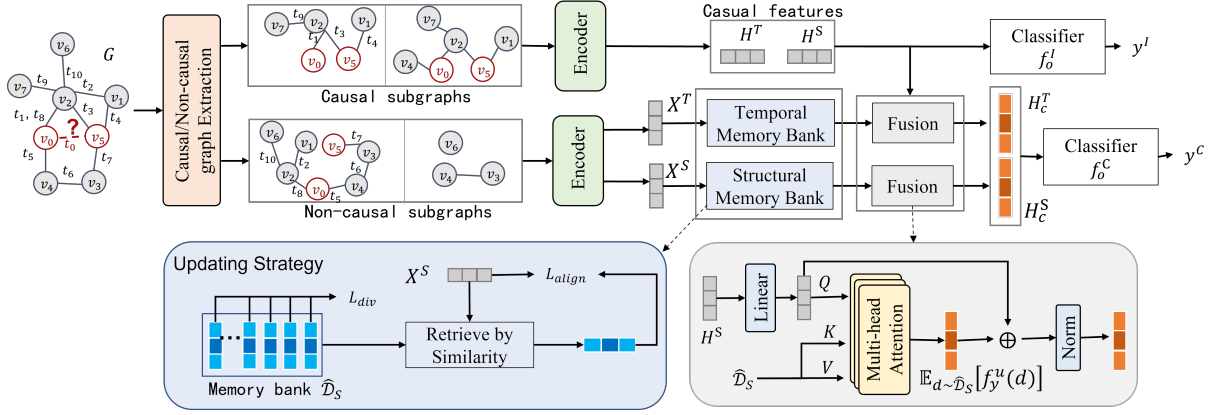


Figure 2: The deep learning implementation of CIR.

the number of prototypes and d represents the feature dimensionality. Each element of $\hat{\mathcal{D}}_*$ is initialized by sampling from a standard normal distribution, after which each prototype is normalized to ensure that they lie on a unit hypersphere. The memory bank is updated during training via gradient descent, guided by two self-supervised loss functions: the *alignment loss* and the *diversity loss*.

Given non-causal graphs X^T and X^S , the alignment loss is defined as:

$$\begin{aligned} z_T &= \text{linear}(X^T) \quad z_S = \text{linear}(X^S) \\ k_T &= \arg \max_k z_T \hat{\mathcal{D}}_T^\top[k] \quad k_S = \arg \max_k z_S \hat{\mathcal{D}}_S^\top[k] \\ \mathcal{L}_{\text{align}} &= 1 - (z_T \hat{\mathcal{D}}_T[k_T] + z_S \hat{\mathcal{D}}_S[k_S]), \end{aligned} \quad (12)$$

Here, z_T and z_S are encoded features of X^T and X^S , respectively. The term $z_* \hat{\mathcal{D}}_*^\top[k_*]$ (with $*$ representing either T or S) measures the similarity between the encoded feature z_* and the k_* -th prototype in $\hat{\mathcal{D}}_*$. Each sample is assigned to the prototype with the highest similarity. When z_* is well aligned with $\hat{\mathcal{D}}_*[k]$, the loss contribution for that sample is minimal.

To prevent redundancy among the prototypes, we impose a diversity loss:

$$\mathcal{L}_{\text{div}} = \frac{1}{K} \left(\sum_{i=1}^K \sum_{j=1}^K \hat{\mathcal{D}}_S[i] \cdot \hat{\mathcal{D}}_S[j] + \sum_{i=1}^K \sum_{j=1}^K \hat{\mathcal{D}}_T[i] \cdot \hat{\mathcal{D}}_T[j] \right). \quad (13)$$

Minimizing this loss encourages the prototypes to be as orthogonal as possible, ensuring that each captures a distinct cluster of non-causal subgraphs.

The overall regularization loss is the sum of the alignment and diversity losses:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{div}}. \quad (14)$$

This loss is backpropagated during training to update both the encoder parameters and the memory bank, thereby refining the prototypes to more accurately represent distinct clusters of non-causal subgraphs.

5.3 Fusion

As shown in Equations 5, the fusion module combines the causal graph representations (H^S and H^T) with the expected

contributions of confounders, as approximated by the memory banks ($\hat{\mathcal{D}}_S$ and $\hat{\mathcal{D}}_T$). This step implements the ICCM-inspired interventional reasoning, where the model accounts for potential spurious effects from non-causal subgraphs and unobserved factors. We approximate the expected effect of confounders as:

$$\begin{aligned} Q_S &= \text{Linear}(H^S) \quad K_S = \hat{\mathcal{D}}_S \quad V_S = \hat{\mathcal{D}}_S \\ \mathbb{E}_{u \sim \hat{\mathcal{D}}_S} [f_y^u(u)] &= \text{Multi-head}(Q_S, K_S, V_S), \\ Q_T &= \text{Linear}(H^T) \quad K_T = \hat{\mathcal{D}}_T \quad V_T = \hat{\mathcal{D}}_T \\ \mathbb{E}_{v \sim \hat{\mathcal{D}}_T} [f_y^v(v)] &= \text{Multi-head}(Q_T, K_T, V_T). \end{aligned} \quad (15)$$

The attention output is then fused with the original query using a residual connection, followed by dropout and layer normalization:

$$\begin{aligned} H_c^S &= \text{LayerNorm}(Q + \text{Dropout}(\mathbb{E}_{u \sim \hat{\mathcal{D}}_S} [f_y^u(u)])) \\ H_c^T &= \text{LayerNorm}(Q + \text{Dropout}(\mathbb{E}_{v \sim \hat{\mathcal{D}}_T} [f_y^v(v)])). \end{aligned} \quad (16)$$

The fused features H_c^S and H_c^T are subsequently processed by a feedforward network and output the final prediction y^C .

5.4 Optimization

We define the overall learning objective as:

$$\begin{aligned} \mathcal{R}_i &= \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} [y_e \log(y_e^I) + (1 - y_e) \log(1 - y_e^I)], \\ \mathcal{R}_c &= \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} [y_e \log(y_e^C) + (1 - y_e) \log(1 - y_e^C)], \end{aligned} \quad (17)$$

$$\mathcal{L} = \lambda_i \mathcal{R}_i(y^I, y) + \lambda_c \mathcal{R}_c(y^C, y) + \lambda_r \mathcal{L}_{\text{reg}},$$

where λ_i , λ_c , and λ_r are hyperparameters, $\mathcal{E}$ denotes the training dataset, which comprises pairs of positive and negative samples. Positive samples are derived from the original edge sets, while negative samples are generated by replacing the destination nodes with randomly sampled nodes from the vocabulary, ensuring a balanced ratio. The variable y_e indicates the ground-truth label for edge e , with $y_e = 1$ for positive samples and $y_e = 0$ for negative samples.

Model	MOOC			REDDIT			WIKI			UCI		
Ratio	0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6
TGN	90.39 $\pm$ 1.27	65.49 $\pm$ 0.81	61.99 $\pm$ 2.21	95.44 $\pm$ 1.09	87.70 $\pm$ 3.55	72.69 $\pm$ 0.31	68.56 $\pm$ 2.03	61.92 $\pm$ 2.63	64.97 $\pm$ 0.65	68.93 $\pm$ 1.41	68.43 $\pm$ 3.58	64.68 $\pm$ 2.47
TGAT	93.75 $\pm$ 0.49	85.25 $\pm$ 1.21	73.08 $\pm$ 1.49	97.32 $\pm$ 0.14	95.34 $\pm$ 0.28	89.21 $\pm$ 0.48	90.90 $\pm$ 0.78	74.72 $\pm$ 0.74	73.07 $\pm$ 2.74	77.66 $\pm$ 1.17	74.55 $\pm$ 0.46	72.59 $\pm$ 0.54
GraphMixer	98.06$\pm$0.27	88.11 $\pm$ 4.62	48.71 $\pm$ 3.02	98.41$\pm$0.33	78.85 $\pm$ 3.10	91.74 $\pm$ 1.32	95.90$\pm$1.15	73.61 $\pm$ 5.39	64.01 $\pm$ 7.36	88.67 $\pm$ 0.68	75.45 $\pm$ 2.93	65.35 $\pm$ 0.39
DIDA	72.78 $\pm$ 3.32	59.87 $\pm$ 1.30	52.00 $\pm$ 0.63	88.26 $\pm$ 0.47	64.06 $\pm$ 0.06	59.05 $\pm$ 0.25	77.40 $\pm$ 0.46	60.94 $\pm$ 1.10	49.47 $\pm$ 0.35	77.17 $\pm$ 1.08	76.09 $\pm$ 2.87	65.92 $\pm$ 0.41
Ada-DyGNN	83.20 $\pm$ 4.74	74.03 $\pm$ 1.21	71.22 $\pm$ 0.51	95.42 $\pm$ 0.17	94.29 $\pm$ 0.54	90.79 $\pm$ 0.55	91.54 $\pm$ 0.23	72.70 $\pm$ 0.61	67.57 $\pm$ 0.44	89.59 $\pm$ 0.50	74.26 $\pm$ 3.43	67.15 $\pm$ 1.58
EAGLE	83.41 $\pm$ 0.97	73.12 $\pm$ 0.36	64.46 $\pm$ 0.37	85.07 $\pm$ 0.86	81.71 $\pm$ 0.44	67.38 $\pm$ 0.35	81.51 $\pm$ 0.56	72.41 $\pm$ 0.31	60.69 $\pm$ 0.59	77.90 $\pm$ 0.11	72.19 $\pm$ 0.61	65.06 $\pm$ 0.55
CIR	97.72$\pm$0.13	90.18$\pm$2.17	77.46$\pm$4.07	98.91$\pm$0.06	98.18$\pm$0.32	93.82$\pm$1.25	93.26$\pm$0.71	80.37$\pm$2.02	76.04$\pm$4.58	91.84$\pm$0.44	76.84$\pm$0.97	74.88$\pm$0.65

Table 1: Comparison with SOTA graph link prediction models w.r.t. AUC (mean $\pm$ standard deviation). The best scores are highlighted in bold.

Model	MOOC			REDDIT			WIKI			UCI		
Ratio	0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6
TGN	91.51 $\pm$ 1.5	59.69 $\pm$ 0.8	33.75 $\pm$ 3.2	96.47 $\pm$ 0.6	86.68 $\pm$ 4.0	55.41 $\pm$ 0.5	76.18 $\pm$ 1.8	64.09 $\pm$ 3.8	50.30 $\pm$ 2.8	66.82 $\pm$ 2.2	64.34 $\pm$ 1.4	43.01 $\pm$ 3.2
TGAT	93.42 $\pm$ 0.3	79.92 $\pm$ 1.8	62.02 $\pm$ 0.6	97.60 $\pm$ 0.1	94.64 $\pm$ 0.4	81.44 $\pm$ 0.9	91.90 $\pm$ 1.5	73.06 $\pm$ 0.6	59.73 $\pm$ 0.7	65.54 $\pm$ 1.0	60.84 $\pm$ 0.5	52.95 $\pm$ 0.4
GraphMixer	98.48$\pm$0.2	87.71 $\pm$ 2.2	31.19 $\pm$ 4.9	98.87$\pm$0.3	82.84 $\pm$ 2.6	90.85 $\pm$ 2.1	97.13$\pm$0.5	75.68 $\pm$ 4.5	61.14 $\pm$ 11.7	86.99 $\pm$ 1.5	73.13$\pm$1.4	56.50 $\pm$ 0.6
DIDA	67.15 $\pm$ 0.6	53.86 $\pm$ 1.2	50.74 $\pm$ 0.4	81.94 $\pm$ 1.6	55.62 $\pm$ 0.2	53.97 $\pm$ 0.3	73.23 $\pm$ 0.5	56.37 $\pm$ 1.0	48.07 $\pm$ 0.5	69.66 $\pm$ 3.2	61.47 $\pm$ 0.8	55.38 $\pm$ 0.8
Ada-DyGNN	81.81 $\pm$ 5.0	73.44 $\pm$ 1.4	70.04 $\pm$ 1.2	94.26 $\pm$ 0.4	90.96 $\pm$ 0.4	90.51 $\pm$ 0.5	92.63 $\pm$ 0.4	70.83 $\pm$ 0.3	64.54 $\pm$ 0.3	88.20 $\pm$ 0.8	72.91 $\pm$ 1.8	62.10$\pm$0.8
EAGLE	72.10 $\pm$ 0.3	72.03 $\pm$ 0.6	63.03 $\pm$ 0.6	79.82 $\pm$ 0.3	77.47 $\pm$ 0.5	65.16 $\pm$ 0.5	76.01 $\pm$ 0.5	66.43 $\pm$ 1.0	58.14 $\pm$ 0.5	74.06 $\pm$ 0.1	62.36 $\pm$ 0.4	55.22 $\pm$ 0.4
CIR	98.14$\pm$0.5	88.24$\pm$0.3	64.53$\pm$2.6	98.95$\pm$0.1	98.15$\pm$0.3	93.65$\pm$2.0	94.96 $\pm$ 0.7	79.10$\pm$4.5	63.46$\pm$6.0	90.59$\pm$0.6	73.55$\pm$1.8	57.36 $\pm$ 0.4

Table 2: Comparison with SOTA graph link prediction models w.r.t. AP.

6 Experiments

We conduct extensive experiments on four benchmark continuous-time dynamic-graph datasets to answer the following research questions:

- **RQ1:** How well does CIR mitigate distribution shift?
- **RQ2:** How efficient is CIR compared to existing baselines?
- **RQ3:** What is the contribution of each component?

All the experiments are conducted on a computer with Intel(R) Core(TM)2 Duo CPU @2.40 GHz processor, 128 GB RAM, and Tesla T4.

6.1 Experimental Settings

Datasets We conducted experiments on four real-world datasets: Wikipedia¹, Reddit², MOOC³, UCI⁴. Following prior work [Zhang *et al.*, 2022], we introduce structural perturbations to the graphs to simulate distributional shifts.

Evaluation Metrics. We partitioned the datasets based on the edge occurrence time: the initial 70% of edges were designated as the training set, the subsequent 15% were allocated to the validation set, and the remaining 15% formed the test set. We employ the average precision (AP) and area under the curve (AUC) as the evaluation metrics for link prediction. AP and AUC are two common metrics used to evaluate the performance of binary classification models. AP is a measure of the average precision across all possible recall thresholds. AUC is a measure of the area under the receiver operating characteristic (ROC) curve.

Training Protocols. An early-stopping mechanism was employed, terminating training when the Average Precision (AP) metric showed no improvement for five consecutive epochs.

The model underwent training for 300 epochs using the Adam optimizer with a learning rate set at 0.0001 and a weight decay of 1e-6. We set the batch size to 600, and the hidden layer dimension to 100. For the extraction of the causal subgraph, we specified the number of recent edges (N) as 50 and employed 1-hop neighbors. All MLP layers were configured to 2. Regarding the link prediction task, negative samples were set at a ratio of 1:5 in the training set and adjusted to 1:1 in both the validation and test sets.

Baselines. Given the limited research on invariant GNNs for CTDGs, we consider three categories of baselines: (1) **CTDG GNN Models:** TGN [Rossi *et al.*, 2020], TGAT [Xu *et al.*, 2020], and GraphMixer [Cong *et al.*, 2023]. These models are specifically designed for CTDGs but lack the ability to produce explainable and invariant results. Therefore, our comparison primarily focuses on link prediction. (2) **Robust DTDG models:** DIDA [Zhang *et al.*, 2022], EAGLE [Yuan *et al.*, 2023], and Ada-DyGNN [Li *et al.*, 2024a].

6.2 Link Prediction results (RQ1)

Table 1 reports the AUC results under varying shortcut ratios (0.2, 0.4, 0.6), while Table 2 presents the corresponding AP scores. Higher values indicate better performance. We summarize the key observations below: (1) CTDG models degrade sharply under stronger shortcuts. TGN suffers a substantial performance drop, with AUC decreasing by 31.4% on MOOC and by 23.8% on REDDIT. GraphMixer achieves competitive performance at 0.2 with CIR, but suffering drastic declines, particularly on MOOC (AUC decrease of 50.3%) and WIKI (AUC decrease of 33.3%). TGAT demonstrates relatively better stability but still degrades at higher shortcut ratios. (2) Robust DTDG models perform relatively well but do not generalize as effectively as CIR. Since DTDG models operate on discrete time steps, their effectiveness on CTDGs may be constrained, as they are not explicitly designed to handle fine-grained temporal dependencies. The relatively competitive performance of Ada-DyGNN and EAGLE sug-

¹<http://snap.stanford.edu/jodie/wikipedia.csv>

²<http://snap.stanford.edu/jodie/reddit.csv>

³<http://snap.stanford.edu/jodie/mooc.csv>

⁴<https://zenodo.org/records/7213796#.Y1cO6y8r30o>

Dataset	MOOC		REDDIT		WIKI		UCI	
	AP	AUC	AP	AUC	AP	AUC	AP	AUC
remove structural encoder	78.34±2.50	75.12±0.45	89.45±0.52	90.31±0.38	65.82±2.95	63.14±5.20	60.27±1.68	58.91±2.75
remove temporal encoder	80.12±2.42	77.85±0.41	91.62±0.49	92.78±0.36	68.30±2.88	66.42±5.08	63.50±1.62	61.75±2.60
remove Y^I	85.21±2.30	82.64±0.38	95.14±0.46	96.07±0.33	74.45±2.75	72.10±4.90	70.12±1.55	68.35±2.42
remove Y^C	81.73±1.26	83.47±0.22	89.31±0.98	90.67±0.52	70.98±1.63	68.49±1.90	67.82±0.96	62.46±0.87
CIR	90.18±2.17	88.24±0.27	98.18±0.32	98.15±0.25	80.37±2.02	79.10±4.53	76.84±0.97	73.55±1.81

Table 3: Ablation study.

gests that their robustness mechanisms still provide advantages, but their full potential might be underestimated due to the CTDG evaluation setting. (3) CIR achieves the best AUC score across all datasets and shortcut ratios. Even at the highest shortcut ratio (0.6), CIR maintains strong performance, significantly outperforming other models, demonstrating superior generalization under distributional shifts. (4) CIR maintains low variance across datasets and shortcut ratios, indicating consistent performance. In contrast, CTDG models, particularly GraphMixer and TGN, exhibit higher variance at larger shortcut ratios, suggesting instability when shortcut dependencies are removed. TGAT shows moderate variance but still struggles with higher shift levels. Among DTDG models, Ada-DyGNN demonstrates lower variance than CTDG models, reflecting its robustness in link prediction, while EAGLE’s variance remains moderate, indicating some inconsistency in handling shifts.

In conclusion, the results demonstrate that CIR achieves SOTA performance in link prediction across varying levels of distribution shift. Unlike existing CTDG and DTDG models, which either suffer from severe performance degradation or exhibit low AUC and AP scores, CIR consistently identifies stable causal structures and maintains high predictive performance with low variance. These results highlight CIR’s strong generalization ability and robustness to spurious correlations in CTDGs.

6.3 Efficiency (RQ2)

Figure 3 reports the average inference time per edge for each method. CIR demonstrates high efficiency, with average running times per edge of 5.4×10^{-4} seconds on Reddit and 5.8×10^{-4} seconds on MOOC. In contrast, TGAT and TGN are slower due to their more complex encoding networks, while DIDA is also slower because it must construct a confounder dictionary for each snapshot. GraphMixer is slightly more efficient than CIR, primarily because it does not output an explainable subgraph. Despite some differences in absolute running times between the Reddit and MOOC datasets, the relative performance across models is consistent, indicating that these efficiency measurements are reproducible across diverse dynamic graph scenarios.

6.4 Ablation Study (RQ3)

Table 3 depicts the ablation study by reporting the AP and AUC across datasets with a fixed shortcut ratio of 0.4. The results reveal that each component contributes significantly to the overall performance. Removing either the structural encoder or the temporal encoder leads to substantial performance drops, particularly on the WIKI and UCI datasets.

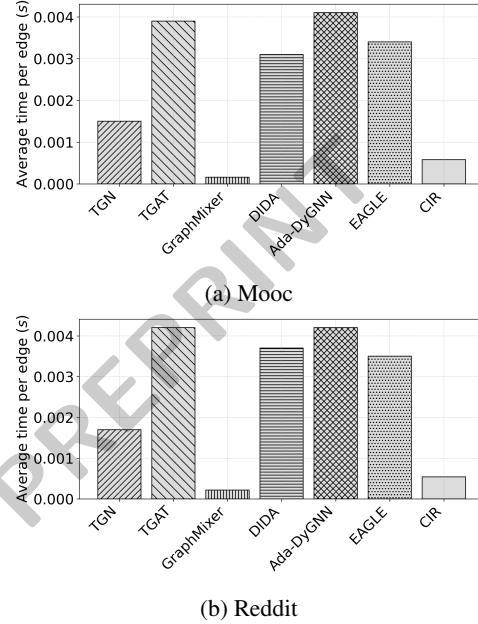


Figure 3: Average running time per edge.

Moreover, excluding the Y^C causes a marked decline in performance. In contrast, the removal of the Y^I has a more modest impact, with performance decreases ranging from about 2% to 8.9%. This suggests that while the ICM plays a supportive role in enhancing performance, the CCM is critical for capturing the essential patterns in the data. In summary, the full CIR model, which integrates all these components, consistently outperforms the ablated versions, highlighting the necessity of each module in achieving optimal performance across diverse datasets.

7 Conclusion

This paper addresses the challenge of developing invariant and self-interpretable GNNs for continuous-time dynamic graphs. By analyzing this problem from a causal effect perspective, we introduce the ICCM, a novel causal inference model meticulously designed to address both IID and OOD scenarios for CTDGs. Building upon the theoretical foundations of ICCM, we propose a novel deep learning architecture, which translates theoretically established causal models into a practical solution for dynamic graphs. Experiments show that CIR outperforms prior methods in link prediction, and robustness under OOD scenarios.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62472091, 62394333, 62572361)

References

- [Ahmed and Chen, 2016] Nahla Mohamed Ahmed and Ling Chen. An efficient algorithm for link prediction in temporal uncertain social networks. *Information Sciences*, 331:120–136, 2016.
- [Chang *et al.*, 2020] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.
- [Cong *et al.*, 2023] Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? In *International Conference on Learning Representations*, 2023.
- [De Winter *et al.*, 2018] Sam De Winter, Tim Decuyper, Sandra Mitrović, Bart Baesens, and Jochen De Weerd. Combining temporal aspects of dynamic networks with node2vec for a more efficient dynamic link prediction. In *Advances in Social Networks Analysis and Mining*, 2018.
- [Deng and Shen, 2024] Jiale Deng and Yanyan Shen. Self-interpretable graph learning with sufficient and necessary explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11749–11756, 2024.
- [Fan *et al.*, 2022] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. Debiasing graph neural networks via learning disentangled causal substructure. *Advances in Neural Information Processing Systems*, 35:24934–24946, 2022.
- [Feng *et al.*, 2023] Kaituo Feng, Changsheng Li, Xiaolu Zhang, and Jun Zhou. Towards open temporal graph neural networks. In *International Conference on Learning Representations*, 2023.
- [Li *et al.*, 2017] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2017.
- [Li *et al.*, 2023] Yiming Li, Yanyan Shen, Lei Chen, and Mingxuan Yuan. Zebra: When temporal graph neural networks meet temporal personalized pagerank. *Proceedings of the VLDB Endowment*, 16(6):1332–1345, 2023.
- [Li *et al.*, 2024a] Hanjie Li, Changsheng Li, Kaituo Feng, Ye Yuan, Guoren Wang, and Hongyuan Zha. Robust knowledge adaptation for dynamic graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6920–6933, 2024. DOI: 10.1109/TKDE.2024.3388453.
- [Li *et al.*, 2024b] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. Disentangled graph self-supervised learning for out-of-distribution generalization. In *Forty-first International Conference on Machine Learning*, 2024.
- [Li *et al.*, 2025a] Guoyi Li, Die Hu, Xiaomeng Fu, Qirui Tang, Yulei Wu, Xiaodan Zhang, and Honglei Lyu. Entity graph alignment and visual reasoning for multimodal fake news detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 2486–2495, 2025.
- [Li *et al.*, 2025b] Peiwen Li, Xin Wang, Zeyang Zhang, Ziwei Zhang, Fang Shen, Jialong Wang, Yang Li, and Wenwu Zhu. Causal-aware graph neural architecture search under distribution shifts. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 1458–1469, 2025.
- [Ma *et al.*, 2019] Yunpu Ma, Volker Tresp, and Erik A Daxberger. Embedding models for episodic knowledge graphs. *Journal of Web Semantics*, 59:100490, 2019.
- [Mahdavi *et al.*, 2018] Sedigheh Mahdavi, Shima Khoshraftar, and Aijun An. dynnode2vec: Scalable dynamic network embedding. In *IEEE International Conference on Big Data*, 2018.
- [Ning *et al.*, 2025] Xuecheng Ning, Yujie Wang, Kui Yu, Jiali Miao, Fuyuan Cao, and Jiye Liang. Summary graph induced invariant learning for generalizable graph learning. *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- [Pearl *et al.*, 2016] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.
- [Peng *et al.*, 2025] Yufei Peng, Cheng Yang, Zhengjie Fan, and Chuan Shi. Data-centric prompt tuning for dynamic graphs. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 2336–2345, 2025.
- [Qin *et al.*, 2023] Xiao Qin, Nasrullah Sheikh, Chuan Lei, Berthold Reinwald, and Giacomo Domeniconi. Seign: A simple and efficient graph neural network for large dynamic graphs. In *IEEE International Conference on Data Engineering*, 2023.
- [Ranjan *et al.*, 2020] Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5470–5477, 2020.
- [Rossi *et al.*, 2020] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *International Conference on Machine Learning*, 2020.
- [Sankar *et al.*, 2020] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *The International Conference on Web Search and Data Mining*, 2020.

- [Sui *et al.*, 2022] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.
- [Tolstikhin *et al.*, 2021] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*, volume 34, pages 24261–24272, 2021.
- [Trivedi *et al.*, 2019] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019.
- [Wang *et al.*, 2025] Botao Wang, Jia Li, Heng Chang, Keli Zhang, and Fugee Tsung. Heterophilic graph neural networks optimization with causal message-passing. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 829–837, 2025.
- [Wu *et al.*, 2022] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. In *International Conference on Learning Representations*, 2022.
- [Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2015.
- [Xu *et al.*, 2020] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020.
- [Yu *et al.*, 2018] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2672–2681, 2018.
- [Yuan *et al.*, 2023] Haonan Yuan, Qingyun Sun, Xingcheng Fu, Ziwei Zhang, Cheng Ji, Hao Peng, and Jianxin Li. Environment-aware dynamic graph learning for out-of-distribution generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. <https://openreview.net/forum?id=n8JWlZYPRz>.
- [Yuan *et al.*, 2025a] Qi Yuan, Yang Liu, Yateng Tang, Xinhuan Chen, Xuehao Zheng, Qing He, and Xiang Ao. Dynamic graph learning with static relations for credit risk assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13133–13141, 2025.
- [Yuan *et al.*, 2025b] Ruiwen Yuan, Yongqiang Tang, and Wensheng Zhang. A structure-aware invariant learning framework for node-level graph ood generalization. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1879–1890, 2025.
- [Zhang *et al.*, 2022] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. *Advances in Neural Information Processing Systems*, 35:6074–6089, 2022.
- [Zhang *et al.*, 2025] He Zhang, Bang Wu, Xiangwen Yang, Xingliang Yuan, Xiaoning Liu, and Xun Yi. Dynamic graph unlearning: A general and efficient post-processing method via gradient transformation. In *Proceedings of the ACM on Web Conference 2025*, pages 931–944, 2025.