Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift

Zeyang Zhang[1]*, Xin Wang[1]†, Ziwei Zhang[1], Haoyang Li[1], Zhou Qin[2], Wenwu Zhu[1]†
[1]Tsinghua University, [2]Alibaba Group
zy-zhang20@mails.tsinghua.edu.cn, {xin_wang, zwzhang}@tsinghua.edu.cn,
lihy18@mails.tsinghua.edu.cn, qinzhou.qinzhou@alibaba-inc.com,
wwzhu@tsinghua.edu.cn

Abstract

Dynamic graph neural networks (DyGNNs) have demonstrated powerful predictive abilities by exploiting graph structural and temporal dynamics. However, the existing DyGNNs fail to handle distribution shifts, which naturally exist in dynamic graphs, mainly because the patterns exploited by DyGNNs may be variant with respect to labels under distribution shifts. In this paper, we propose to handle spatio-temporal distribution shifts in dynamic graphs by discovering and utilizing *invariant patterns*, i.e., structures and features whose predictive abilities are stable across distribution shifts, which faces two key challenges: 1) How to discover the complex variant and invariant spatio-temporal patterns in dynamic graphs, which involve both time-varying graph structures and node features. 2) How to handle spatio-temporal distribution shifts with the discovered variant and invariant patterns. To tackle these challenges, we propose the Disentangled Intervention-based Dynamic graph Attention networks (**DIDA**). Our proposed method can effectively handle spatio-temporal distribution shifts in dynamic graphs by discovering and fully utilizing invariant spatio-temporal patterns. Specifically, we first propose a disentangled spatio-temporal attention network to capture the variant and invariant patterns. Then, we design a spatio-temporal intervention mechanism to create multiple interventional distributions by sampling and reassembling variant patterns across neighborhoods and time stamps to eliminate the spurious impacts of variant patterns. Lastly, we propose an invariance regularization term to minimize the variance of predictions in intervened distributions so that our model can make predictions based on invariant patterns with stable predictive abilities and therefore handle distribution shifts. Experiments on three real-world datasets and one synthetic dataset demonstrate the superiority of our method over state-of-the-art baselines under distribution shifts. Our work is the first study of spatio-temporal distribution shifts in dynamic graphs, to the best of our knowledge.

1 Introduction

Dynamic graphs widely exist in real-world applications, including financial networks [1, 2], social networks [3, 4], traffic networks [5, 6], etc. Distinct from static graphs, dynamic graphs can represent temporal structure and feature patterns, which are more complex yet common in reality. Dynamic graph neural networks (DyGNNs) have been proposed to tackle highly complex structural and temporal information over dynamic graphs, and have achieved remarkable progress in many predictive tasks [7, 8].

*This work was done during author's internship at Alibaba Group
†Corresponding authors

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Nevertheless, the existing DyGNNs fail to handle spatio-temporal distribution shifts, which naturally exist in dynamic graphs for various reasons such as survivorship bias [9], selection bias [10, 11], trending [12], etc. For example, in financial networks, external factors like period or market would affect the correlations between the payment flows and transaction illegitimacy [13]. Trends or communities also affect interaction patterns in coauthor networks [14] and recommendation networks [15]. If DyGNNs highly rely on spatio-temporal patterns which are variant under distribution shifts, they will inevitably fail to generalize well to the unseen test distributions.

To address this issue, in this paper, we study the problem of handling spatio-temporal distribution shifts in dynamic graphs through discovering and utilizing *invariant patterns*, i.e., structures and features whose predictive abilities are stable across distribution shifts, which remain unexplored in the literature. However, this problem is highly non-trivial with the following challenges:

- How to discover the complex variant and invariant spatio-temporal patterns in dynamic graphs, which include both graph structures and node features varying through time?
- How to handle spatio-temporal distribution shifts in a principled manner with discovered variant and invariant patterns?

To tackle these challenges, we propose a novel DyGNN named Disentangled Intervention-based Dynamic Graph Attention Networks (**DIDA**[3]). Our proposed method handles distribution shifts well by discovering and utilizing invariant spatio-temporal patterns with stable predictive abilities. Specifically, we first propose a disentangled spatio-temporal attention network to capture the variant and invariant patterns in dynamic graphs, which enables each node to attend to all its historic neighbors through a disentangled attention message-passing mechanism. Then, inspired by causal inference literatures [16, 17], we propose a spatio-temporal intervention mechanism to create multiple intervened distributions by sampling and reassembling variant patterns across neighborhoods and time, such that spurious impacts of variant patterns can be eliminated. To tackle the challenges that i) variant patterns are highly entangled across nodes and ii) directly generating and mixing up subsets of structures and features to do intervention is computationally expensive, we approximate the intervention process with summarized patterns obtained by the disentangled spatio-temporal attention network instead of original structures and features. Lastly, we propose an invariance regularization term to minimize prediction variance in multiple intervened distributions. In this way, our model can capture and utilize invariant patterns with stable predictive abilities to make predictions under distribution shifts. Extensive experiments on one synthetic dataset and three real-world datasets demonstrate the superiority of our proposed method over state-of-the-art baselines under distribution shifts. The contributions of our work are summarized as follows:

- We propose Disentangled Intervention-based Dynamic Graph Attention Networks (**DIDA**), which can handle spatio-temporal distribution shifts in dynamic graphs. This is the first study of spatio-temporal distribution shifts in dynamic graphs, to the best of our knowledge.
- We propose a disentangled spatio-temporal attention network to capture variant and invariant graph patterns. We further design a spatio-temporal intervention mechanism to create multiple intervened distributions and an invariance regularization term based on causal inference theory to enable the model to focus on invariant patterns under distribution shifts.
- Experiments on three real-world datasets and one synthetic dataset demonstrate the superiority of our method over state-of-the-art baselines.

2 Problem Formulation

In this section, we formulate the problem of spatio-temporal distribution shift in dynamic graphs.

Dynamic Graph. Consider a graph $\mathcal{G}$ with the node set $\mathcal{V}$ and the edge set $\mathcal{E}$. A dynamic graph can be defined as $\mathcal{G} = (\{\mathcal{G}^t\}_{t=1}^T)$, where T is the number of time stamps, $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ is the graph slice at time stamp t, $\mathcal{V} = \bigcup_{t=1}^T \mathcal{V}^t$, $\mathcal{E} = \bigcup_{t=1}^T \mathcal{E}^t$. For simplicity, a graph slice is also denoted as $\mathcal{G}^t = (\mathbf{X}^t, \mathbf{A}^t)$, which includes node features and adjacency matrix at time t. We use $\mathbf{G}^t$ to denote a random variable of $\mathcal{G}^t$.

[3] Our codes are publicly available at https://github.com/wondergo2017/DIDA

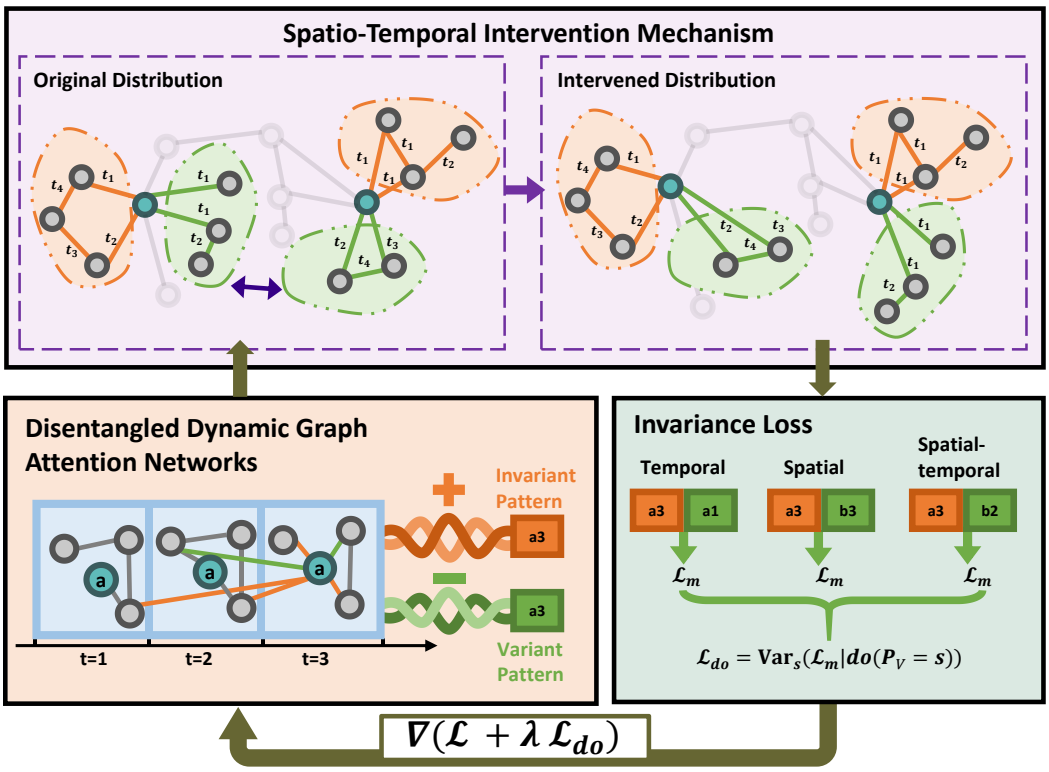


Figure 1: The framework of our proposed method **DIDA**. (Bottom left) For a given dynamic graph with multiple timestamps, the disentangled dynamic graph attention networks first obtain summarizations of high-order invariant and variant patterns by disentangled spatio-temporal message passing. (Top) Then the spatio-temporal intervention mechanism creates multiple intervened distributions by sampling and reassembling variant patterns across space and time for each node. (Bottom right) Last, invariance loss is calculated by using samples from intervened distributions to optimize the model so that it can focus on invariant patterns to make predictions.

Prediction tasks. For dynamic graphs, the prediction task can be summarized as using past graphs to make predictions, i.e. $p(\mathbf{Y}^t|\mathbf{G}^1, \mathbf{G}^2, \ldots, \mathbf{G}^t)$=$p(\mathbf{Y}^t|\mathbf{G}^{1:t})$, where label $\mathbf{Y}^t$ can be node properties or occurrence of links between nodes at time $t+1$. In this paper, we mainly focus on node-level tasks, which are commonly adopted in dynamic graph literatures [7, 8]. Following [18, 19], we factorize the distribution of graph trajectory into ego-graph trajectories, i.e. $p(\mathbf{Y}^t \mid \mathbf{G}^{1:t}) = \prod_v p(\mathbf{y}^t \mid \mathbf{G}_v^{1:t})$. An ego-graph induced from node v at time t is defined as $\mathcal{G}_v^t = (\mathbf{X}_v^t, \mathbf{A}_v^t)$ where $\mathbf{A}_v^t$ is the adjacency matrix including all edges in node v's L-hop neighbors at time t, i.e. $\mathcal{N}_v^t$, and $\mathbf{X}_v^t$ includes the features of nodes in $\mathcal{N}_v^t$. The optimization objective is to learn an optimal predictor with empirical risk minimization

$$\min_{\theta} \mathbb{E}_{(y^t, \mathcal{G}_v^{1:t}) \sim p_{tr}(\mathbf{y}^t, \mathbf{G}_v^{1:t})} \mathcal{L}(f_\theta(\mathcal{G}_v^{1:t}), y^t) \tag{1}$$

where f_θ is a learnable dynamic graph neural networks, We use $\mathbf{G}_v^{1:t}$,$\mathbf{y}^t$ to denote the random variable of the ego-graph trajectory and its label, and $\mathcal{G}_v^{1:t}$,y^t refer to the respective instances.

Spatio-temporal distribution shift. However, the optimal predictor trained with the training distribution may not generalize well to the test distribution when there exists a distribution shift problem. In the literature of dynamic graph, researchers are devoted to capture laws of network dynamics which are stable in systems [20, 21, 22, 23, 24]. Following them, we assume the conditional distribution is the same $p_{tr}(\mathbf{Y}^t|\mathbf{G}^{1:t}) = p_{te}(\mathbf{Y}^t|\mathbf{G}^{1:t})$, and only consider the covariate shift problem where $p_{tr}(\mathbf{G}^{1:t}) \neq p_{te}(\mathbf{G}^{1:t})$. Besides temporal distribution shift which naturally exists in time-varying data [25, 12, 26, 27, 28] and structural distribution shift in non-euclidean data [29, 18, 30], there exists a much more complex spatio-temporal distribution shift in dynamic graphs. For example, the distribution of ego-graph trajectories may vary across periods or communities.

3 Method

In this section, we propose Disentangled Intervention-based Dynamic Graph Attention Networks (**DIDA**) to handle spatio-temporal distribution shift in dynamic graphs. First, we propose a disentangled dynamic graph attention network to extract invariant and variant spatio-temporal patterns. Then we propose a spatio-temporal intervention mechanism to create multiple intervened data distributions. Finally, we optimize the model with invariance loss to make predictions relying on invariant patterns.

3.1 Handling Spatio-Temporal Distribution Shift

Spatio-Temporal Pattern. In recent decades of development of dynamic graphs, some scholars endeavor to conclude insightful patterns of network dynamics to reflect how real-world networks evolve through time [31, 32, 33, 34]. For example, the laws of triadic closure describe that two nodes with common neighbors (patterns) tend to have future interactions in social networks [35, 36, 23]. Besides structural information, node attributes are also an important part of the patterns, e.g., social interactions can be also affected by gender and age [37]. Instead of manually concluding patterns, we aim at learning the patterns using DyGNNs so that the more complex spatio-temporal patterns with mixed features and structures can be mined in dynamic graphs. Therefore, we define the spatio-temporal pattern used for node-level prediction as a subset of ego-graph trajectory

$$P^t(v) = m_v^t(\mathcal{G}_v^{1:t}) \tag{2}$$

where $m_v^t(\cdot)$ selects structures and attributes from the ego-graph trajectory. In [23], the pattern can be explained as an open triad with similar neighborhood, and the model tend to make link predictions to close the triad with $\hat{y}_{u,v}^t = f_\theta(P^t(u), P^t(v))$ based on the laws of triadic closure [38]. DyGNNs aim at exploiting predictive spatio-temporal patterns to boost prediction ability. However, the predictive power of some patterns may vary across periods or communities due to spatio-temporal distribution shift. Inspired by the causal theory [16, 17], we make the following assumption

Assumption 1 *For a given task, there exists a predictor $f(\cdot)$, for samples ($\mathcal{G}_v^{1:t}$,y^t) from any distribution, there exists an invariant pattern $P_I^t(v)$ and a variant pattern $P_V^t(v)$ such that $y^t = f(P_I^t(v)) + \epsilon$ and $P_I^t(v) = \mathcal{G}_v^{1:t} \backslash P_V^t(v)$, i.e., $\mathbf{y}^t \perp \mathbf{P}_V^t(v) \mid \mathbf{P}_I^t(v)$.*

Assumption 1 shows that invariant patterns $\mathbf{P}_I^t(v)$ are sufficiently predictive for label y^t and can be exploited across periods and communities without adjusting the predictor, while the influence of variant patterns $\mathbf{P}_V^t(v)$ on $\mathbf{y}^t$ is shielded by the invariant patterns.

Training Objective. Our main idea is that to obtain better generalization ability, the model should rely on invariant patterns instead of variant patterns, as the former is sufficient for prediction while the predictivity of the latter could be variant under distribution shift. Along this, our objective can be transformed to

$$\begin{aligned} &\min_{\theta_1,\theta_2} \mathbb{E}_{(y^t,\mathcal{G}_v^{1:t})\sim p_{tr}(\mathbf{y}^t,\mathbf{G}_v^{1:t})} \mathcal{L}(f_{\theta_1}(\tilde{P}_I^t(v)), y^t) \\ &s.t \quad \phi_{\theta_2}(\mathcal{G}_v^{1:t}) = \tilde{P}_I^t(v), \mathbf{y}^t \perp \tilde{\mathbf{P}}_V^t(v) \mid \tilde{\mathbf{P}}_I^t(v). \end{aligned} \tag{3}$$

where $f_{\theta_1}(\cdot)$ make predictions based on the invariant patterns, $\phi_{\theta_2}(\cdot)$ aims at finding the invariant patterns. Backed by causal theory[16, 17], it can be transformed into

$$\begin{aligned} &\min_{\theta_1,\theta_2} \mathbb{E}_{(y^t,\mathcal{G}_v^{1:t})\sim p_{tr}(\mathbf{y}^t,\mathbf{G}_v^{1:t})} \mathcal{L}(f_{\theta_1}(\phi_{\theta_2}(\mathcal{G}_v^{1:t})), y^t) + \\ &\lambda \text{Var}_{s\in\mathcal{S}}(\mathbb{E}_{(y^t,\mathcal{G}_v^{1:t})\sim p_{tr}(\mathbf{y}^t,\mathbf{G}_v^{1:t}|\text{do}(\mathbf{P}_V^t=s))} \mathcal{L}(f_{\theta_1}(\phi_{\theta_2}(\mathcal{G}_v^{1:t})), y^t)) \end{aligned} \tag{4}$$

where '`do`' denotes do-calculas to intervene the original distribution [39, 17], $\mathcal{S}$ denotes the intervention set and λ is a balancing hyperparameter. The idea can be informally described that as in Eq. (3), variant patterns $\mathbf{P}_V^t$ have no influence on the label $\mathbf{y}^t$ given the invariant patterns $\mathbf{P}_I^t$, then the prediction would not be varied if we intervene the variant patterns and keep invariant patterns untouched. More details about the connections between objective Eq.(3) and Eq.(4) can be found in Appendix.

Remark 1 *Minimizing the variance term in Eq.* (4) *help the model to satisfy the constraint of $\mathbf{y}^t \perp \tilde{\mathbf{P}}_V^t(v) \mid \tilde{\mathbf{P}}_I^t(v)$ in Eq.* (3)*, i.e., $p(\mathbf{y}^t \mid \tilde{\mathbf{P}}_I^t(v), \tilde{\mathbf{P}}_V^t(v)) = p(\mathbf{y}^t \mid \tilde{\mathbf{P}}_I^t(v))$*

3.2 Disentangled Dynamic Graph Attention Networks

Dynamic Neighborhood. To simultaneously consider the spatio-temporal information, we define the dynamic neighborhood as $\mathcal{N}^t(u) = \{v : (u, v) \in \mathcal{E}^t\}$, which includes all nodes that have interactions with node u at time t.

Disentangled Spatio-temporal Graph Attention Layer. To capture spatio-temporal pattern for each node, we propose a spatio-temporal graph attention to enable each node to attend to its dynamic neighborhood simultaneously. For a node u at time stamp t and its neighbors $v \in \mathcal{N}^{t'}(u), \forall t' \leq t$, we calculate the Query-Key-Value vectors as:

$$\mathbf{q}_u^t = \mathbf{W}_q(\mathbf{h}_u^t || \text{TE}(t)), \mathbf{k}_v^{t'} = \mathbf{W}_k(\mathbf{h}_v^{t'} || \text{TE}(t')), \mathbf{v}_v^{t'} = \mathbf{W}_v(\mathbf{h}_v^{t'} || \text{TE}(t')) \tag{5}$$

where $\mathbf{h}_u^t$ denotes the representation of node u at the time stamp t, $\mathbf{q}$, $\mathbf{k}$, $\mathbf{v}$ represents the query, key and value vector, respectively, and we omit the bias term for brevity. TE(t) denotes temporal encoding techniques to obtain embeddings of time t so that the time of link occurrence can be considered inherently [40, 41]. Then, we can calculate the attention scores among nodes in the dynamic neighborhood to obtain the structural masks

$$\mathbf{m}_I = \text{Softmax}(\frac{\mathbf{q} \cdot \mathbf{k}^T}{\sqrt{d}}), \mathbf{m}_V = \text{Softmax}(-\frac{\mathbf{q} \cdot \mathbf{k}^T}{\sqrt{d}}) \tag{6}$$

where d denotes feature dimension, $\mathbf{m}_I$ and $\mathbf{m}_V$ represent the masks of invariant and variant structural patterns. In this way, dynamic neighbors with higher attention scores in invariant patterns will have lower attention scores in variant ones, which means the invariant and variant patterns have a negative correlation. To capture invariant featural pattern, we adopt a learnable featural mask $\mathbf{m}_f = \text{Softmax}(\mathbf{w}_f)$ to select features from the messages of dynamic neighbors. Then the messages of dynamic neighborhood can be summarized with respective masks,

$$\begin{aligned} \mathbf{z}_I^t(u) &= \text{Agg}_I(\mathbf{m}_I, \mathbf{v} \odot \mathbf{m}_f)) \\ \mathbf{z}_V^t(u) &= \text{Agg}_V(\mathbf{m}_V, \mathbf{v})) \end{aligned} \tag{7}$$

where $\text{Agg}(\cdot)$ denotes aggregating and summarizing messages from dynamic neighborhood. To further disentangle the invariant and variant patterns, we design different aggregation functions $\text{Agg}_I(\cdot)$ and $\text{Agg}_V(\cdot)$ to summarize specific messages from masked dynamic neighborhood respectively. Then the pattern summarizations are added up as hidden embeddings to be fed into subsequent layers.

$$\mathbf{h}_u^t \leftarrow \mathbf{z}_I^t(u) + \mathbf{z}_V^t(u) \tag{8}$$

Overall Architecture. The overall architecture is a stacking of spatio-temporal graph attention layers. Like classic graph message-passing networks, this enables each node to access high-order dynamic neighborhood indirectly, where $\mathbf{z}_I^t(u)$ and $\mathbf{z}_V^t(u)$ at l-th layer can be a summarization of invariant and variant patterns in l-order dynamic neighborhood. In practice, the attention can be easily extended to multi-head attention [42] to stable the training process and model multi-faceted graph evolution [43].

3.3 Spatio-Temporal Intervention Mechanism

Direct Intervention. One way of intervening variant pattern distribution as Eq. (4) is directly generating and altering the variant patterns. However, this is infeasible in practice due to the following reasons: First, since it has to intervene the dynamic neighborhood and features node-wisely, the computational complexity is unbearable. Second, generating variant patterns including time-varying structures and features is another intractable problem.

Approximate Intervention. To tackle the problems mentioned above, we propose to approximate the patterns $\mathbf{P}^t$ with summarized patterns $\mathbf{z}^t$ found in Sec. 3.2. As $\mathbf{z}_I^t(u)$ and $\mathbf{z}_V^t(u)$ act as summarizations of invariant and variant spatio-temporal patterns for node u at time t, we approximate the intervention process by sampling and replacing the variant pattern summarizations instead of altering original structures and features with generated ones. To do spatio-temporal intervention, we collect variant patterns of all nodes at all time, from which we sample one variant pattern to replace the variant patterns of other nodes across time. For example, we can use the variant pattern of node v at time t_2 to replace the variant pattern of node u at time t_1 as

$$\mathbf{z}_I^{t_1}(u), \mathbf{z}_V^{t_1}(u) \leftarrow \mathbf{z}_I^{t_1}(u), \mathbf{z}_V^{t_2}(v) \tag{9}$$

As the invariant pattern summarization is kept the same, the label should not be changed. Thanks to the disentangled spatio-temporal graph attention, we get variant patterns across neighborhoods and time, which can act as natural intervention samples inside data so that the complexity of the generation problem can also be avoided. By doing Eq. (9) multiple times, we can obtain multiple intervened data distributions for the subsequent optimization.

3.4 Optimization with Invariance Loss

Based on the multiple intervened data distributions with different variant patterns, we can next optimize the model to focus on invariant patterns to make predictions. Here, we introduce invariance loss to instantiate Eq. (4). Let $\mathbf{z}_I$ and $\mathbf{z}_V$ be the summarized invariant and variant patterns, we calculate the task loss by only using the invariant patterns

$$\mathcal{L} = \ell(f(\mathbf{z}_I), \mathbf{y}) \tag{10}$$

where $f(\cdot)$ is the predictor. The task loss let the model utilize the invariant patterns to make predictions. Then we calculate the mixed loss as

$$\mathcal{L}_m = \ell(g(\mathbf{z}_V, \mathbf{z}_I), \mathbf{y}) \tag{11}$$

where another predictor $g(\cdot)$ makes predictions using both invariant patterns $\mathbf{z}_V$ and variant patterns $\mathbf{z}_I$. The mixed loss measure the model's prediction ability when variant patterns are also exposed to the model. Then the invariance loss is calculated by

$$\mathcal{L}_{do} = \text{Var}_{s_i \in \mathcal{S}}(\mathcal{L}_m | \text{do}(\mathbf{P}_V^t = s_i)) \tag{12}$$

where 'do' denotes the intervention mechanism as mentioned in Section. 3.3. The invariance loss measures the variance of the model's prediction ability under multiple intervened distributions. The final training objective is

$$\min_{\theta} \mathcal{L} + \lambda \mathcal{L}_{do} \tag{13}$$

where the task loss $\mathcal{L}$ is minimized to exploit invariant patterns while the invariance loss $\mathcal{L}_{do}$ helps the model to discover invariant and variant patterns, and λ is a hyperparameter to balance between two objectives. After training, we only adopt invariant patterns to make predictions in the inference stage. The overall algorithm is summarized in Table 1.

Algorithm 1 Training pipeline for **DIDA**

Require: Training epochs L, number of intervention samples S, hyperparameter λ

for $l = 1, \ldots, L$ **do**
  Obtain $\mathbf{z}_V^t, \mathbf{z}_I^t$ for each node and time as described in Section 3.2
  Calculate task loss and mixed loss as Eq. (10) and Eq. (11)
  Sample S variant patterns from collections of $\mathbf{z}_V^t$, to construct intervention set $\mathcal{S}$
  for s in $\mathcal{S}$ **do**
    Replace the nodes' variant pattern summarizations with s as Section 3.3
    Calculate mixed loss as Eq. (11)
  end for
  Calculate invariance loss as Eq. (12)
  Update the model according to Eq. (13)
end for

4 Experiments

In this section, we conduct extensive experiments to verify that our framework can handle spatio-temporal distribution shifts by discovering and utilizing invariant patterns. More Details of the settings and other results can be found in Appendix.

Baselines. We adopt several representative GNNs and Out-of-Distribution(OOD) generalization methods as our baselines:

- Static GNNs: **GAE** [44], a representative static GNN with stacking of graph convolutions; **VGAE** [44] further introduces variational variables into GAE.

- Dynamic GNNs: **GCRN** [45],a representative dynamic GNN that first adopts a GCN[44] to obtain node embeddings and then a GRU [46] to model the dynamics; **EvolveGCN** [13] adopts a LSTM[47] or GRU [46] to flexibly evolve the GCN[44] parameters instead of directly learning the temporal node embeddings; **DySAT** [43] models dynamic graph using structural and temporal self-attention.
- OOD generalization methods: **IRM** [48] aims at learning an invariant predictor which minimizes the empirical risks for all training domains; **GroupDRO** [49] reduces differences in risk across training domains to reduce the model's sensitivity to distributional shifts; **V-REx** [50] puts more weight on training domains with larger errors when minimizing empirical risk.

4.1 Real-world Datasets

Settings. We use 3 real-world dynamic graph datasets, including COLLAB, Yelp and Transaction. We adopt the challenging inductive future link prediction task, where the model exploits past graphs to make link prediction in the next time step. Each dataset can be split into several partial dynamic graphs based on its field information. For brevity, we use 'w/ DS' and 'w/o DS' to represent test data with and without distribution shift respectively. To measure models' performance under spatio-temporal distribution shift, we choose one field as 'w/ DS' and the left others are further split into training, validation and test data ('w/o DS') chronologically. Note that the 'w/o DS' is a merged dynamic graph without field information and 'w/ DS' is unseen during training, which is more practical and challenging in real-world scenarios. More details on their spatio-temporal distribution shifts are provided in Appendix. Here we briefly introduce the real-world datasets as follows

- **COLLAB** [51][4] is an academic collaboration dataset with papers that were published during 1990-2006. Node and edge represent author and coauthorship respectively. Based on the field of co-authored publication, each edge has the field information including "Data Mining", "Database", "Medical Informatics", "Theory" and "Visualization". The time granularity is year, including 16 time slices in total. We use "Data Mining" as 'w/ DS' and the left as 'w/o DS'.
- **Yelp** [43][5] is a business review dataset, containing customer reviews on business. Node and edge represent customer/business and review behavior respectively. We consider interactions in five categories of business including "Pizza", "American (New) Food", "Coffee & Tea ", "Sushi Bars" and "Fast Food" from January 2019 to December 2020. The time granularity is month, including 24 time slices in total. We use "Pizza" as 'w/ DS' and the left as 'w/o DS'.
- **Transaction**[6] is a secondary market transaction dataset, which records transaction behaviors of users from 10th April 2022 to 10th May 2022. Node and edge represent user and transaction respectively. The transactions have 4 categories, including "Pants", "Outwears", "Shirts" and "Hoodies". The time granularity is day, including 30 time slices in total. We use "Pants" as 'w/ DS' and the left as 'w/o DS'.

Results. Based on the results on real-world datasets in Table. 1, we have the following observations:

- Baselines fail dramatically under distribution shift: 1) Although DyGNN baselines perform well on test data without distribution shift, their performance drops greatly under distribution shift. In particular, the performance of DySAT, which is the best-performed DyGNN in 'w/o DS', drop by nearly 12%, 12% and 5% in 'w/ DS'. In Yelp and Transaction, GCRN and EGCN even underperform static GNNs, GAE and VGAE. This phenomenon shows that the existing DyGNNs may exploit variant patterns and thus fail to handle distribution shift. 2) Moreover, as generalization baselines are not specially designed to consider spatio-temporal distribution shift in dynamic graphs, they only have limited improvements in Yelp and Transaction. In particular, they rely on ground-truth environment labels to achieve OOD generalization, which are unavailable for real dynamic graphs. The inferior performance indicates that they cannot generalize well without accurate environment labels, which verifies that lacking environmental labels is also a key challenge for handling distribution shifts of dynamic graphs.
- Our method can better handle distribution shift than the baselines, especially in stronger distribution shift. **DIDA** improves significantly over all baselines in 'w/ DS' for all datasets. Note that

[4]https://www.aminer.cn/collaboration.

[5]https://www.yelp.com/dataset

[6]Collected from Alibaba.com

Table 1: Results(AUC%) of different methods on real-world datasets. The best results are in bold and the second-best results are underlined. 'w/o DS' and 'w/ DS' denote test data with and without distribution shift.

Model	**COLLAB**		**Yelp**		**Transaction**	
Test Data	w/o DS	w/ DS	w/o DS	w/ DS	w/o DS	w/ DS
GAE	77.15±0.50	74.04±0.75	70.67±1.11	64.45±5.02	71.90±0.32	73.44±0.41
VGAE	86.47±0.04	74.95±1.25	76.54±0.50	65.33±1.43	79.31±0.37	75.66±0.30
GCRN	82.78±0.54	69.72±0.45	68.59±1.05	54.68±7.59	78.99±0.28	71.24±0.35
EGCN	86.62±0.95	76.15±0.91	78.21±0.03	53.82±2.06	73.22±1.11	66.49±0.97
DySAT	88.77±0.23	76.59±0.20	78.87±0.57	66.09±1.42	81.55±0.66	76.18±0.43
IRM	87.96±0.90	75.42±0.87	66.49±10.78	56.02±16.08	81.65±0.50	75.61±0.61
VREx	88.31±0.32	76.24±0.77	79.04±0.16	66.41±1.87	81.72±0.35	76.24±0.52
GroupDRO	88.76±0.12	76.33±0.29	**79.38±0.42**	66.97±0.61	81.50±0.24	75.92±0.37
DIDA	**91.97±0.05**	**81.87±0.40**	78.22±0.40	**75.92±0.90**	**83.08±0.33**	**77.61±0.59**

Table 2: Results(AUC%) of different methods on synthetic dataset. The best results are in bold and the second-best results are underlined. Larger $\overline{p}$ denotes higher distribution shift level.

Model $\backslash \overline{p}$	**0.4**		**0.6**		**0.8**	
Split	Train	Test	Train	Test	Train	Test
GCRN	69.60±1.14	72.57±0.72	74.71±0.17	72.29±0.47	75.69±0.07	67.26±0.22
EGCN	78.82±1.40	69.00±0.53	79.47±1.68	62.70±1.14	81.07±4.10	60.13±0.89
DySAT	84.71±0.80	70.24±1.26	89.77±0.32	64.01±0.19	94.02±1.29	62.19±0.39
IRM	85.20±0.07	69.40±0.09	89.48±0.22	63.97±0.37	**95.02±0.09**	62.66±0.33
VREx	84.77±0.84	70.44±1.08	89.81±0.21	63.99±0.21	94.06±1.30	62.21±0.40
GroupDRO	84.78±0.85	70.30±1.23	89.90±0.11	64.05±0.21	94.08±1.33	62.13±0.35
DIDA	**87.92±0.92**	**85.20±0.84**	**91.22±0.59**	**82.89±0.23**	92.72±2.16	**72.59±3.31**

Yelp has stronger temporal distribution shift since COVID-19 happens in the midway, strongly affecting consumers' behavior in business, while **DIDA** outperforms the most competitive baseline GroupDRO by 9% in 'w/ DS'. In comparison to similar field information in Yelp (all restaurants) and Transaction (all costumes), COLLAB has stronger spatial distribution shift since the fields are more different to each other, while **DIDA** outperforms the most competitive baseline DySAT by 5% in 'w/ DS'.

4.2 Synthetic Dataset

Settings. To evaluate the model's generalization ability under spatio-temporal distribution shift, following [18], we introduce manually designed shifts in dataset COLLAB with all fields merged. Denote original features and structures as $\mathbf{X}_1^t \in \mathbb{R}^{N\times d}$ and $\mathbf{A}^t \in \{0,1\}^{N\times N}$. For each time t, we uniformly sample $p(t)|\mathcal{E}^{t+1}|$ positive links and $(1-p(t))|\mathcal{E}^{t+1}|$ negative links in $\mathbf{A}^{t+1}$. Then they are factorized into variant features $\mathbf{X}_2^t \in \mathbb{R}^{N\times d}$ with property of structural preservation. Two portions of features are concatenated as $\mathbf{X}^t = [\mathbf{X}_1^t, \mathbf{X}_2^t]$ as input node features for training and inference. The sampling probability $p(t) = \text{clip}(\overline{p} + \sigma cos(t), 0, 1)$ refers to the intensity of shifts, where the variant features $\mathbf{X}_2^t$ constructed with higher $p(t)$ will have stronger correlations with future link $\mathbf{A}^{t+1}$. We set $\overline{p}_{test} = 0.1, \sigma_{test} = 0, \sigma_{train} = 0.05$ and vary $\overline{p}_{train}$ in from 0.4 to 0.8 for evaluation. Since the correlations between $\mathbf{X}_2^t$ and label $\mathbf{A}^{t+1}$ vary through time and neighborhood, patterns include $\mathbf{X}_2^t$ are variant under distribution shifts. As static GNNs can not support time-varing features, we omit their results.

Results. Based on the results on synthetic dataset in Table. 2, we have the following observations:

- Our method can better handle distribution shift than the baselines. Although the baselines achieve high performance when training, their performance drop drastically in the test stage, which shows that the existing DyGNNs fail to handle distribution shifts. In terms of test results, **DIDA** consistently outperforms DyGNN baselines by a significantly large margin. In particular, **DIDA** surpasses the best-performed baseline by nearly 13%/10%/5% in test results for different shift levels. For the general OOD baselines, they reduce the variance in some cases while their

improvements are not significant. Instead, **DIDA** is specially designed for dynamic graphs and can exploit the invariant spatio-temporal patterns to handle distribution shift.

- Our method can exploit invariant patterns to consistently alleviate harmful effects of variant patterns under different distribution shift levels. As shift level increases, almost all baselines increase in train results and decline in test results. This phenomenon shows that as the relationship between variant patterns and labels goes stronger, the existing DyGNNs become more dependent on the variant patterns when training, causing their failure in test stage. Instead, the rise in train results and drop in test results of **DIDA** are significantly lower than baselines, which demonstrates that **DIDA** can exploit invariant patterns and alleviate the harmful effects of variant patterns under distribution shift.

4.3 Complexity Analysis

We analyze the computational complexity of **DIDA** as follows. Denote $|V|$ and $|E|$ as the total number of nodes and edges in the graph, respectively, and d as the dimensionality of the hidden representation. The spatio-temporal aggregation has a time complexity of $O(|E|d + |V|d^2)$. The disentangled component adds a constant multiplier 2, which does not affect the time complexity of aggregation. Denote $|E_p|$ as the number of edges to predict and $|S|$ as the size of the intervention set. Our intervention mechanism has a time complexity of $O(|E_p||S|d)$ in training, and does not put extra time complexity in inference. Therefore, the overall time complexity of **DIDA** is $O(|E|d + |V|d^2 + |E_p||S|d)$. Notice that $|S|$ is a hyper-parameter and is usually set as a small constant. In summary, **DIDA** has a linear time complexity with respect to the number of nodes and edges, which is on par with the existing dynamic GNNs.

4.4 Ablation study

In this section, we conduct ablation studies to verify the effectiveness of the proposed spatio-temporal intervention mechanism and disentangled graph attention in **DIDA**.

Spatio-temporal intervention mechanism. We remove the intervention mechanism mentioned in Sec 3.3. From Figure 2, we can see that without spatio-temporal intervention, the model's performance drop significantly especially in the synthetic dataset, which verifies that our intervention mechanism helps the model to focus on invariant patterns to make predictions.

Disentangled graph attention. We further remove the disentangled attention mentioned in Sec 3.2. From Figure 2, we can see that disentangled attention is a critical component in the model design, especially in Yelp dataset. Moreover, without disentangled module, the model is unable to obtain variant and invariant patterns for the subsequent intervention.

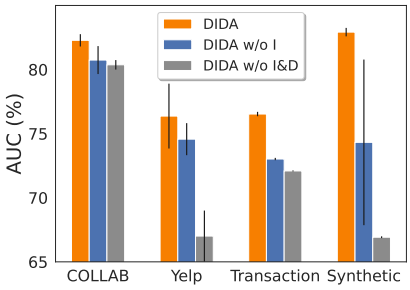


Figure 2: Ablation studies on intervention mechanism and disentangled attention, where 'w/o I' denotes removing the spatio-temporal intervention mechanism in **DIDA** and 'w/o I&D' further removes disentangled attention.

5 Related Work

Dynamic Graph Neural Networks. To tackle the complex structural and temporal information in dynamic graphs, considerable research attention has been devoted to dynamic graph neural networks (DyGNNs) [7, 8]. A classic of DyGNNs first adopt a GNN to aggregate structural information for graph at each time, followed by a sequence model like RNN [52, 53, 54, 45] or temporal self-attention [43] to process temporal information. Another classic of DyGNNs first introduce time-encoding techniques to represent each temporal link as a function of time, followed by a spatial module like GNN or memory module [20, 55, 40, 41] to process structural information. To obtain more fine-grained continous node embeddings in dynamic graphs, some work further leverages neural interaction processes [56] and ordinary differential equation [57]. DyGNNs have been widely applied

in real-world applications, including dynamic anomaly detection [58], event forecasting [59], dynamic recommendation [60], social character prediction [61], user modeling [62], temporal knowledge graph completion [63], etc. In this paper, we consider DyGNNs under spatio-temporal distribution shift, which remains unexplored in dynamic graph neural networks literature.

Out-of-Distribution Generalization. Most existing machine learning methods assume that the testing and training data are independent and identically distributed, which is not guaranteed to hold in many real-world scenarios [64]. In particular, there might be uncontrollable distribution shifts between training and testing data distribution, which may lead to sharp drop of model performance. To solve this problem, Out-of-Distribution (OOD) generalization problem has recently become a central research topic in various areas [65, 64, 66]. Recently, several works attempt to handle distribution shift on graphs [67, 29, 18, 68, 11, 69, 70, 71, 72, 73]. Another classic of OOD methods most related to our works handle distribution shifts on time-series data [25, 26, 12, 27, 28, 74]. Current works consider either only structural distribution shift for static graphs or only temporal distribution shift for time-series data. However, spatio-temporal distribution shifts in dynamic graphs are more complex yet remain unexplored. To the best of our knowledge, this is the first study of spatio-temporal distribution shifts in dynamic graphs.

Disentangled Representation Learning. Disentangled representation learning aims to characterize the multiple latent explanatory factors behind the observed data, where the factors are represented by different vectors [75]. Besides its applications in computer vision [76, 77, 78, 79, 80] and recommendation [81, 82, 83, 84, 85, 86], several disentangled GNNs have proposed to generalize disentangled representation learning in graph data recently. DisenGCN [87] and IPGDN [88] utilize the dynamic routing mechanism to disentangle latent factors for node representations. FactorGCN [89] decomposes the input graph into several interpretable factor graphs. DGCL [90, 91] aim to learn disentangled graph-level representations with self-supervision. Some works factorize deep generative models based on node, edge, static, dynamic factors [92] or spatial, temporal, graph factors [93] to achieve interpretable dynamic graph generation.

6 Conclusion

In this paper, we propose Disentangled Intervention-based Dynamic Graph Attention Networks (**DIDA**) to handle spatio-temporal distribution shift in dynamic graphs. First, we propose a disentangled dynamic graph attention network to capture invariant and variant spatio-temporal patterns. Then, based on the causal inference literature, we design a spatio-temporal intervention mechanism to create multiple intervened distributions and propose an invariance regularization term to help the model focus on invariant patterns under distribution shifts. Extensive experiments on three real-world datasets and one synthetic dataset demonstrate that our method can better handle spatio-temporal distribution shift than state-of-the-art baselines. One limitation is that in this paper we mainly consider dynamic graphs in scenarios of discrete snapshots, and we leave studying spatio-temporal distribution shifts in continous dynamic graphs for further explorations.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China No. 2020AAA0106300, National Natural Science Foundation of China (No. 62250008, 62222209, 62102222, 62206149), China National Postdoctoral Program for Innovative Talents No. BX20220185 and China Postdoctoral Science Foundation No. 2022M711813. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Diego C Nascimento, Bruno A Pimentel, Renata MCR Souza, Lilia Costa, Sandro Gonçalves, and Francisco Louzada. Dynamic graph in a symbolic data framework: An account of the causal relation using covid-19 reports and some reflections on the financial world. *Chaos, Solitons & Fractals*, 153:111440, 2021.

[2] Shilei Zhang, Toyotaro Suzumura, and Li Zhang. Dyngraphtrans: Dynamic graph embedding via modified universal transformer networks for financial transaction data. In *2021 IEEE International Conference on Smart Data Services (SMDS)*, pages 184–191. IEEE, 2021.

[3] Tanya Y Berger-Wolf and Jared Saia. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 523–528, 2006.

[4] Derek Greene, Donal Doyle, and Padraig Cunningham. Tracking the evolution of communities in dynamic social networks. In *2010 international conference on advances in social networks analysis and mining*, pages 176–183. IEEE, 2010.

[5] Hao Peng, Bowen Du, Mingsheng Liu, Mingzhe Liu, Shumei Ji, Senzhang Wang, Xu Zhang, and Lifang He. Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. *Information Sciences*, 578:401–416, 2021.

[6] Hao Peng, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, Zeyu Yang, Linfeng Du, Senzhang Wang, et al. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521:277–290, 2020.

[7] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access*, 9:79143–79168, 2021.

[8] Yuecai Zhu, Fuyuan Lyu, Chengming Hu, Xi Chen, and Xue Liu. Learnable encoder-decoder architecture for dynamic graph: A survey. *arXiv preprint arXiv:2203.10480*, 2022.

[9] Stephen J Brown, William Goetzmann, Roger G Ibbotson, and Stephen A Ross. Survivorship bias in performance studies. *The Review of Financial Studies*, 5(4):553–580, 1992.

[10] Richard A Berk. An introduction to sample selection bias in sociological data. *American sociological review*, pages 386–398, 1983.

[11] Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust gnns: Overcoming the limitations of localized graph training data. *Advances in Neural Information Processing Systems*, 34, 2021.

[12] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

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

[14] Tian Jin, Qiong Wu, Xuan Ou, and Jianjun Yu. Community detection and co-author recommendation in co-author networks. *International Journal of Machine Learning and Cybernetics*, 12(2):597–609, 2021.

[15] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. Causal representation learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 3562–3571, 2022.

[16] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2, 2000.

[17] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[18] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*, 2022.

[19] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. *Advances in Neural Information Processing Systems*, 33:5862–5874, 2020.

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

[21] Zhenyu Qiu, Wenbin Hu, Jia Wu, Weiwei Liu, Bo Du, and Xiaohua Jia. Temporal network embedding with high-order nonlinear information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5436–5443, 2020.

[22] Hong Huang, Zixuan Fang, Xiao Wang, Youshan Miao, and Hai Jin. Motif-preserving temporal network embedding. In *IJCAI*, pages 1237–1243, 2020.

[23] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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

[25] Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad-Javad Darvishi-Bayazi, Guillaume Dumas, and Irina Rish. Woods: Benchmarks for out-of-distribution generalization in time series tasks. *arXiv preprint arXiv:2203.09978*, 2022.

[26] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 402–411, 2021.

[27] Praveen Venkateswaran, Vinod Muthusamy, Vatche Isahagian, and Nalini Venkatasubramanian. Environment agnostic invariant risk minimization for classification of sequential datasets. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1615–1624, 2021.

[28] Wang Lu, Jindong Wang, Yiqiang Chen, and Xinwei Sun. Diversify to generalize: Learning generalized representations for time series classification. 2021.

[29] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*, 2022.

[30] Mucong Ding, Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Micah Goldblum, David Wipf, Furong Huang, and Tom Goldstein. A closer look at distribution shifts and out-of-distribution generalization on graphs. 2021.

[31] Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(11):P11005, 2011.

[32] Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

[33] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 601–610, 2017.

[34] Marinka Zitnik, Rok Sosič, Marcus W Feldman, and Jure Leskovec. Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences*, 116(10):4426–4433, 2019.

[35] James S Coleman. *Foundations of social theory*. Harvard university press, 1994.

[36] Hong Huang, Jie Tang, Lu Liu, JarDer Luo, and Xiaoming Fu. Triadic closure pattern analysis and prediction in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3374–3389, 2015.

[37] Lauri Kovanen, Kimmo Kaski, János Kertész, and Jari Saramäki. Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences. *Proceedings of the National Academy of Sciences*, 110(45):18070–18075, 2013.

[38] Georg Simmel. *The sociology of georg simmel*, volume 92892. Simon and Schuster, 1950.

[39] Jin Tian, Changsung Kang, and Judea Pearl. *A characterization of interventional distributions in semi-Markovian causal models*. eScholarship, University of California, 2006.

[40] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

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

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

[43] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 519–527, 2020.

[44] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[45] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

[46] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[48] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[49] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[50] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021.

[51] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In *KDD'2012*, 2012.

[52] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1975–1985, 2021.

[53] Li Sun, Zhongbao Zhang, Jiawei Zhang, Feiyang Wang, Hao Peng, Sen Su, and Philip S Yu. Hyperbolic variational graph neural network for modeling dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4375–4383, 2021.

[54] Ehsan Hajiramezanali, Arman Hasanzadeh, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. Variational graph recurrent neural networks. *Advances in neural information processing systems*, 32, 2019.

[55] Weilin Cong, Yanhong Wu, Yuandong Tian, Mengting Gu, Yinglong Xia, Mehrdad Mahdavi, and Chuncheng Jason Chen. Dynamic graph representation learning via graph transformer networks. *arXiv preprint arXiv:2111.10447*, 2021.

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

[57] Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *KDD*, pages 705–715, 2021.

[58] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *Proceedings of the 30th ACM international conference on Information & Knowledge Management*, pages 3747–3756, 2021.

[59] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1585–1595, 2020.

[60] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenburg, and Jure Leskovec. Hierarchical temporal convolutional networks for dynamic recommender systems. In *The world wide web conference*, pages 2236–2246, 2019.

[61] Yanbang Wang, Pan Li, Chongyang Bai, and Jure Leskovec. Tedic: Neural modeling of behavioral patterns in dynamic social interaction networks. In *Proceedings of the Web Conference 2021*, pages 693–705, 2021.

[62] Haoyang Li, Peng Cui, Chengxi Zang, Tianyang Zhang, Wenwu Zhu, and Yishi Lin. Fates of microscopic social ecosystems: Keep alive or dead? In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 668–676, 2019.

[63] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. Temp: Temporal message passing for temporal knowledge graph completion. *arXiv preprint arXiv:2010.03526*, 2020.

[64] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

[65] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv e-prints*, pages arXiv–2103, 2021.

[66] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *Proceeding of the Thirty-ninth International Conference on Machine Learning*, 2022.

[67] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987*, 2022.

[68] Yongqiang Chen, Yonggang Zhang, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Invariance principle meets out-of-distribution generalization on graphs. *arXiv preprint arXiv:2202.05441*, 2022.

[69] Yijian Qin, Xin Wang, Ziwei Zhang, Pengtao Xie, and Wenwu Zhu. Graph neural architecture search under distribution shifts. In *International Conference on Machine Learning*, pages 18083–18095. PMLR, 2022.

[70] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[71] Zeyang Zhang, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning to solve travelling salesman problem with hardness-adaptive curriculum. *arXiv preprint arXiv:2204.03236*, 2022.

[72] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. Generalizing graph neural networks on out-of-distribution graphs. *arXiv preprint arXiv:2111.10657*, 2021.

[73] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.

[74] Huaxiu Yao, Caroline Choi, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Proceedings of the Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[75] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[76] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 31, 2018.

[77] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2018.

[78] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[79] Emily L Denton et al. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, 30, 2017.

[80] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017.

[81] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. Disentangled representation learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[82] Hong Chen, Yudong Chen, Xin Wang, Ruobing Xie, Rui Wang, Feng Xia, and Wenwu Zhu. Curriculum disentangled recommendation with noisy multi-feedback. *Advances in Neural Information Processing Systems*, 34:26924–26936, 2021.

[83] Xin Wang, Hong Chen, and Wenwu Zhu. Multimodal disentangled representation for recommendation. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[84] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 483–491, 2020.

[85] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. *Advances in neural information processing systems*, 32, 2019.

[86] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[87] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *International conference on machine learning*, pages 4212–4221. PMLR, 2019.

[88] Yanbei Liu, Xiao Wang, Shu Wu, and Zhitao Xiao. Independence promoted graph disentangled networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4916–4923, 2020.

[89] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33:20286–20296, 2020.

[90] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems*, 34:21872–21884, 2021.

[91] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Disentangled graph contrastive learning with independence promotion. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[92] Wenbin Zhang, Liming Zhang, Dieter Pfoser, and Liang Zhao. Disentangled dynamic graph deep generation. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 738–746. SIAM, 2021.

[93] Yuanqi Du, Xiaojie Guo, Hengning Cao, Yanfang Ye, and Liang Zhao. Disentangled spatiotemporal graph generative models. *arXiv preprint arXiv:2203.00411*, 2022.

[94] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

[95] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020.

[96] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR, 2020.

[97] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020.

[98] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the national academy of sciences*, 101(11):3747–3752, 2004.

[99] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[100] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

[102] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in neural information processing systems*, 32, 2019.

[103] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[104] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Notations

Notations	Descriptions
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A graph with the node set and edge set
$\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$	Graph slice at time t
$\mathbf{X}^t, \mathbf{A}^t$	Features and adjacency matrix of a graph at time t
$\mathcal{G}^{1:t}, Y^t, \mathbf{G}^{1:t}, \mathbf{Y}^t$	Graph trajectory, label and their corresponding random variable
$\mathcal{G}_v^{1:t}, y^t, \mathbf{G}_v^{1:t}, \mathbf{y}^t$	Ego-graph trajectory, the node's label and their corresponding random variable
$f(\cdot), g(\cdot)$	Predictors
$P, \mathbf{P}$	Pattern and its corresponding random variable
$m(\cdot)$	Function to select structures and features from ego-graph trajectories
$\text{do}(\cdot)$	do-calculus
$\phi(\cdot)$	Function to find invariant patterns
d	The dimensionality of node representation
$\mathbf{q}, \mathbf{k}, \mathbf{v}$	Query, key and value vector
$\mathcal{N}^t(u)$	Dynamic neighborhood of node u at time t
$\mathbf{m}_I, \mathbf{m}_V, \mathbf{m}_f$	Structural mask of invariant and variant patterns, and featural mask
$\mathbf{z}_I^t(u), \mathbf{z}_V^t(u)$	Summarizations of invariant and variant patterns for node u at time t
$\text{Agg}_I(\cdot)$, $\text{Agg}_V(\cdot)$	Aggregation functions for invariant and variant patterns
$\mathbf{h}_u^t$	Hidden embeddings for node u at time t
ℓ	Loss function
$\mathcal{L}, \mathcal{L}_m, \mathcal{L}_{do}$	Task loss, mixed loss and invariance loss

B More Details on Section 3.1

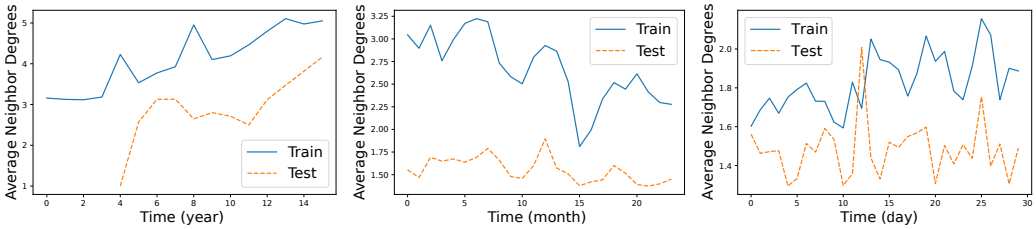
Background of Assumption 1. It is widely adopted in out-of-distribution generalization literature [25, 48, 94, 29, 95, 96, 97] about the assumption that the relationship between labels and some parts of features is invariant across data distributions, and these subsets of features with such properties are called invariant features. In this paper, we use invariant patterns $\mathbf{P}_I$ to denote the invariant structures and features. From the causal perspective, we can formulate the data-generating process in dynamic graphs with a structural causal model (SCM) [16, 17], $\mathbf{P}_V \rightarrow \mathbf{G} \leftarrow \mathbf{P}_I \rightarrow \mathbf{y}$ and $\mathbf{P}_V \leftarrow \mathbf{P}_I$, where the arrow between variables denotes casual relationship, and the subscript v and superscript t are omitted for brevity. $\mathbf{P}_V \rightarrow \mathbf{G} \leftarrow \mathbf{P}_I$ denotes that variant and invariant patterns construct the ego-graph trajectories observed in the data, while $\mathbf{P}_I \rightarrow \mathbf{y}$ denotes that invariant patterns determine the ground truth label $\mathbf{y}$, no matter how the variant patterns change inside data across different distributions. Sometimes, the correlations between variant patterns and labels may be built by some exogenous factors like periods and communities. In some distributions, $\mathbf{P}_V \leftarrow \mathbf{P}_I$ would open a backdoor path [17] $\mathbf{P}_V \leftarrow \mathbf{P}_I \rightarrow \mathbf{y}$ so that variant patterns $\mathbf{P}_V$ and labels $\mathbf{y}$ are correlated statistically, and this correlation is also called spurious correlation. If the model highly relies on the relationship between variant patterns and labels, it will fail under distribution shift, since such relationship varies across distributions. Hence, we propose to help the model focus on invariant patterns to make predictions and thus handle distribution shift.

Connnections in Remark 1. To eliminate the spurious correlation between variant patterns and labels, one way is to block the backdoor path by using do-calculus to intervene variant patterns. By applying do-calculus on one variable, all in-coming arrows(causal relationship) to it will be removed [17] and the intervened distributions will be created. In our case, the operator $\text{do}(\mathbf{P}_V)$ will cut the causal relationship from invariant patterns to variant patterns, i.e. disabling $\mathbf{P}_V \leftarrow \mathbf{P}_I$ and then blocking the backdoor path $\mathbf{P}_V \leftarrow \mathbf{P}_I \rightarrow \mathbf{y}$. Hence, the model can learn the direct causal effects from invariant patterns to labels in the intervened distributions $p(\mathbf{y}, \mathbf{G}|\text{do}(\mathbf{P}_V))$, and the risks should be the same across these intervened distributions. Therefore we can minimize the variance of empirical risks under different intervened distributions to help the model focus on the relationship between invariant patterns and labels. On the other hand, if we have the optimal predictor $f_{\theta_1}^*$ and pattern finder $\phi_{\theta_2}^*$ according to Eq.(3), then the variance term in Eq.(4) is minimized as the variant patterns will not affect the predictions of $f_{\theta_1}^* \circ \phi_{\theta_2}^*$ across different intervened distributions.

C Additional Experiments

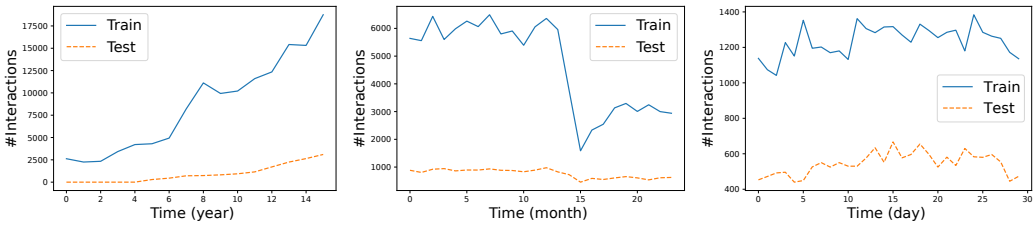
C.1 Distribution Shifts in Real-world Datasets

We illustrate the distribution shifts in the real-world datasets with two statistics, number of links and average neighbor degrees [98]. Figure 3 shows that the average neighbor degrees are lower in test data compared to training data. Lower average neighbor degree indicates that the nodes have less affinity to connect with high-degree neighbors. Moreover, in COLLAB, the test data has less history than training data, i.e. the graph trajectory is not always complete in training and test data distribution. This phenomenon of incomplete history is



(a) COLLAB (b) Yelp (c) Transaction

Figure 3: Average neighbor degrees in the graph slice as time goes.



(a) COLLAB (b) Yelp (c) Transaction

Figure 4: Number of links in the graph slice as time goes.

common in real-world scenarios, e.g. not all the users join the social platforms at the same time. Figure 4 shows that the number of links and its trend also differ in training and test data. In COLLAB, #links of test data has a slower rising trend than training data. In Yelp, #links of training and test data both have a drop during time 13-15 and rise again thereafter, due to the outbreak of COVID-19, which strongly affected the consumers' behavior.

C.2 Spatial or Temporal Intervention

We compare two other versions of **DIDA** , where **DIDA-S** only uses spatial intervention and **DIDA-T** only uses temporal intervention. For **DIDA-S**, we put the constraint that the variant patterns used to intervene must come from the same timestamp in Eq.(9) so that the variant patterns across time are forbidden for intervention. Similarly, we put the constraint that the variant patterns used to intervene must come from the same node in

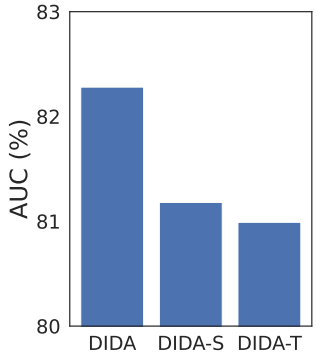


Figure 5: Comparison of different intervention mechansim on COLLAB dataset, where **DIDA-S** only uses spatial intervention and **DIDA-T** only uses temporal intervention.

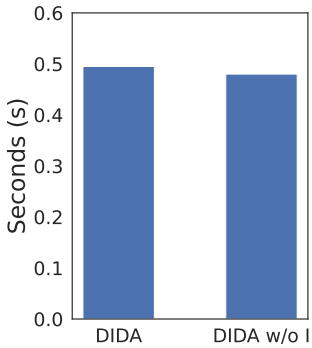
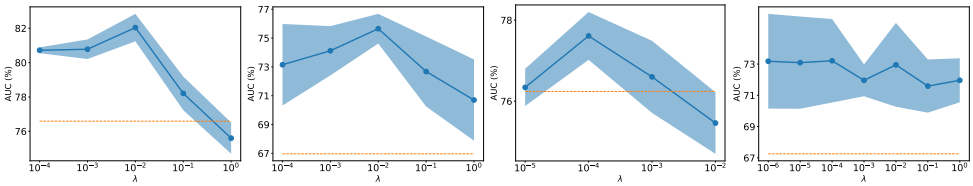


Figure 6: Comparison in terms of training time for each epoch on COLLAB dataset, where 'w/o I' means removing intervention mechanism in **DIDA**.



(a) COLLAB (b) Yelp (c) Transaction (d) Synthetic

Figure 7: Sensitivity of hyperparameter λ. The area shows the average AUC and standard deviations in the test stage. The dashed line represents the average AUC of the best performed baseline.

Eq.(9) for **DIDA-T**. Figure 5 shows that **DIDA** improves significantly over the other two ablated versions, which verifies that it is important to take into consideration both the spatial and temporal aspects of distribution shifts.

C.3 Efficiency of Intervention

For **DIDA** and **DIDA** without intervention mechanism, we compare their training time for each epoch on COLLAB dataset. As shown in Figure 6, the intervention mechanism adds few costs in training time (lower than 5%). Moreover, as **DIDA** does not use the intervention mechanism in the test stage, it does not add extra computational costs in the inference time.

C.4 Hyperparameter Sensitivity

We analyze the sensitivity of hyperparameter λ in **DIDA** for each dataset. From Figure 7, we can see that as λ is too small or too large, the model's performance drops in most datasets. It shows that λ acts as a balance between how **DIDA** exploits the patterns and satisfies the invariance constraint.

C.5 Case Study

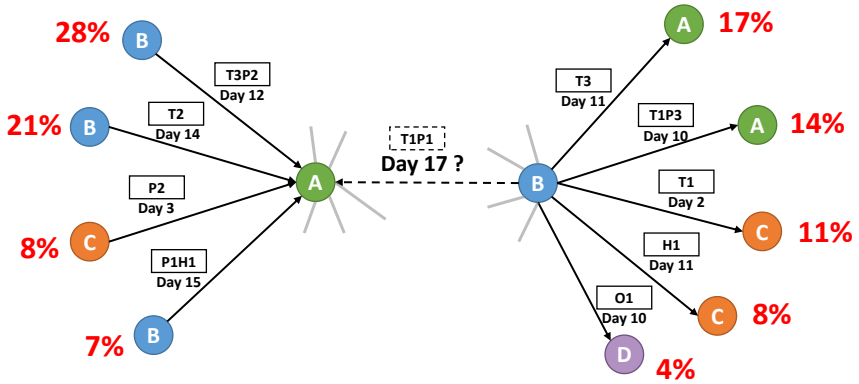


Figure 8: Case study: **DIDA**'s attention scores for invariant patterns, which are shown in percentage and marked red. Nodes and links represent users and transactions respectively. Links with smaller attention scores are omitted for brevity. For each link, the direction denotes a selling behavior, and it is tagged with the trading goods and transaction time on the link. For example, the link from B to A with tagging 'Day 12' and 'T3P2' represents that user B sells user A three T-shirts and two pants on day 12. The trading goods have four types: T, P, H, O represent T-shirt, Pants, Hoddie, and Outerwear respectively. Based on the transactions in their dynamic neighborhood, **DIDA** predicts whether user A will buy something from user B on day 17, and the dashed bounding box 'T1P1' refers to ground-truth trading goods.

Figure 8 illustrates **DIDA**'s attention scores for invariant patterns. We use a well-trained **DIDA** on Transaction dataset and show the scores of the structural mask for invariant patterns, i.e. $\mathbf{m}_I$ in Eq.(6). In this case, **DIDA** predicts whether user A will buy something ('T1P1') from user B on day 17, based on the transactions in their dynamic neighborhood. We have the following observations:

Table 3: Summarization of dataset statistics.

Dataset	# Timestamps	# Nodes	# Links	Temporal Granularity	Feature Dimension
COLLAB	16	23035	151790	year	32
Yelp	24	13095	65375	month	32
Transaction	30	29526	53448	day	15

- Transactions with higher attention scores are more correlated with the transaction to predict. The transactions with attention scores 28% and 21% include goods 'T3P2' and 'T2', which are closed to the ground-truth trading goods 'T1P1' from B to A on day 17. On the right side, similarly, the transactions with attention scores 17%, 14% and 11% include goods 'T3', 'T1P3' and 'T1', which are closed to 'T1P1' as well. In contrast, the transactions with other unrelated goods like 'H1' and 'O1' have even smaller attention scores.
- Dynamic information is critical in the attention scores for invariant patterns. For the transactions with attention scores 21% and 8% on the left side, they include goods 'T2' and 'P2' which are both similar to 'T1P1'. However, the latter's attention score is much lower than the former's. This is because the latter happens much earlier than the former, and **DIDA** learns to attend to more recent transactions to capture users' recent interest.

These observations indicate that **DIDA** can summarize invariant patterns in dynamic neighborhood to capture the users' interests in trading (user A as a recent T-shirt/Pants buyer and user B as a recent T-shirt/Pants seller) and make predictions by matching the summarized interests.

D Reproducibility Details

D.1 Training & Evaluation

Hyperparameters. For all methods, the hidden dimension is set to 16, the number of layers is set to 2. We adopt the Adam optimizer [99] with a learning rate 0.01, weight decay 5e-7 and set the patience of early stopping on the validation set as 50. For other hyperparameters, we adopt grid search for the best parameters using the validation split. For **DIDA**, we set the number of intervention samples as 1000 for all datasets, and λ as 1e-2,1e-2,1e-4,1e-1 for COLLAB, Yelp, Transaction, and Synthetic dataset respectively. The training pipeline for **DIDA** is shown in Algo. 1

Evaluation. We randomly sample negative samples from nodes that do not have links, and the negative samples for validation and testing set are kept the same for all comparing methods. The number of negative samples is the same as positive ones. We use Area under the ROC Curve (AUC) as the evaluation metric. We use the inner product of the two learned node representations to predict links and use cross-entropy as the loss function ℓ. We randomly run the experiments three times, and report the average results and standard deviations.

D.2 Dataset Details

We summarize dataset statistics in Table 3 and describe dataset details as follows.

COLLAB. [51][7] We use word2vec [100] to extract 32-dimensional feature from paper abstracts and average to obtain author features. We use 10,1,5 chronological graph slices for training, validation and test respectively. The dataset includes 23035 nodes and 151790 links in total.

Yelp. [43][8] We use word2vec [100] to extract 32-dimensional feature from reviews and average to obtain user and business features. We select users and items with interactions of more than 10. We use 15,1,8 chronological graph slices for training, validation and test respectively. The dataset includes 13095 nodes and 65375 links in total.

Transaction.[9] We calculate the distribution of users' historical transaction categories as the initial 15-dimensional features. We use 20,2,8 chronological graph slices for training, validation and test respectively. The dataset includes 29526 nodes and 53448 links in total.

Synthetic. We use the same features as $\mathbf{X}_1^t$ and structures as $\mathbf{A}^t$ in COLLAB, and introduce features $\mathbf{X}_2^t$ with variable correlation with supervision signals. $\mathbf{X}_2^t$ are obtained by training the embeddings $\mathbf{X}_2 \in \mathbb{R}^{N\times d}$ with reconstruction loss $\ell(\mathbf{X}_2\mathbf{X}_2^T, \tilde{\mathbf{A}}^{t+1})$, where $\tilde{\mathbf{A}}^{t+1}$ refers to the sampled links, and ℓ refers to cross-entropy loss function. The embeddings $\mathbf{X}_2^t$ are trained with Adam optimizer, learning rate 1e-1, weight decay 1e-5 and

[7] https://www.aminer.cn/collaboration.

[8] https://www.yelp.com/dataset

[9] Collected from Alibaba.com

earlystop patience 50. In this way, we empirically find that the inner product predictor can achieve results of over 99% AUC by using $\mathbf{X}_2^t$ to predict the sampled links $\tilde{\mathbf{A}}^{t+1}$, so that the generated features can have strong correlations with the sampled links. By controlling the p mentioned in the Section 4.2, we can control the correlations of $\mathbf{X}^t$ and labels $\mathbf{A}^{t+1}$ to vary in training and test stage.

D.3 Baseline Details

Backbones. This class of methods aim at improving the modeling ability for dynamic graphs.

- **GAE.** [44] A representative static GNN with stacking of graph convolutions.
- **VGAE.** [44] A representative static GNN which introduces variational variables into GAE.
- **GCRN.** [45] A representative dynamic GNN that first adopts a GCN to obtain node embeddings and then a GRU to model the dynamics.
- **DySAT.** [43] A representative dynamic GNN that models dynamic graph using structural and temporal self-attention.
- **EvolveGCN.** [13] A representative dynamic GNN that uses an RNN to evolve the GCN parameters instead of directly learning the temporal node embeddings.

OOD Generalization methods. This class of methods aim at improving the robustness and generalization ability of models against distribution shift. For fair comparison, we randomly split the samples into different domains, as the field information is unknown to all methods. Since they are general OOD generalization methods and are not specifically designed for dynamic graphs, we adopt DySAT as their backbone, which is the best-performed DyGNN on training dataset.

- **IRM** [48] aims at learning an invariant predictor which minimizes the empirical risks for all training domains.
- **VREx** [50] reduces differences in risk across training domains to reduce the model's sensitivity to distributional shifts.
- **GroupDRO** [49] puts more weight on training domains with larger errors when minimizing empirical risk.

D.4 Details of DIDA

First Layer. Before stacking of disentangled spatio-temporal graph attention Layers, we use a fully-connected layer $\text{FC}(\cdot)$ to transform the features into hidden embeddings.

$$\text{FC}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{14}$$

Agg Functions. We implement the aggregation function for invariant and variant patterns as

$$\begin{aligned} \tilde{\mathbf{z}}_I^t(u) &= \sum_i \mathbf{m}_{I,i}(\mathbf{v}_i \odot \mathbf{m}_f), \quad \mathbf{z}_I^t(u) = \text{FFN}(\tilde{\mathbf{z}}_I^t(u) + \mathbf{h}_u^t) \\ \tilde{\mathbf{z}}_V^t(u) &= \sum_i \mathbf{m}_{V,i}\mathbf{v}_i, \quad \mathbf{z}_V^t(u) = \text{FFN}(\tilde{\mathbf{z}}_V^t(u)) \end{aligned} \tag{15}$$

FFN. The FFN includes a layer normalization [101], multi-layer perceptron and skip connection.

$$\text{FFN}(\mathbf{x}) = \alpha \cdot \text{MLP}(\text{LayerNorm}(\mathbf{x})) + (1-\alpha)\cdot \mathbf{x} \tag{16}$$

where α is a learnable parameter.

Predictor. For link prediction task, we implement the predictor $f(\cdot)$ in Eq.(10) as inner product of hidden embeddings, i.e. $f(\mathbf{z}_I^t(u), \mathbf{z}_I^t(v)) = \mathbf{z}_I^t(u) \cdot (\mathbf{z}_I^t(v))^T$, which is conformed to classic link prediction settings. To implement the predictor $g(\cdot)$ in Eq.(11), we adopt the biased training technique following [102], i.e. $g(\mathbf{z}_V^t(u), \mathbf{z}_I^t(u), \mathbf{z}_V^t(v), \mathbf{z}_I^t(v)) = f(\mathbf{z}_I^t(u), \mathbf{z}_I^t(v)) \cdot \sigma(f(\mathbf{z}_V^t(u), \mathbf{z}_V^t(v)))$.

D.5 Configurations

Experiments on COLLAB, Yelp and Synthetic datasets are conducted with:

- Operating System: Ubuntu 18.04.1 LTS
- CPU: Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz
- GPU: NVIDIA GeForce RTX 3090 with 24 GB of memory
- Software: Python 3.8.13, Cuda 11.3, PyTorch [103] 1.11.0, PyTorch Geometric [104] 2.0.3.

Experiments on Transaction dataset are conducted with:

- Operating System: Ubuntu 18.04.5 LTS
- CPU: Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz
- GPU: NVIDIA Tesla V100 with 16 GB of memory
- Software: Python 3.6.12, Cuda 10.1, PyTorch [103] 1.8.2, PyTorch Geometric [104] 2.0.3.