Disentangled Dynamic Graph Attention Network for Out-of-distribution Sequential Recommendation

ZEYANG ZHANG, Department of Computer Science and Technology, Tsinghua University, China
XIN WANG*, Department of Computer Science and Technology, BNRist, Tsinghua University, China
HAIBO CHEN, Department of Computer Science and Technology, Tsinghua University, China
HAOYANG LI, Department of Computer Science and Technology, Tsinghua University, China
WENWU ZHU*, Department of Computer Science and Technology, BNRist, Tsinghua University, China

Sequential recommendation, leveraging user-item interaction histories to provide personalized and timely suggestions, has drawn significant research interest recently. With the power of exploiting spatio-temporal dynamics, dynamic graph neural networks (DyGNNs) show great potential in sequential recommendation by modeling the dynamic relationship between users and items. However, spatio-temporal distribution shifts naturally exist in out-of-distribution sequential recommendation, where both user-item relationships and temporal sequences demonstrate pattern shifts. The out-of-distribution scenarios may lead to the failure of existing DyGNNs in handling spatio-temporal distribution shifts in sequential recommendation, given that the patterns they exploit tend to be variant w.r.t labels under distribution shifts. In this paper, we propose Disentangled Intervention-based Dynamic graph Attention networks with Invariance Promotion (**I-DIDA**) to handle spatio-temporal distribution shifts in sequential recommendation by discovering and utilizing *invariant patterns*, *i.e.*, structures and features whose predictive abilities are stable across distribution shifts. Specifically, we first propose a disentangled spatio-temporal attention network to capture the variant and invariant patterns. By utilizing the disentangled patterns, we design a spatio-temporal intervention mechanism to create multiple interventional distributions and an environment inference module to infer the latent spatio-temporal environments, and minimize the invariance loss to leverage the invariant patterns with stable predictive abilities under distribution shifts. Extensive experiments demonstrate the superiority of our method over state-of-the-art sequential recommendation baselines under distribution shifts.

CCS Concepts: • **Information systems → Recommender systems**.

Additional Key Words and Phrases: Recommender systems, Sequential Recommendation, Graph Machine Learning, Dynamic Graph Neural Network, Out-Of-Distribution Generalization

ACM Reference Format:
Zeyang Zhang, Xin Wang*, Haibo Chen, Haoyang Li, and Wenwu Zhu*. 2024. Disentangled Dynamic Graph Attention Network for Out-of-distribution Sequential Recommendation. *ACM Transactions on Information Systems* 1, 1 (October 2024), 41 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Correspondence should be addressed to these authors.
Authors' addresses: Zeyang Zhang, Department of Computer Science and Technology, Tsinghua University, China, zy-zhang20@mails.tsinghua.edu.cn; Xin Wang*, Department of Computer Science and Technology, BNRist, Tsinghua University, China, xin_wang@tsinghua.edu.cn; Haibo Chen, Department of Computer Science and Technology, Tsinghua University, China, chb24@mails.tsinghua.edu.cn; Haoyang Li, Department of Computer Science and Technology, Tsinghua University, China, lihy18@mails.tsinghua.edu.cn; Wenwu Zhu*, Department of Computer Science and Technology, BNRist, Tsinghua University, China, wwzhu@tsinghua.edu.cn.

1046-8188/2024/10-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As the internet's vast information continues to expand dramatically, the need for recommender systems has become paramount in alleviating information overload across online platforms such as e-commerce, search engines, and social media [63, 140, 159]. While conventional collaborative filtering methods primarily focus on static user-item interactions, there is a recent surge of research attention towards harnessing the dynamic nature of user preferences, which recognizes that user tastes evolve over time, influenced by their historical interactions. Consequently, there's a growing interest in sequential recommendation techniques that exploit users' interaction histories to deliver more accurate predictions.

Sequential recommendation [41] involves predicting the next item a user will interact with based on their historical interactions. Unlike traditional recommendation approaches that focus solely on predicting users' preferences for individual items, sequential recommendation systems take into account the sequential nature of user-item interactions, acknowledging that user preferences evolve over time. By leveraging sequential patterns and historical interactions, these systems aim to provide more accurate and personalized recommendations, anticipating users' preferences not only based on their current interests but also on their past behaviors. With applications ranging from e-commerce platforms to content streaming services, sequential recommendation plays a crucial role in enhancing user engagement, satisfaction, and overall experience by delivering timely and relevant suggestions tailored to users' evolving preferences and needs.

Dynamic graph neural networks (DyGNNs) [9, 46, 96, 112, 162, 177], with their adeptness in capturing temporal dependencies within user-item interaction histories, hold significant promise for modeling sequential recommendation tasks and providing superior recommendations. Distinct from static graphs, dynamic graphs can represent temporal structure and feature patterns, which are more complex yet common in reality. By discovering temporal patterns effectively, they can offer nuanced insights into user preferences and behaviors, thereby enhancing recommendation accuracy and relevance for more personalized and better user experiences.

Nevertheless, spatio-temporal distribution shifts naturally exist in sequential recommendation for various reasons such as survivorship bias [13], selection bias [10, 176], trending [65], *etc.* Users' interests may differ from different communities [62],*i.e.*, distributions may shift among users and items in the spatial dimension. For example, for visits related to anime-themed clothing, anime enthusiasts might be interested in the clothing because of their interest in a particular character from the anime. Consequently, their next purchase might also be related to merchandise featuring that character. On the other hand, non-anime enthusiasts might focus more on the comfort and appearance of the clothing and would likely continue purchasing other lifestyle products. Users' interests over items may shift through time in the user-item temporal sequences [131],*i.e.*, distributions may also shift among items sequences in the temporal dimension. For example, in summer, users might purchase shorts after buying a T-shirt to stay cool, whereas in winter, they may opt for sweaters following a T-shirt purchase to keep warm. In real-world sequential recommendation, the distribution shift could be in both spatial and temporal dimensions, leading to more complex spatio-temporal distribution shifts. 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 in sequential recommendation.

To address this issue, in this paper, we study the problem of handling spatio-temporal distribution shifts in sequential recommendation 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 sequential recommendation, 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 method named Disentangled Intervention-based Dynamic Graph Attention Networks with Invariance Promotion (**I-DIDA**). Our proposed method handles distribution shifts well by discovering and utilizing invariant spatio-temporal patterns with stable predictive abilities in sequential recommendation. 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 [45, 100], 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. Inspired by invariant learning literature, we further learn invariant patterns across environments to promote invariance under distribution shifts. However, the environments on dynamic graphs are complex and usually unlabeled. Thus, we leverage variant patterns to enhance the invariance properties of the captured invariant patterns in the training process, by inferring the latent spatio-temporal environments and minimizing the prediction variance among these environments. 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 four real-world datasets, including node classification and link prediction tasks, demonstrate the superiority of our proposed method over state-of-the-art baselines under distribution shifts. We also conduct experiments on various real-world sequential recommendation datasets. This involves constructing dynamic graphs from the sequences of user-item interactions and modeling sequential recommendation tasks as dynamic graph link prediction problems. Our findings demonstrate that our approach significantly outperforms state-of-the-art sequential recommendation benchmarks. This superiority stems from our ability to harness spatio-temporal information within user-item interaction histories and adeptly address shifts in spatio-temporal distributions. The contributions of our work are summarized as follows:

- We propose Disentangled Intervention-based Dynamic Graph Attention Networks with Invariance Promotion (**I-DIDA**), which can handle spatio-temporal distribution shifts in sequential recommendation.
- 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.
- We further promote the invariance property by minimizing the prediction variance among the latent environments inferred by the variant patterns.
- Experiments on one synthetic dataset and several real-world datasets show that our method significantly improves over state-of-the-art dynamic GNN and OOD generalization baselines, showing our method's ability of handling spatio-temporal distribution shifts on dynamic graphs.

- Experiments on several real-world sequential recommendation datasets demonstrate the superiority of our method over state-of-the-art sequential recommendation baselines, showing that our method is able to leverage the spatio-temporal information in user-item interaction history and to effectively handle the spatio-temporal distribution shifts in sequential recommendation.

This manuscript is an extension of our paper published at NeurIPS 2022 [169]. Compared with the conference version, we make significant contributions from the following aspects:

- The newly proposed **I-DIDA** model is able to learn invariant patterns on dynamic graphs via enforcing sample-level and environment-level prediction invariance among the latent spatio-temporal patterns so as to improve the generalization ability of dynamic graph neural networks under spatio-temporal distribution shifts.
- The newly proposed environment-level invariance regularization can inherently boost the invariance property of the invariant patterns in the training process without adding extra time and memory complexity.
- **I-DIDA** jointly integrates spatio-temporal intervention mechanism and environment inference into a unified framework, so that the model can focus on invariant patterns to make predictions.
- More extensive experiments demonstrate that **I-DIDA** is able to show significant improvements over the state-of-the-art baseline methods and the original model proposed in the earlier conference paper.
- We further conduct experiments on several real-world sequential recommendation datasets by constructing dynamic graphs from the user-item interaction sequences and modeling the sequential recommendation tasks as dynamic graph link prediction problems. The results show that our method achieves significantly better performance than state-of-the-art sequential recommendation baselines by leveraging the spatio-temporal information in user-item interaction history and effectively handling the spatio-temporal distribution shifts in sequential recommendation.

The rest of this paper is organized as follows. We introduce the problem formulation and notations in Section 2. In Section 3, we describe the details of our proposed framework. We present the experimental results on dynamic graphs in Section 4 and the results on sequential recommendations in Section 5. We review the related work in Section 6. Finally, we conclude our work in Section 7.

2 PROBLEM FORMULATION AND NOTATIONS

In this section, we introduce the dynamic graph and prediction tasks, and formulate the problem of spatio-temporal distribution shift in dynamic graphs. The notations adopted in this paper are summarized in Table 1.

2.1 Dynamic Graph

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, $\mathcal{V} = \bigcup_{t=1}^T \mathcal{V}^t$, $\mathcal{E} = \bigcup_{t=1}^T \mathcal{E}^t$. We use $\mathbf{G}^t$ to denote a random variable of $\mathcal{G}^t$.

2.2 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 [112, 177]. Following [59, 144], we factorize the distribution

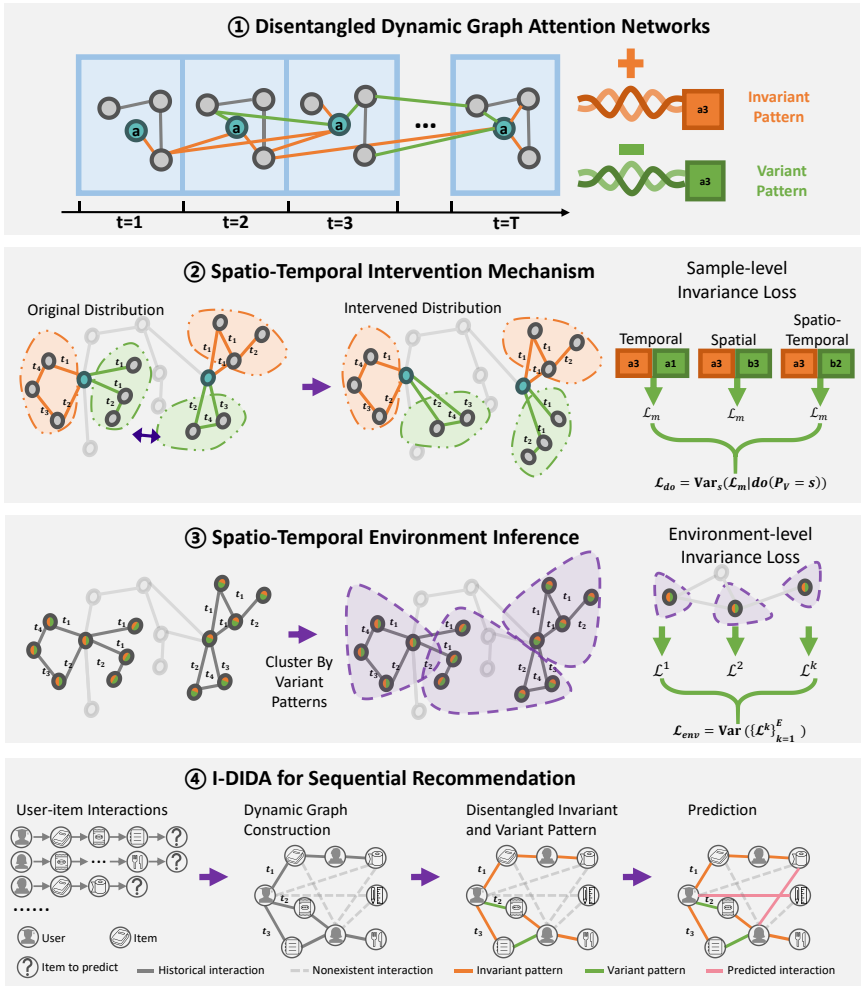


Fig. 1. The framework of our proposed method **I-DIDA**: 1. (Part ①) For a given dynamic graph with multiple timestamps, the disentangled dynamic graph attention networks first obtain summarizations of high-order invariant and variant patterns by disentangled spatio-temporal message passing. 2. (Part ②) Then the spatio-temporal intervention mechanism creates multiple intervened distributions by sampling and reassembling variant patterns across space and time for each node. By utilizing the samples from the intervened distributions, the sample-level invariance loss is calculated to optimize the model so that it can focus on invariant patterns to make predictions. 3. (Part ③) Finally, the spatio-temporal environment inference module infers the environments by clustering the variant patterns, and an environment-level invariance loss is proposed to promote the invariance of the invariant patterns. In this way, the method can make predictions based on the invariant spatio-temporal patterns which have stable predictive abilities across distributions, and therefore handle the problem of distribution shifts on dynamic graphs. 4. (Part ④) We apply **I-DIDA** to sequential recommendation tasks by first constructing the dynamic graphs from the user-item interaction sequences and extract the invariant patterns on dynamic graphs to make recommendations for the target users.

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 composed of the adjacency matrix including all edges in node v's

Table 1. The summary of notations.

Notations	Descriptions
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A graph with the node set and the edge set
$\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$	Graph slice at time t
$\mathcal{G}^{1:t}, Y^t, \mathbf{G}^{1:t}, \mathbf{Y}^t$	The graph trajectory, label and their corresponding random variable across times
$\mathcal{G}_v^{1:t}, y^t, \mathbf{G}_v^{1:t}, \mathbf{y}^t$	Ego-graph trajectory, the node's label and their corresponding random variable
$f(\cdot), g(\cdot)$	The predictor functions
$P, \mathbf{P}$	A pattern and its corresponding random variable
$m(\cdot)$	A function to select structures and features from ego-graph trajectories
$\mathrm{do}(\cdot)$	The do-calculus in causal inference
$\phi(\cdot)$	A function to find invariant patterns
d	The dimensionality of node representation
$\mathbf{q}, \mathbf{k}, \mathbf{v}$	The query, key, and value vector
$\mathcal{N}^t(u)$	The dynamic neighborhood of node u at time t
$\mathbf{m}_I, \mathbf{m}_V, \mathbf{m}_f$	The structural mask of invariant and variant patterns, and the featural mask
$\mathbf{z}_I^t(u), \mathbf{z}_V^t(u)$	Summarizations of invariant and variant patterns for node u at time t
$\mathrm{Agg}_I(\cdot), \mathrm{Agg}_V(\cdot)$	Aggregation functions for invariant and variant patterns
$\mathbf{h}_u^t$	Hidden embeddings for node u at time t
ℓ	The loss function
$\mathcal{L}, \mathcal{L}_m, \mathcal{L}_{do}$	The task loss, mixed loss, sample-level invariance loss
$\mathcal{L}^k, \mathcal{L}^{env}$	The k-th environment loss and the environment-level invariance loss
K	The number of environments
$\mathcal{K}, k(u^t)$	The environment set and the environment for the node u at time t.
$\mathcal{D} = \{(u, i, t)\}$	the user-item interaction sequences
$\mathcal{U}$	User set
$\mathcal{I}$	Item set
$\mathcal{S}^u$: $(\mathcal{S}_1^u, \mathcal{S}_2^u, \ldots, \mathcal{S}_{\lvert\mathcal{S}^u\rvert}^u)$	Item sequence for user u
T^u: $(T_1^u, T_2^u, \ldots, T_{\lvert\mathcal{S}^u\rvert}^u)$	Time sequence for user u

L-hop neighbors at time t, *i.e.*, $\mathcal{N}_v^t$, and 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.

2.3 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 graphs, researchers are devoted to capturing laws of network dynamics which are stable in systems [57, 102, 123, 138, 175]. 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 the temporal distribution shift which naturally exists in time-varying data [37, 43, 65, 88, 126] and the structural distribution shift in non-euclidean data [35, 144, 146], 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 METHODOLOGIES

In this section, we introduce our Disentangled Intervention-based Dynamic Graph Attention Networks with Invariance Promotion (**I-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, coupled with an invariance loss to minimize the prediction variance among intervened distributions. Finally, we propose an environmental invariance regularization to promote the quality of invariant patterns, and optimize the model with both invariance regularizations to encourage the model to rely on invariant patterns to make predictions.

3.1 Handling Spatio-Temporal Distribution Shift

3.1.1 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 [8, 69, 97, 178]. For example, the laws of triadic closure describe that two nodes with common neighbors (patterns) tend to have future interactions in social networks [31, 58, 175]. 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 [70]. 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 [175], the pattern can be explained as an open triad with similar neighborhood, and the model tends 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 [110]. 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 [45, 100], 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., $\mathbf{y}^t \perp \mathbf{P}_V^t(v) \mid \mathbf{P}_I^t(v)$.*

In the Assumption 1, $P_I^t(v) = \mathcal{G}_v^{1:t} \backslash P_V^t(v)$ denotes that the dynamic graph is composed of the invariant patterns and variant patterns. The assumption 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.

3.1.2 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. However, the objective is challenging due to 1) the invariant and variant patterns are

not labeled, and the model should be optimized to distinguish these patterns, 2) the properties of invariance and sufficiency should be achieved by specially designed mechanisms so that the model can rely on invariant patterns to make accurate predictions under distribution shifts. To this end, we propose two invariance loss from two levels for guiding the model to find and rely on invariant patterns, which are respectively inspired by the causal theory and invariant learning literature.

3.1.3 Sample-Level Invariance Loss. By causal theory [45, 100], Eq. (3) can be transformed into

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

where 'do' denotes do-calculas to intervene the original distribution [45, 121], $\mathcal{S}$ 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. As this loss intervenes the distributions in the sample-level (*i.e.*, nodes), and pursues the invariance of the invariant patterns for each sample, we name the variance term in Eq. (4) as sample-level invariance loss.

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.,* $p(\mathbf{y}^t \mid \tilde{\mathbf{P}}_I^t(v), \tilde{\mathbf{P}}_V^t(v)) = p(\mathbf{y}^t \mid \tilde{\mathbf{P}}_I^t(v))$.

3.1.4 Environment-Level Invariance Loss. Invariant learning [3, 71, 106] is a promising research direction with the goal of empowering the model with invariant predictive abilities under distribution shifts. Environments, commonly as a critical concept for the method assumption and design in the invariant learning literature, refer to where the observed instances are sampled from, which may have variant correlations with labels. In road networks, for example, two traffic jams in different places and times may happen simultaneously by chance or there can be causal relations, e.g., the road structure let one traffic jam block other roads and inevitably lead to another traffic jam. In this case, places and times may act as the environments which may have spurious correlations with labels and should not be exploited by the model under distribution shifts. Inspired by invariant learning, we propose to promote the invariance property of the invariant patterns by designing an environment-level invariance loss,

$$\mathrm{Var}_{k\in\mathcal{K}}(\mathbb{E}_{(y^t,\mathcal{G}_v^{1:t})\sim p_{tr}(\mathbf{y}^t,\mathbf{G}_v^{1:t}|k)} \mathcal{L}(f_{\theta_1}(\phi_{\theta_2}(\mathcal{G}_v^{1:t})), y^t)), \tag{5}$$

where k denotes the k-th environment from the environment set $\mathcal{K}$, and $p_{tr}(\mathbf{y}^t, \mathbf{G}_v^{1:t}|k)$ denotes the data distribution of the k-th environment. Intuitively, minimizing the environment-level invariance loss encourages the model to make stable predictions regardless of the environments.

Together with the sample-level invariance loss and environment-level invariance loss, we can help the model discover the invariant and variant patterns, and rely on invariant patterns to make predictions. We will describe how to implement these insights in an end-to-end manner in the following sections.

3.2 Disentangled Dynamic Graph Attention Networks

3.2.1 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. For node u at time t_1, the dynamic neighborhoods $\mathcal{N}^t(u), t \le t_1$ describe the historical structural information of u^t, which enables different views of historical structural information based on the current time, e.g., u^{t_2} and u^{t_3} may aggregate different messages from $\mathcal{N}^{t_1}(u)$ for $t_1 \le t_2 \le t_3$. For example, the interest of the same user may have evolved

through time, and the messages, even from the same neighborhood, adopted by the user to conduct transactions also vary. The model should be designed to be aware of these evolving patterns in the dynamic neighborhood. Note that the defined dynamic neighborhood includes only 1-order spatial neighbors at time t for the brevity of notations, while the concept of n-order neighbors can be extended by considering the neighbors which can be reached by n-hop paths. Following classical message passing networks, we take into consideration the information of the n-order neighborhood by stacking multiple layers for message passing and aggregation.

3.2.2 Disentangled Spatio-temporal Graph Attention Layer. To capture spatio-temporal patterns 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

$$\begin{aligned} \mathbf{q}_u^t &= \mathbf{W}_q\Big(\mathbf{h}_u^t || \mathrm{TE}(t)\Big), \\ \mathbf{k}_v^{t'} &= \mathbf{W}_k\Big(\mathbf{h}_v^{t'} || \mathrm{TE}(t')\Big), \\ \mathbf{v}_v^{t'} &= \mathbf{W}_v\Big(\mathbf{h}_v^{t'} || \mathrm{TE}(t')\Big), \end{aligned} \tag{6}$$

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. For simplicity of notations, the vectors in this paper are represented as row vectors. $\mathrm{TE}(t)$ denotes the temporal encoding techniques to obtain embeddings of time t so that the time of link occurrence can be considered inherently [105, 150]. Then, we can calculate the attention scores among nodes in the dynamic neighborhood to obtain the structural masks,

$$\begin{aligned} \mathbf{m}_I &= \mathrm{Softmax}(\frac{\mathbf{q} \cdot \mathbf{k}^T}{\sqrt{d}}), \\ \mathbf{m}_V &= \mathrm{Softmax}(-\frac{\mathbf{q} \cdot \mathbf{k}^T}{\sqrt{d}}), \end{aligned} \tag{7}$$

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 $\mathbf{m}_f = \mathrm{Softmax}(\mathbf{w}_f)$ to select features from the messages of dynamic neighbors. Then the messages of the dynamic neighborhood can be summarized with respective masks,

$$\begin{aligned} \mathbf{z}_I^t(u) &= \mathrm{Agg}_I(\mathbf{m}_I, \mathbf{v} \odot \mathbf{m}_f), \\ \mathbf{z}_V^t(u) &= \mathrm{Agg}_V(\mathbf{m}_V, \mathbf{v}), \end{aligned} \tag{8}$$

where $\mathrm{Agg}(\cdot)$ denotes aggregating and summarizing messages from the dynamic neighborhood. To further disentangle the invariant and variant patterns, we design different aggregation functions $\mathrm{Agg}_I(\cdot)$ and $\mathrm{Agg}_V(\cdot)$ 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,

$$\mathbf{h}_u^t \leftarrow \mathbf{z}_I^t(u) + \mathbf{z}_V^t(u). \tag{9}$$

3.2.3 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 [124] to stable the training process and model multi-faceted graph evolution [107].

3.3 Spatio-Temporal Intervention Mechanism

3.3.1 Direct Intervention. One way of intervening the distribution of the variant pattern 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.

3.3.2 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

$$\mathbf{z}_I^{t_1}(u), \mathbf{z}_V^{t_1}(u) \leftarrow \mathbf{z}_I^{t_1}(u), \mathbf{z}_V^{t_2}(v). \tag{10}$$

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. (10) multiple times, we can obtain multiple intervened data distributions for the subsequent optimization.

3.4 Spatio-Temporal Environment Inference

It is challenging to obtain environment labels on dynamic graphs, since the environments on dynamic graphs are complex that include spatio-temporal information and may also vary by periods or communities. For these reasons, environment labels are not available on dynamic graphs in practice. To tackle this problem, we introduce the spatio-temporal environment inference module in this section.

Recall that in Sec. 3.2, we obtain the summarized invariant and variant spatio-temporal patterns $\mathbf{z}_I^t$ and $\mathbf{z}_V^t$, which can be further exploited to infer the environment labels $k(u^t)$ for each node u at time t. Since the invariant patterns capture the invariant relationships between predictive ego-graph trajectories and labels, the variant patterns in turn capture variant correlations under different distributions, which could be helpful for discriminating spatio-temporal environments. Inspired by [79, 84], we utilize the variant patterns to infer the latent environments. Specifically, to infer the node environment labels $\mathbf{K} \in \mathcal{K}^{N\times T}$, we adopt an off-the-shelf clustering algorithm K-means in this paper, while other more sophisticated clustering methods can be easily incorporated,

$$\mathbf{K} = \text{K-means}([\mathbf{z}_V^1, \mathbf{z}_V^2, \ldots, \mathbf{z}_V^T]), \tag{11}$$

where $k(u^t) \in \mathcal{K}$ denote the corresponding environment label for each node u at time t, $\mathcal{K}$={0,1,...,K} denotes the set of K environments, and K is a hyperparameter that reflects the assumption of the number of the environments. Using $\mathbf{K}$, we can partition the nodes at different time on dynamic graphs into multiple training environments. Note that the spatio-temporal environment inference module is unsupervised without any ground-truth environment labels, which is more practical on real-world dynamic graphs.

3.5 Optimization with Invariance Loss

3.5.1 Sample-Level 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

$$\mathcal{L} = \ell(f(\mathbf{z}_I), \mathbf{y}), \tag{12}$$

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

$$\mathcal{L}_m = \ell(g(\mathbf{z}_V, \mathbf{z}_I), \mathbf{y}), \tag{13}$$

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

$$\mathcal{L}_{do} = \mathrm{Var}_{s_i \in \mathcal{S}}(\mathcal{L}_m | \mathrm{do}(\mathbf{P}_V^t = s_i)), \tag{14}$$

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.

3.5.2 Environment-Level Invariance Loss. After obtaining the environment labels by the spatio-temporal environment inference module in Sec. 3.4, we have the samples from different environments and the loss of the k-th environment is calculated by

$$\mathcal{L}^k = \ell(f(\{\mathbf{z}_I^t(u) : k(u^t) = k\}, \mathbf{y}), \tag{15}$$

and the environment-level invariance loss can be calculated by

$$\mathcal{L}_{env} = \mathrm{Var}(\{\mathcal{L}^k\}_{k=1}^K). \tag{16}$$

In this way, minimizing the variance term encourages the invariance of the model predictions among different environments, which potentially reduces the effects of spurious correlations that may be caused by the spatio-temporal environments under distribution shifts.

3.5.3 Overall Training Objective. The final training objective is

$$\min_{\theta} \mathcal{L} + \lambda_{do}\mathcal{L}_{do} + \lambda_e \mathcal{L}_{env}, \tag{17}$$

where the task loss $\mathcal{L}$ is minimized to exploit invariant patterns, while the sample-level invariance loss $\mathcal{L}_{do}$ and environment-level invariance loss $\mathcal{L}_{env}$ help the model to discover invariant and variant patterns, and λ_{do} and λ_e are hyperparameters 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 Algorithm 1.

3.6 Discussions

3.6.1 Complexity Analysis. We analyze the computational complexity of **I-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. Denote K as the number of environments, T as the number of iterations for the K-means algorithm. Our intervention mechanism has a time complexity of $O(|E_p||S|d)$ and the environment inference module has a time complexity of $O(K|V|Td)$ in training.

Algorithm 1 Training pipeline for **I-DIDA**

Require: Training epochs L, number of intervention samples S, number of environments K, hyperparameters λ_{do} and λ_e.

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. (12) and Eq. (13)
  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. (13)
  end for
  Calculate the sample-level invariance loss as Eq. (14)
  Infer the environment labels as Eq. (11)
  for $k = 1, \ldots, K$ **do**
    Calculate the k-th environment loss as Eq. (15)
  end for
  Calculate the environment-level invariance loss as Eq. (16)
  Update the model according to Eq. (17)
end for

Moreover, these modules do not put extra time complexity in inference, since they are only adopted in the training state.

Therefore, the overall time complexity of **I-DIDA** is $O(|E|d + |V|d^2 + |E_p||S|d + K|V|Td)$. Notice that $|S|$ is a hyper-parameter and is usually set as a small constant. In summary, **I-DIDA** has a linear time complexity with respect to the number of nodes and edges, which is on par with the existing dynamic GNNs.

3.6.2 Background of Assumption 1. It is widely adopted in out-of-distribution generalization literature [1, 3, 18, 43, 95, 104, 146] 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) [45, 100], $\mathbf{P}_V \rightarrow \mathbf{G} \leftarrow \mathbf{P}_I \rightarrow \mathbf{y}$ 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 [45] $\mathbf{P}_V \leftarrow \mathbf{P}_I \rightarrow \mathbf{y}$ 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.

3.6.3 Connections 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 the variant patterns. By applying do-calculus on one variable, all in-coming arrows(causal relationship) to it will be removed [45] and the intervened distributions will be created. In our case, the operator 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.

In this paper, we refer **I-DIDA** as our method Disentangled Intervention-based Dynamic Graph Attention Networks with Invariance Promotion, and DIDA as a special case where $\lambda_e = 0$.

3.7 Application to Sequential Recommendation

In this section, we apply **I-DIDA** to the sequential recommendation task. Note that in the context of sequential recommendation, the invariant patterns in the assumption 1 refer to the user interests, including both the recent and long-term interests, which determine the next item bought by the users. Models may exploit variant patterns, like popular items for specific communities or periods, to make predictions, while neglecting the user interests, and thus have deteriorated performance under distribution shifts. We first introduce the problem setting and then describe the dynamic graph construction and algorithm pipeline for sequential recommendation.

3.7.1 Sequential Recommendation Dynamic Graph Construction. Suppose we have the user-item interaction sequences $\mathcal{D} = \{(u, i, t)\}$, where $u \in \mathcal{U}$ denotes the user, $i \in \mathcal{I}$ denotes the item, and t denotes the discrete timestamp which means the year of the interaction in the dataset. We construct the dynamic graph $\mathcal{G}^{1:T}$ as follows. Denote $u \in \mathcal{U}$ to be the user node $v_i, i \in 0, \ldots, |\mathcal{U}| - 1$, and $i \in \mathcal{I}$ to be the item node $v_{|\mathcal{U}|+i}, i \in 0, \ldots, |\mathcal{I}| - 1$. For each time t, we construct the static graph $\mathcal{G}^t$ by using the user-item pairs (u, i, t) in the interaction sequence $\mathcal{D}$. Then we stack the static graphs $\mathcal{G}^t$ for $t = 1, 2, \ldots, T$ to obtain the dynamic graph $\mathcal{G}^{1:T}$. For the datasets without node features, we randomly initialize the user and item node features, which does not introduce any additional information compared to the baselines.

3.7.2 Algorithm pipeline for Sequential Recommendation. For sequential recommendation tasks, modeling the historical item sequence for each user holds great significance. The original **I-DIDA** emphasizes discovering invariant patterns on the dynamic graphs of user-item interactions. It implicitly considers the historical item sequence through multiple hops of user-item interactions rather than explicitly modeling. In implementing **I-DIDA** for sequential recommendation tasks, we enhance **I-DIDA** to explicitly model historical item sequence. We achieve this by using self-attention to directly learn weights and aggregate information among items. For simplicity, we adopt the self-attention from SASRec to explicitly model historical item sequence, while this module can be extended to other similar models. To this end, our training for sequential recommendation tasks can be divided into two stages, where in the first stage, we adopt **I-DIDA** to obtain the invariant patterns over user-item interactions, and in the second stage, we adopt SASRec to model item sequence to predict the next item for each user.

Next, we introduce the implementation details of each stage. In the first stage, we train the **I-DIDA** model to obtain the invariant and variant patterns for each user and item at different

times $\mathbf{z}_V, \mathbf{z}_I \in \mathbb{R}^{T\times|V|\times d}$, where T is the number of timestamps, $|V|$ is the number of nodes, and d is the dimensionality of the hidden representation. In the second stage, we utilize the self-attention blocks in SASRec to model the historical item sequences and calculate the item embedding $\mathbf{z}_{sas} \in \mathbb{R}^{|I|\times d}$. For each user u, we have the item sequence $\mathcal{S}^u$: $(\mathcal{S}^u_1, \mathcal{S}^u_2, \ldots, \mathcal{S}^u_{|\mathcal{S}^u|})$ and corresponding time sequence T^u: $(T^u_1, T^u_2, \ldots, T^u_{|\mathcal{S}^u|})$ representing the time when the user interacts with items. Denote the next item to be predicted $\mathcal{S}^u_j$ as i, where $j = 1, 2, \ldots, |\mathcal{S}^u|$. We merge $\mathbf{z}_{sas}(i)$ for item i with the item's corresponding invariant pattern $\mathbf{z}^t_I(i)$, where $t = T^u_i$, *i.e.*,

$$\mathbf{z}_{mer}(i) = \mathbf{F}_{item}(\mathbf{z}_{sas}(i)||\mathbf{z}^t_I(i)), \tag{18}$$

where $\mathbf{F}_{item}$ is a linear layer, $||$ is a concatenation operation and $\mathbf{z}_{mer}(i)$ is the merged item embedding considering both user-item interactions and historical item sequence. We also obtain one part of user embeddings via summarizing the historical item sequences, *i.e.*,

$$z^u_j = f(\mathbf{z}_{sas}(\mathcal{S}^u_1), \mathbf{z}_{sas}(\mathcal{S}^u_2), \ldots, \mathbf{z}_{sas}(\mathcal{S}^u_{j-1})), \tag{19}$$

where $f(\cdot)$ is a self-attention mechanism. We merge $\mathbf{z}^u_j$ with the user's corresponding invariant pattern $\mathbf{z}^t_I(u)$, where $t = T^u_i$, *i.e.*,

$$\mathbf{z}_{mer}(u) = \mathbf{F}_{user}(\mathbf{z}^u_j||\mathbf{z}^t_I(u)), \tag{20}$$

where $\mathbf{F}_{user}$ is a linear layer and $\mathbf{z}_{mer}(u)$ is the merged user embedding considering both user-item interactions and historical item sequence. We then use the merged user embedding $\mathbf{z}_{mer}(u)$ and the merged item embedding $\mathbf{z}_{mer}(i)$ to calculate the task loss, *i.e.*,

$$\mathcal{L} = \ell(g(\mathbf{z}_{mer}(u), \mathbf{z}_{mer}(i)), \mathbf{y}), \tag{21}$$

where g is the linear predictor to predict the interactions between users and items, and ℓ is a cross-entropy loss function. The overall training pipeline for sequential recommendation tasks is summarized in Algorithm 2.

3.7.3 Complexity Analysis. The time complexity of the first stage is $O(|E|d + |V|d^2 + |E_p||S|d + K|V|Td)$, which is the same as the overall time complexity of **I-DIDA**. Here, $|E|$ represents the number of interactions in the user-item dynamic graph, $|V|$ is the sum of user and item nodes, d denotes the dimensionality of the hidden representation, $|E_p|$ is the number of edges to predict, $|S|$ is the size of the intervention set, K is the number of environments, and T is the number of iterations for the K-means algorithm. The time complexity of the second stage is $O(|\mathcal{U}|NL^2d + |\mathcal{U}|Ld^2)$. The former term, $O(|\mathcal{U}|NL^2d)$, is the time complexity of the self-attention mechanism, where $|\mathcal{U}|$ is the number of users, N is the number of layers of self-attention blocks, L is the sequence length, and d is the dimensionality of the hidden representation. The latter term, $O(|\mathcal{U}|Ld^2)$, is the time complexity of the feed-forward network. The overall time complexity of the training pipeline on the sequential recommendation task is $O(|E|d + |V|d^2 + |E_p||S|d + K|V|Td + |\mathcal{U}|NL^2d + |\mathcal{U}|Ld^2)$, which is comparable to the baselines.

4 EXPERIMENTS ON DYNAMIC GRAPHS

In this section, we conduct extensive experiments to verify that our framework can handle spatio-temporal distribution shifts by discovering and utilizing invariant patterns.

4.1 Baselines

We adopt several representative GNNs and Out-of-Distribution (OOD) generalization methods as our baselines. The first group of these methods is static GNNs, including:

Algorithm 2 Training pipeline for **I-DIDA**on sequential recommendation tasks

Require: Training epochs of **I-DIDA** L_D, training epochs of SASRec L_S, iteration number L_I, number of intervention samples S, number of environments K, hyperparameters λ_{do} and λ_e.

for $l = 1, \ldots, L_D$ **do**
  Obtain $\mathbf{z}_V^t, \mathbf{z}_I^t$ for each user and item at different times from **I-DIDA**
  Calculate sample-level invariance loss and environment-level invariance loss as in **Algorithm 1**
  Update the **I-DIDA** model
end for
Obtain $\mathbf{z}_I \in \mathbb{R}^{|\mathcal{I}| \times d}$ for each user and item at different times from **I-DIDA**
for $l = 1, \ldots, L_S$ **do**
  for $i = 1, \ldots, L_I$ **do**
    Sample a mini-batch from the training dataset
    Obtain $\mathbf{z}^{sas} \in \mathbb{R}^{|\mathcal{I}| \times d}$ for each item from SASRec
    Merge $\mathbf{z}^{sas}(i)$ in items sequence of user u with $\mathbf{z}_I^t(i)$ for each item $i = \mathcal{S}_j^u, j = 1, 2, \ldots, |\mathcal{S}^u|$ in the sequence at time T_i^u to obtain $z_{mer}(i)$
    Calculate user embedding $\mathbf{z}_j^u = f(\mathbf{z}^{sas}(\mathcal{S}_1^u), \mathbf{z}^{sas}(\mathcal{S}_2^u), \ldots, \mathbf{z}^{sas}(\mathcal{S}_{j-1}^u))$ for each user u
    Merge $\mathbf{z}_j^u$ with $\mathbf{z}_I^t(u)$ for each user u at time T_i^u to obtain $z_{mer}(u)$
    Calculate task loss as Eq.21
    Update the SASRec
  end for
end for

- **GAE** [67] is a representative static graph neural network with a stack of graph convolutions to capture the information of structures and attributes on graphs.
- **VGAE** [67] further introduces variational variables into GAE to obtain more robust and generalized graph representations.

The second group of these methods includes the following dynamic GNNs:

- **GCRN** [108] is a representative dynamic GNN that first adopts a GCN[67] to obtain node embeddings and then a GRU [30] to model the network evolution.
- **EvolveGCN** [98] adopts an LSTM [54] or GRU [30] to flexibly evolve the GCN [67] parameters instead of directly learning the temporal node embeddings, which is applicable to frequent change of the node set on dynamic graphs.
- **DySAT** [107] aggregates neighborhood information at each graph snapshot using structural attention and models network dynamics with temporal self-attention so that the weights can be adaptively assigned for the messages from different neighbors in the aggregation.

And the third group of these methods consists of OOD generalization methods:

- **IRM** [3] aims at learning an invariant predictor which minimizes the empirical risks for all training domains to achieve out-of-distribution generalization.
- **GroupDRO** [106] puts more weight on training domains with larger errors when minimizing empirical risk to minimize worst-group risks across training domains.
- **VREx** [71] reduces differences in risk across training domains to reduce the model's sensitivity to distributional shifts.

These representative OOD generalization methods aim at improving the robustness and generalization ability of models against distribution shift, which requires explicit environment labels to calculate the loss. For fair comparisons, we randomly split the samples into different domains, as the

Table 2. Summarization of dataset statistics. Evolving features denote whether the node features vary through time. Unseen nodes denote whether the test nodes are partially or fully unseen in the past.

Dataset	COLLAB	Yelp	Synthetic	OGBN-Arxiv	Aminer
# Timestamps	16	24	16	20	17
# Nodes	23,035	13,095	23,035	168,195	43,141
# Links	151,790	65,375	151,790	3,127,274	851,527
Temporal Granularity	Year	Month	Year	Year	Year
Feature Dimension	32	32	64	128	128
Evolving Features	No	No	Yes	No	No
Unseen Nodes	Partial	Partial	Partial	Full	Full
Classification Tasks	Link	Link	Link	Node	Node

field information is unknown to all methods. Since they are general OOD generalization methods and are not specifically designed for dynamic graphs, we adopt the best-performed DyGNN on the training datasets as their backbones.

4.2 Real-world Link Prediction Datasets

4.2.1 Experimental Settings. We use two real-world dynamic graph datasets, including COLLAB and Yelp. 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. Here we briefly introduce the real-world datasets as follows:

- **COLLAB** [119][1] 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'. We use word2vec [93] to extract 32-dimensional features from paper abstracts and average to obtain author features. We use 10,1,5 chronological graph slices for training, validation and testing respectively. The dataset includes 23,035 nodes and 151,790 links in total.
- **Yelp** [107][2] is a business review dataset, containing customer reviews on the 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'. We use word2vec [93] to extract 32-dimensional features from reviews and averages to obtain user and business features. We select users and items with interactions of more than 10. We use 15, 1, 8

[1]https://www.aminer.cn/collaboration
[2]https://www.yelp.com/dataset

Table 3. Results (AUC%) of different methods on real-world link prediction 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 \ Dataset	**COLLAB**		**Yelp**	
Test Data	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
VGAE	86.47±0.04	74.95±1.25	76.54±0.50	65.33±1.43
GCRN	82.78±0.54	69.72±0.45	68.59±1.05	54.68±7.59
EGCN	86.62±0.95	76.15±0.91	78.21±0.03	53.82±2.06
DySAT	<u>88.77±0.23</u>	<u>76.59±0.20</u>	78.87±0.57	66.09±1.42
IRM	87.96±0.90	75.42±0.87	66.49±10.78	56.02±16.08
VREx	88.31±0.32	76.24±0.77	<u>79.04±0.16</u>	66.41±1.87
GroupDRO	88.76±0.12	76.33±0.29	**79.38±0.42**	<u>66.97±0.61</u>
DIDA	91.97±0.05	81.87±0.40	78.22±0.40	75.92±0.90
I-DIDA	**92.17±0.40**	**82.40±0.70**	78.17±0.76	**76.90±1.87**

chronological graph slices for training, validation and test respectively. The dataset includes 13,095 nodes and 65,375 links in total.

4.2.2 Experimental Results. Based on the results on real-world link prediction datasets in Table 3, 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', drops by nearly 12%, 12% and 5% in 'w/ DS'. In Yelp, 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. 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. **I-DIDA** improves significantly over all baselines in 'w/ DS' for all datasets. Note that Yelp has stronger temporal distribution shift since COVID-19 happens in the midway, strongly affecting consumers' behavior in business, while **I-DIDA** outperforms the most competitive baseline GroupDRO by 9% in 'w/ DS'. In comparison to similar field information in Yelp (all restaurants), COLLAB has stronger spatial distribution shift since the fields are more different to each other, while **I-DIDA** outperforms the most competitive baseline DySAT by 5% in 'w/ DS'.

4.3 Real-world Node Classification Datasets

4.3.1 Experimental Settings. We use 2 real-world dynamic graph datasets, including OGBN-Arxiv [56] and Aminer [111, 120]. The two datasets are both citation networks, where nodes represent papers,

Table 4. Results (ACC%) of different methods on real-world node classification datasets. The best results are in bold and the second-best results are underlined.

Model \ Dataset	**OGBN-Arxiv**			**Aminer**		
Split	2015-2016	2017-2018	2019-2020	2015	2016	2017
GRCN	46.77±2.03	45.89±3.41	46.61±3.29	47.96±1.12	<u>51.33±0.62</u>	<u>42.93±0.71</u>
EGCN	48.70±2.12	47.31±3.45	46.93±5.17	44.14±1.12	46.28±1.84	37.71±1.84
DySAT	48.83±1.07	47.24±1.24	46.87±1.37	48.41±0.81	49.76±0.96	42.39±0.62
IRM	<u>49.57±1.02</u>	<u>48.28±1.51</u>	46.76±3.52	48.44±0.13	50.18±0.73	42.40±0.27
VREx	48.21±2.44	46.09±4.13	46.60±5.02	48.70±0.73	49.24±0.27	42.59±0.37
GroupDRO	49.51±2.32	47.44±4.06	<u>47.10±4.39</u>	<u>48.73±0.61</u>	49.74±0.26	42.80±0.36
DIDA	51.46±1.25	49.98±2.04	50.91±2.88	50.34±0.81	51.43±0.27	44.69±0.06
I-DIDA	**51.53±1.22**	**50.44±1.83**	**51.87±2.01**	**51.12±0.33**	**52.35±0.82**	**45.09±0.23**

and edges from u to v with timestamp t denote the paper u published at year t cites the paper v. The node classification task on dynamic graphs is challenging since the nodes come in the future, e.g., new papers are published in the future, so that the model should exploit the spatio-temporal information to classify the nodes. Following [144], we also use the inductive learning settings, *i.e.*, the test nodes are strictly unseen during training, which is more practical and challenging in real-world dynamic graphs. Here, we briefly introduce the real-world datasets as follows.

- **OGBN-Arxiv** [56] is a citation network between all Computer Science (CS) arXiv papers indexed by MAG [129]. Each paper has a 128-dimensional feature vector obtained by averaging the embeddings of words in its title and abstract, where the embeddings of individual words are computed by running the skip-gram model [94] over the MAG corpus. The task is to predict the 40 subject areas of arXiv CS papers, e.g., cs.AI, cs.LG, and cs.OS. We train on papers published between 2001 - 2011, validate on those published in 2012-2014, and test on those published since 2015. With the volume of scientific publications doubling every 12 years over the past century, spatio-temporal distribution shifts naturally exist on these dynamic graphs. The dataset has 168,195 nodes and 3,127,274 links in total.
- **Aminer** [111, 120] is a citation network extracted from DBLP, ACM, MAG, and other sources. We use word2vec [93] to extract 128-dimensional features from paper abstracts and average to obtain paper features. We select the top 20 venues, and the task is to predict the venues of the papers. Similar to the OGBN-Arxiv dataset, we train on papers published between 2001 - 2011, validate on those published in 2012-2014, and test on those published since 2015. As the test nodes are not seen during training, the model is tested to exploit the invariant spatio-temporal patterns and make stable predictions under distribution shifts. The dataset has 43,141 nodes and 851,527 links in total.

4.3.2 Experimental Results. Based on the results on real-world node classification datasets in Table 4, we have the following observations:

- Most baselines have significant performance drops as time goes. On OGBN-Arxiv, for example, EGCN gradually drops from 48.70% to 46.93% from 2015 to 2020. This phenomenon may result from the spatio-temporal distribution shifts on dynamic graphs as time goes, e.g., there has been a significant increase in the quantity of academic papers being published, and topics as well as the citation patterns might be different from the past. Moreover, general out-of-distribution baselines have performance improvement over the DyGNN baselines,

while the improvements are far from satisfactory since they are not specially designed for handling the complex spatio-temporal distribution shifts on dynamic graphs.

- Our method significantly alleviates the performance drop as time goes. On OGBN-Arxiv, for example, **I-DIDA** has a performance improvement of 2%, 2%, 4% from 2015 to 2020 in comparisons with the best baselines, which verifies that our method can capture the invariant and variant spatio-temporal patterns inside data and exploit the invariant patterns to make predictions under distribution shifts. Moreover, our method has less variance in most cases, which may be due to that the sample-level and environment-level invariance loss can reduce the effects of the spurious correlations to obtain better performance under distribution shifts.

4.4 Synthetic Datasets

4.4.1 Experimental Settings. To evaluate the model's generalization ability under spatio-temporal distribution shift, following [144], 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 a 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-varying features, we omit their results.

Here, we detail the construction of variant features $\mathbf{X}_2^t$. 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 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 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.

4.4.2 Experimental Results. Based on the results on the synthetic dataset in Table. 5, 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 drops drastically in the test stage, which shows that the existing DyGNNs fail to handle distribution shifts. In terms of test results, **I-DIDA** consistently outperforms DyGNN baselines by a significantly large margin. In particular, **I-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, **I-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 the test stage. Instead,

Table 5. Results (AUC%) of different methods on the 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	<u>72.57±0.72</u>	74.71±0.17	<u>72.29±0.47</u>	75.69±0.07	<u>67.26±0.22</u>
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	<u>85.20±0.07</u>	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	<u>89.90±0.11</u>	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
I-DIDA	**88.50±0.46**	**85.27±0.06**	**92.27±1.02**	**83.00±1.08**	<u>94.23±0.23</u>	**74.87±1.59**

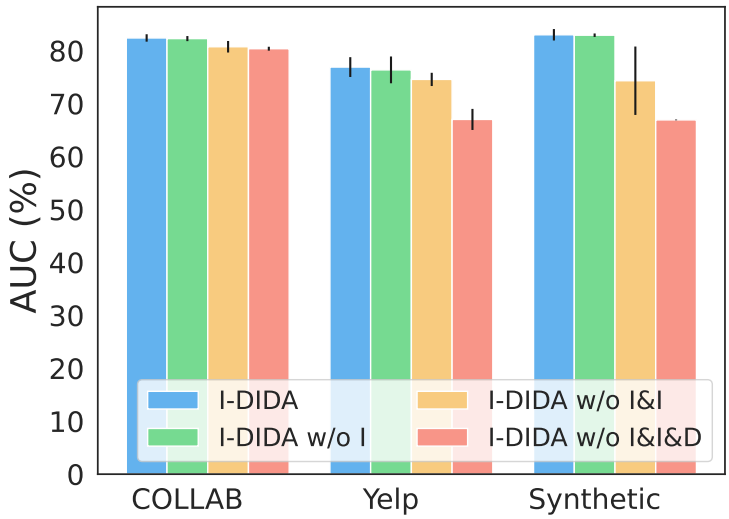
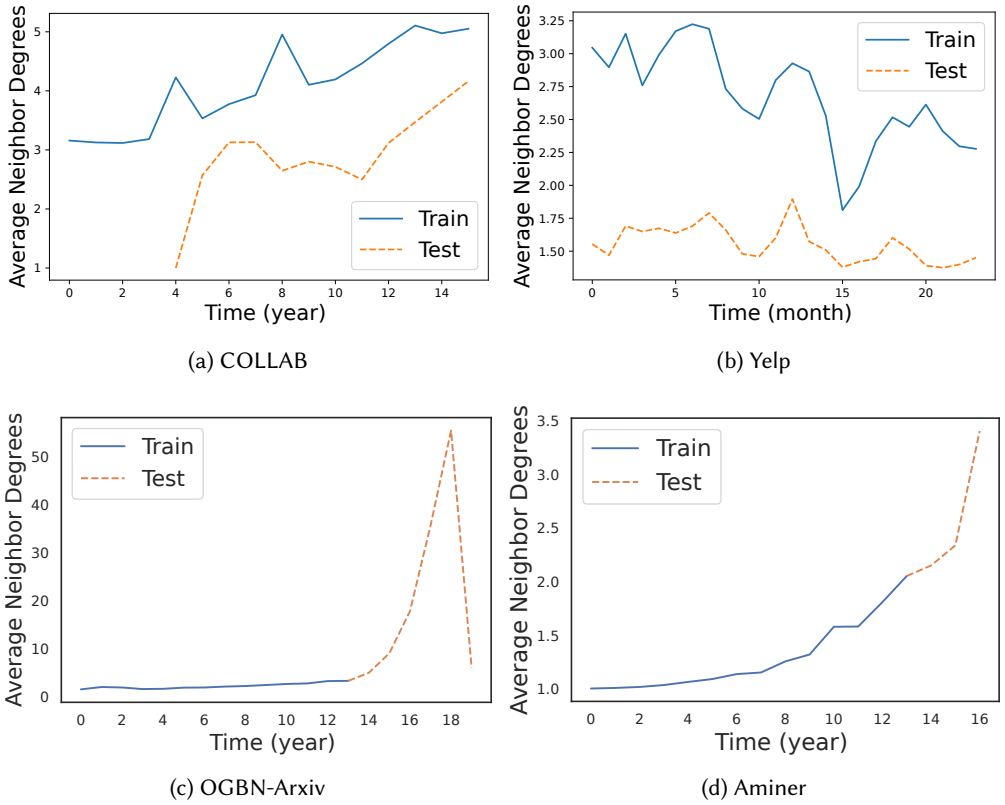


Fig. 2. Ablation studies on the environment inference, intervention mechanism and disentangled attention, where 'w/o I' removes the spatio-temporal environment inference module, 'w/o I&I' further removes the spatio-temporal intervention mechanism and 'w/o I&I&D' further removes disentangled attention. (Best viewed in color)

the rise in train results and drop in test results of **I-DIDA** are significantly lower than baselines, which demonstrates that **I-DIDA** can exploit invariant patterns and alleviate the harmful effects of variant patterns under distribution shift.

4.5 Ablation Studies

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



(a) COLLAB (b) Yelp

(c) OGBN-Arxiv (d) Aminer

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

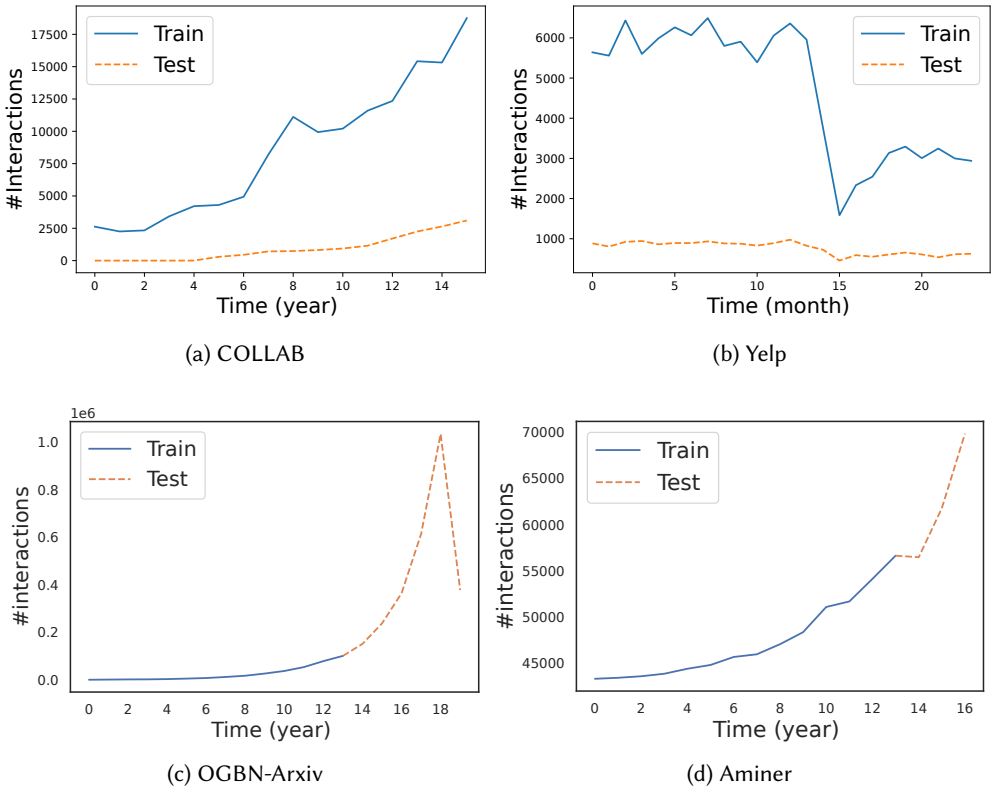
4.5.1 Spatio-Temporal Environment Inference. We remove the environment inference module mentioned in Sec. 3.4. From Figure 2, we can see that without the spatio-temporal environment inference module, the model has a performance drop especially in the Yelp dataset, which verifies that our environment-level invariance loss helps the model to promote the invariance properties of the invariant patterns.

4.5.2 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.

4.5.3 Disentangled Dynamic 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.

4.6 Additional Experiments

4.6.1 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 [6]. Figure 3 shows



(a) COLLAB (b) Yelp

(c) OGBN-Arxiv (d) Aminer

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

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 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. Similarly, Figure 3 and Figure 4 show that the number of links and the average neighbor degrees have a drastic increase in the test split on the Aminer and OGBN-Arxiv datasets, leading that the recent patterns on dynamic graphs might be significantly different from the past.

4.6.2 Spatial or Temporal Intervention. We compare two other versions of **I-DIDA** , where I-DIDA-S only uses spatial intervention and I-DIDA-T only uses temporal intervention. For I-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 Eq.(9) for I-DIDA-T. Figure 5a shows that **I-DIDA** improves significantly over the other two ablated versions,

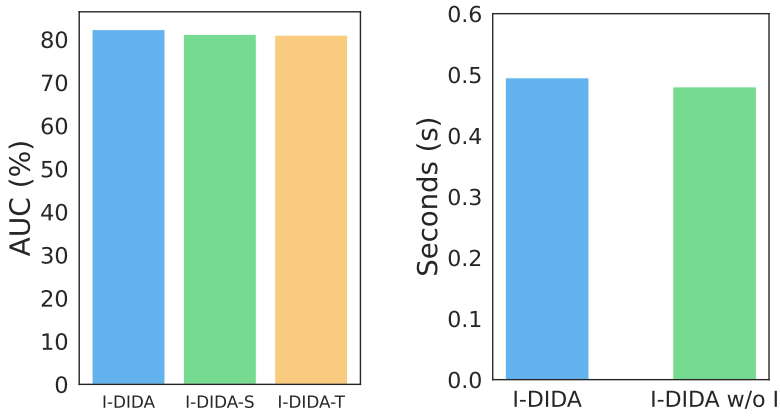


Fig. 5. (a) Comparison of different intervention mechansim on COLLAB dataset, where I-DIDA-S only uses spatial intervention and I-DIDA-T only uses temporal intervention. (b) Comparison in terms of training time for each epoch on COLLAB dataset, where 'w/o I' means removing intervention mechanism in **I-DIDA**. (Best viewed in color)

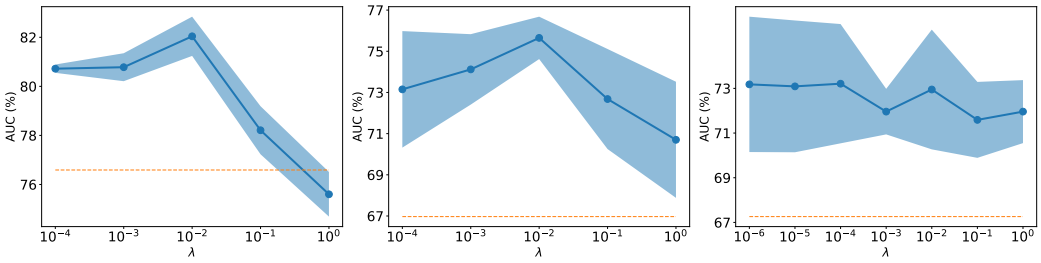
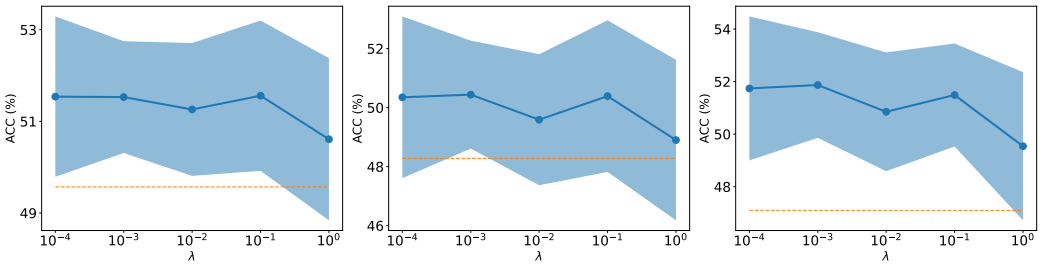


Fig. 6. Sensitivity of hyperparameter λ_{do} on different datasets. The area shows the average AUC and standard deviations in the test stage. The dashed line represents the average AUC of the best-performed baseline.

which verifies that it is important to take into consideration both the spatial and temporal aspects of distribution shifts.

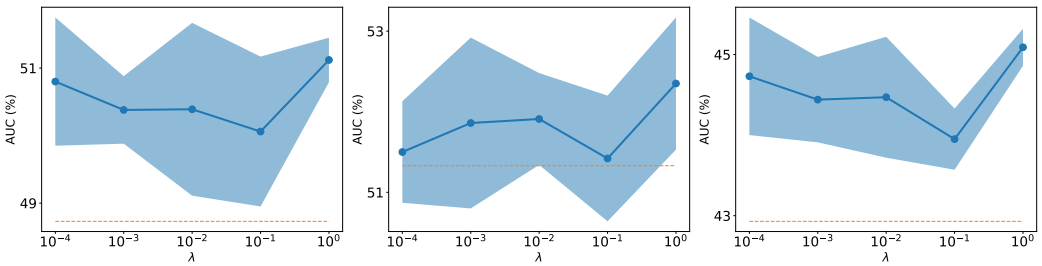
4.6.3 Efficiency of Intervention. For **I-DIDA** and **I-DIDA** without intervention mechanism, we compare their training time for each epoch on COLLAB dataset. As shown in Figure 5b, the intervention mechanism adds few costs in training time (lower than 5%). Moreover, as **I-DIDA** does not use the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

4.6.4 Hyperparameter Sensitivity. We analyze the sensitivity of hyperparameter λ_{do} in **I-DIDA** for each dataset. From Figure 6, we can see that as λ_{do} is too small or too large, the model's performance drops in most datasets. It shows that λ_{do} acts as a balance between how **I-DIDA** exploits the patterns and satisfies the invariance constraint. From Figure 8 and Figure 7, the model



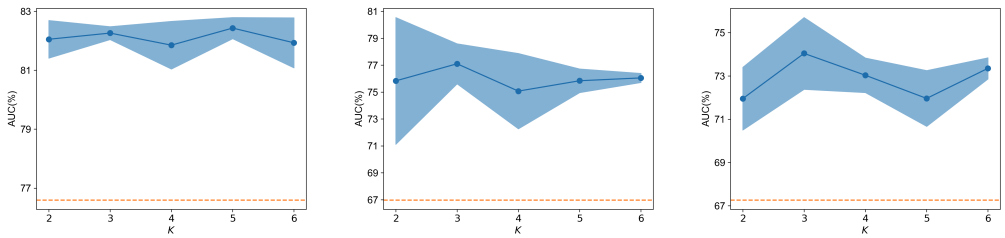
(a) 2015-2016 (b) 2017-2018 (c) 2019-2020

Fig. 7. Sensitivity of hyperparameter λ_e on the OGBN-Arxiv dataset. The area shows the average accuracy and standard deviations in the test stage, which ranges from 2015 to 2020. The dashed line represents the average accuracy of the best-performed baseline.



(a) 2015 (b) 2016 (c) 2017

Fig. 8. Sensitivity of hyperparameter λ_e on the Aminer dataset. The area shows the average accuracy and standard deviations in the test stage, which ranges from 2015 to 2017. The dashed line represents the average accuracy of the best-performed baseline.



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

Fig. 9. Sensitivity of hyperparameter K on different datasets. 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.

significantly outperforms the best-performed baseline with a large range of hyperparameters λ_e. It shows that the environment-level invariance loss promotes the invariance properties of the invariant patterns, and similarly, the hyperparameter λ_e controls the balance between the empirical

risk minimization and the invariance constraint. From Figure 9, the model outperforms the best-performed baseline with number of environments K from 2 to 6, which shows that the model is robust to the number of environments.

4.7 Implementation Details

4.7.1 Hyperparameters. For all methods, we adopt the Adam optimizer [66] with a learning rate 0.01, weight decay 5e-7 and set the patience of early stopping on the validation set as 50. The hidden dimension is set to 16 for link prediction tasks and 32 for node classification tasks. The number of layers is set to 2. Other hyper-parameters are selected using the validation datasets. For DIDA, we set the number of intervention samples as 1000 for link prediction tasks, and 100 for node classification tasks, and set λ_{do} as 1e-2,1e-2,1e-1,1e-4,1e-4 for COLLAB, Yelp, Synthetic, Arxiv and Aminer dataset respectively. For **I-DIDA**, we adopt cosine distance for all datasets, coefficient λ_e as 1e-2,1e-2,1e-1,1e-4,1 for COLLAB, Yelp, Synthetic, Arxiv and Aminer dataset respectively, and the environment number K as 4 for all datasets.

4.7.2 Evaluation Details. For link prediction tasks, 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 the 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. For node classification tasks, we adopt cross-entropy as the loss function ℓ and use Accuracy (ACC) as the evaluation metric.

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

$$\mathrm{FC}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}. \tag{22}$$

We implement the aggregation function for the invariant and variant patterns as

$$\begin{aligned} \tilde{\mathbf{z}}_I^t(u) &= \sum_i \mathbf{m}_{I,i}(\mathbf{v}_i \odot \mathbf{m}_f), \\ \mathbf{z}_I^t(u) &= \mathrm{FFN}(\tilde{\mathbf{z}}_I^t(u) + \mathbf{h}_u^t), \end{aligned} \tag{23}$$

$$\begin{aligned} \tilde{\mathbf{z}}_V^t(u) &= \sum_i \mathbf{m}_{V,i}\mathbf{v}_i, \\ \mathbf{z}_V^t(u) &= \mathrm{FFN}(\tilde{\mathbf{z}}_V^t(u)), \end{aligned} \tag{24}$$

where the FFN includes a layer normalization [4], multi-layer perceptron and skip connection,

$$\mathrm{FFN}(\mathbf{x}) = \alpha \cdot \mathrm{MLP}(\mathrm{LayerNorm}(\mathbf{x})) + (1-\alpha)\cdot \mathbf{x}, \tag{25}$$

where α is a learnable parameter. For link prediction tasks, we implement the predictor $f(\cdot)$ in Eq.(10) as inner product of hidden embeddings, *i.e.*,

$$f(\mathbf{z}_I^t(u), \mathbf{z}_I^t(v)) = \mathbf{z}_I^t(u) \cdot (\mathbf{z}_I^t(v))^T, \tag{26}$$

which conforms to classic link prediction settings. To implement the predictor $g(\cdot)$ in Eq.(11), we adopt the biased training technique following [14], *i.e.*,

$$\begin{aligned} &g(\mathbf{z}_V^t(u), \mathbf{z}_I^t(u), \mathbf{z}_V^t(v), \mathbf{z}_I^t(v)) \\ =&f(\mathbf{z}_I^t(u), \mathbf{z}_I^t(v)) \cdot \sigma(f(\mathbf{z}_V^t(u), \mathbf{z}_V^t(v))), \end{aligned} \tag{27}$$

For node classification tasks, we implement the predictor $f(\cdot)$ in Eq.(10) as a linear classifer, *i.e.*,

$$f(\mathbf{z}_I^t(u)) = \mathbf{W}\mathbf{z}_I^t(u) + \mathbf{b}. \tag{28}$$

Following [146], we use an additional shortcut loss to train the linear classifier of the variant patterns for the node u, *i.e.*,

$$\mathcal{L}_s = \ell(f(\mathbf{z}_V^t(u)), \mathbf{y}_u) \tag{29}$$

Note that this loss is just used for training the classifier, and does not update other neural networks, e.g., the disentangled dynamic graph attention. Similarly, we implement the predictor $g(\cdot)$ in Eq.(11) as

$$g(\mathbf{z}_V^t(u), \mathbf{z}_I^t(u)) = f(\mathbf{z}_I^t(u)) \cdot \sigma(f(\mathbf{z}_V^t(u)). \tag{30}$$

4.7.4 Configurations. We implement our method with PyTorch, and conduct the experiments on all datasets 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 [99] 1.11.0, PyTorch Geometric [42] 2.0.3.

5 EXPERIMENTS ON SEQUENTIAL RECOMMENDATION

In this section, we conduct extensive experiments for sequential recommendation tasks to verify that our framework can handle spatio-temporal distribution shifts in sequential recommendation.

5.1 Baselines

We compare our approach with following sequential recommendation baselines following the literature [40, 114]:

- **POP** is a straightforward approach that ranks items based on their popularity.
- **BPR-MF** [103] leverages implicit feedback to learn personalized item rankings, utilizing matrix factorization as a foundational method for recommendation.
- **NCF** [51] employs a neural network architecture in place of the traditional inner product to model interactions between users and items.
- **FPMC** [12] integrates matrix factorization with Markov chains to effectively capture and model user preferences.
- **GRU4Rec** [53] utilizes Gated Recurrent Units to model the sequential information inherent in user behaviors.
- **LightGCN** [50] is a graph-based model that simplifies the message passing process in GCNs.
- **TransRec** [49] is a model that leverages latent transition space to embed items.
- **Caser** [118] incorporates convolution operations to effectively model high-order Markov chains.
- **SASRec** [64] maximizes the utilization of self-attention mechanism and stands out as one of the early adopters of Transformers for sequential recommendation tasks.
- **Bert4Rec** [114] applies the Cloze objective to sequential recommendation, where it predicts the masked item by leveraging both the left and right context.
- **DSSRec** [91] integrates disentangled representation learning and self-supervised learning to effectively balance the weights of multiple interests.
- **ComiRec** [17] incorporates attention mechanism and user interests to enhance recommendation performance.
- **DT4SR** [39] utilizes distributions to encode items and sequences and introduces two transformers for modeling the mean and covariance embeddings.

- **ICLRec** [28] utilizes contrastive learning techniques to effectively capture and model distinct purchasing interests.
- **STOSA** [40] employs a stochastic Gaussian distribution to effectively capture the similarity between various items by treating their embedding.
- **DROS** [154] is a competitive and generic learning framework that enhance the sequential recommendation performance in the dynamic environment by leveraging a distributional robust optimization strategy. Note that DROS is a model-agnostic method that can be applied to general sequential recommendation models. We choose SASRec-DROS as a baseline due to its competitive performance.

5.2 Datasets

We conduct experiments on the following publicly available datasets:

- **Amazon**[3]: A dataset is a widely used dataset for sequential recommendation. It contains user-item interactions from the Amazon website. In our experiments, we use the 'Beauty', 'Home and Kitchen', 'Tools and Home Improvement', 'Toys and Games' and 'Office' subsets of the Amazon dataset.
- **MovieLens**[4]: This dataset contains multiple user ratings for multiple movies. We use MovieLens-1m (ML-1m) and MovieLens-20m (ML-20m) datasets in our experiments.

In the experiments, we divide the sequence of each user into training, validation, and testing sets. The last item in the sequence is utilized for testing, the second-to-last item is for validation, and all the remaining items are for training. We adopt the same evaluation design as in the original papers of the competitive baselines. For the experimental data of SASRec and Bert4Rec, we follow popularity negative sampling, that for each ground-truth item in the testing set, we randomly sample 100 negative items that the user has not interacted with. The popularity of the items is used as the sampling probability. On the experimental data of STOSA, we adopt a non-sampling strategy that all items that the user has not interacted with are considered as negative samples. The details of datasets statistics used for popularity negative sampling are shown in Table 6 and statistics used for non-sampling strategy are shown in Table 7. Note that statistics of Beauty dataset in Table 6 differ from those in Table 7 is because the latter employs 5-core settings.

Table 6. Summarization of dataset statistics for sequential recommendation with popularity negative sampling.

Dataset	Beauty	ML-1M	ML-20M
# user	40,226	6,040	138,493
# items	54,542	3,416	26,744
# interactions	353,962	999,611	20,000,263
density	0.02%	4.84%	0.54%
average interactions per user	8.7993	165.4985	144.4135

5.3 Results

The performance of different methods is presented in Table 8 and Table 9. Based on the experiment results, we have the following observations. Sequential models like GRU4Rec and Caser outperform

[3]http://jmcauley.ucsd.edu/data/amazon/
[4]https://grouplens.org/datasets/movielens/

Table 7. Summarization of dataset statistics for sequential recommendation with non-sampling strategy.

Dataset	Home	Beauty	Toys	Tools	Office
# user	66,519	22,363	19,412	16,638	4,905
# items	28,238	12,102	11,925	10,218	2,421
# interactions	551,582	198,502	167,597	134,476	53,258
density	0.03%	0.07%	0.07%	0.08%	0.45%
average interactions per user	8.2936	8.8764	8.6337	8.0825	10.8579

non-sequential models such as BPR-MF and LightGCN, indicating that non-sequential models which solely rely on user behavior information and neglect the temporal aspect lead to suboptimal recommendation performance. Among the non-sequential models, LightGCN shows the most promising results, demonstrating the effectiveness of incorporating graph data to capture user interaction behavior. On the other hand, Transformer-based approaches such as SASRec and Bert4Rec not only leverage temporal information but also capture diverse user intents, resulting in superior performance. SASRec-DROS brings a significant improvement over SASRec, demonstrating that distribution shift is a critical issue in sequential recommendation tasks. Our method consistently shows improvements and outperforms most other baselines. The results demonstrate that **I-DIDA** is effective in capturing invariant patterns between different environments and can effectively handle the spatio-temporal distribution shifts in sequential recommendation tasks.

Table 8. Overall performance comparisons on sequential recommendation datasets with popularity negative sampling. The best and second-best results are bold and underlined.

Dataset	Metric	POP	BPR-MF	NCF	FPMC	GRU4Rec	Caser	SASRec	Bert4Rec	DROS	IDIDA
Beauty	HR@5	0.0392	0.1209	0.1305	0.1387	0.1315	0.1625	0.1934	0.2343	0.2076	**0.2446**
	NDCG@5	0.0230	0.0814	0.0855	0.0902	0.0812	0.1050	0.1436	0.1711	0.1481	**0.1770**
	AUC	0.5201	0.5434	0.5467	0.5534	0.5867	0.6041	0.6634	0.6679	0.6461	**0.6680**
ML-1m	HR@5	0.0715	0.2866	0.1932	0.4297	0.4673	0.5353	0.5435	0.5902	0.5864	**0.6051**
	NDCG@5	0.0416	0.1903	0.1146	0.2885	0.3196	0.3832	0.3980	0.4515	0.4315	**0.4615**
	AUC	0.5251	0.7411	0.7349	0.7556	0.8311	0.8469	0.8725	0.8805	0.8810	**0.8827**
ML-20m	HR@5	0.0805	0.2128	0.1358	0.3601	0.4657	0.3804	0.5727	0.5439	0.5788	**0.5820**
	NDCG@5	0.0511	0.1332	0.0771	0.2239	0.3090	0.2538	0.4208	0.4018	0.4220	**0.4276**
	AUC	0.5329	0.7213	0.7009	0.7211	0.7780	0.8393	0.8884	0.8863	**0.9005**	0.8959

5.4 Ablation studies

In this section, we conduct ablation studies to investigate the effectiveness of the environment inference, intervention mechanism and disentangled attention in **I-DIDA** for sequential recommendation tasks.

5.4.1 Spatio-Temporal Environment Inference. We remove the spatio-temporal environment inference module mentioned in Sec. 3.4. The results are shown in Figure 10. We can see that the spatio-temporal environment inference module is crucial for capturing the spatio-temporal distribution shifts. The model without the spatio-temporal environment inference module performs worse than the full model, which demonstrates that our environment-level invariance loss is effective in capturing the spatio-temporal distribution shifts in sequential recommendation tasks.

Table 9. Overall performance comparisons on sequential recommendation datasets with non-sampling strategy. The best and second-best results are bold and underlined. where the 'OOM' means the out of memory error.

Dataset	Metric	LightGCN	TransRec	Caser	SASRec	Bert4Rec	DSSRec	ComiRec	DT4SR	ICLRec	STOSA	DROS	IDIDA
Home	HR@5	0.0095	0.0063	OOM	0.0127	0.0105	0.0123	0.0092	0.0129	0.0153	0.0133	0.0137	**0.0171**
	NDCG@5	0.0060	0.0040	OOM	0.0087	0.0067	0.0085	0.0058	0.0082	0.0101	0.0093	0.0097	**0.0124**
	MRR	0.0071	0.0052	OOM	0.0094	0.0092	0.0086	0.0079	0.0093	0.0102	0.0100	0.0101	**0.0128**
Beauty	HR@5	0.0300	0.0321	0.0309	0.0416	0.0396	0.0436	0.0351	0.0449	0.0500	0.0504	0.471	**0.0549**
	NDCG@5	0.0174	0.0204	0.0214	0.0274	0.0257	0.0308	0.0219	0.0296	0.0326	0.0351	0.0332	**0.0384**
	MRR	0.0203	0.0236	0.0231	0.0291	0.0294	0.0314	0.0265	0.0323	0.0322	0.0360	0.0345	**0.0393**
Tools	HR@5	0.0231	0.0210	0.0129	0.0284	0.0189	0.0283	0.0283	0.0289	0.0326	0.0312	0.0312	**0.0347**
	NDCG@5	0.0152	0.0134	0.0091	0.0194	0.0123	0.0202	0.0204	0.0196	0.0218	0.0217	0.0208	**0.0244**
	MRR	0.0170	0.0152	0.0106	0.0207	0.0160	0.0211	0.0212	0.0206	0.0230	0.0226	0.0212	**0.0250**
Toys	HR@5	0.0266	0.0222	0.0240	0.0551	0.0300	0.0565	0.0366	0.0550	0.0598	0.0577	0.0567	**0.0619**
	NDCG@5	0.0173	0.0143	0.0210	0.0377	0.0206	0.0387	0.0233	0.0360	0.0414	0.0412	0.0400	**0.0443**
	MRR	0.0200	0.0166	0.0221	0.0385	0.0244	0.0392	0.0272	0.0387	0.0415	0.0415	0.0400	**0.0444**
Office	HR@5	0.0226	0.0343	0.0302	0.0656	0.0485	0.0599	0.0438	0.0630	0.0653	0.0677	0.0669	**0.0695**
	NDCG@5	0.0157	0.0219	0.0186	0.0428	0.0309	0.0395	0.0304	0.0421	0.0452	0.0461	0.0444	**0.0481**
	MRR	0.0181	0.0263	0.0268	0.0457	0.0408	0.0407	0.0376	0.0475	0.0495	0.0502	0.0475	**0.0512**

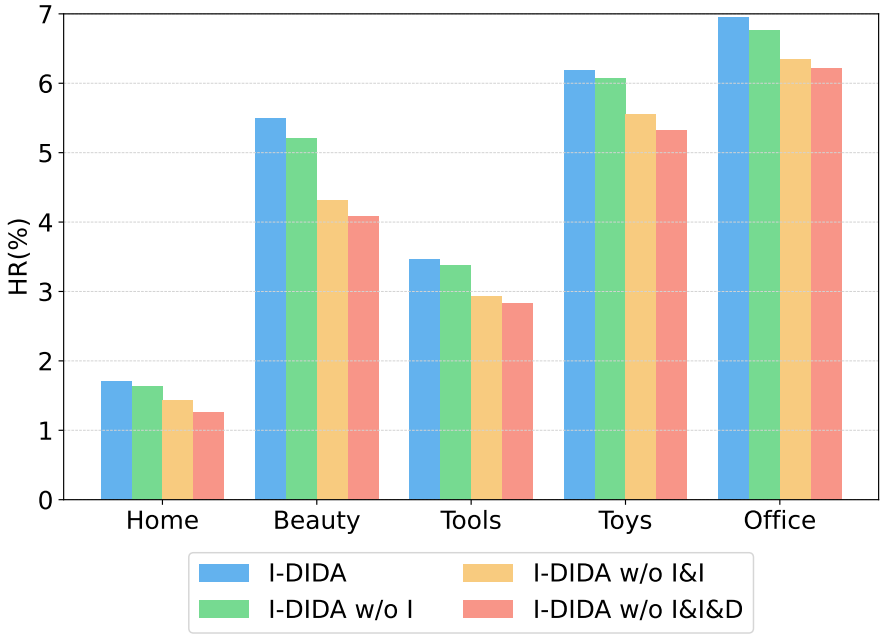


Fig. 10. Ablation studies on the environment inference, intervention mechanism and disentangled attention, where 'w/o I' removes the spatio-temporal environment inference module, 'w/o I&I' further removes the spatio-temporal intervention mechanism and 'w/o I&I&D' further removes disentangled attention. (Best viewed in color)

5.4.2 Spatio-Temporal Intervention Mechanism. We remove the spatio-temporal intervention mechanism mentioned in Sec. 3.3. The results are shown in Figure 10. We can see that the module is a

crucial component in **I-DIDA** for utilizing the invariant patterns and variant patterns. The model without the spatio-temporal intervention mechanism performs worse than the full model,which demonstrates that our spatio-temporal intervention mechanism help the model to focus on invariant patterns to make predictions in sequential recommendation tasks.

5.4.3 Disentangled Dynamic Graph Attention. we remove the disentangled dynamic graph attention mentioned in Sec 3.2. The results are shown in Figure 10. We can see that the model without disentangled dynamic graph attention performs worse than the full model, which demonstrates that our disentangled dynamic graph attention module is a crucial component in **I-DIDA** for obtaining variant patterns and invariant patterns for the subsequent prediction in sequential recommendation tasks.

5.5 Implementation details

5.5.1 Hyperparameters. In **I-DIDA** for sequential recommendation, we adopt the Adam optimizer [66] with a learning rate 0.01, weight decay 5e-7 and set the patience of early stopping on the validation set as 100. The hidden dimension is set to 32. The number of layers is set to 2. Other hyperparameters are selected based on the validation set. We set the number of intervention samples as 100, set the environment number K as 5, set λ_{do} as 1e-4,1e-4,1e-3,1e-3,1e-3,1e-3,1e-3,1e-3 for Home, Beauty(5-core), Tools, Toys, Office, Beauty, ML-1m and ML-20m dataset respectively and set λ_e as 1e-2,1e-2,1e-2,1e-3,1e-2,1e-2, 1e-2,1e-2 for Home, Beauty(5-core), Tools, Toys, Office, Beauty, ML-1m and ML-20m dataset respectively.

5.5.2 Evaluation Details. In our experiments, we adopt Hit Ratio(HR@K), Normalized Discounted Cumulative Gain(NDCG@K), Mean Reciprocal Rank(MRR) and Area Under the ROC Curve(AUC) as the evaluation metrics to evaluate the performance of all methods. HR@K measures the proportion of users for whom the ground truth item is in the top-K recommended items. NDCG@K and MRR measure the ranking quality of the recommended items. We set K as 5 in our experiments.

5.5.3 Configurations. We implement our method with PyTorch and conduct the experiments on all datasets with same configurations in Section 4.

6 RELATED WORK

In this section, we review the related works of dynamic graph neural networks, out-of-distribution generalization, disentangled representation learning, and sequential recommendation.

6.1 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) [112, 172, 177].

A classic of DyGNNs first adopt a GNN to aggregate structural information for the graph at each time, followed by a sequence model like RNN [47, 108, 115, 151] or temporal self-attention [107] to process temporal information. GCRN [108] models the structural information for each graph snapshot at different timestamps with graph convolution networks [68] and adopt GRU [30] to model the graph evolution along the temporal dimension. DyGGNN [116] adopts gated graph neural networks to learn the graph topology at each time step and LSTM [54] to propagate the temporal information among the time steps. Variational inference is further introduced to model the node dynamics in the latent space [47]. DySAT [107] aggregates neighborhood information at each snapshot similar to graph attention networks [125] and aggregates temporal information with temporal self-attention. By introducing the attention mechanism, the model can draw context from all past graphs to adaptively assign weights for messages from different time and neighbors.

Some works [5, 115, 151] learn the embeddings of dynamic graphs in hyperbolic space to exploit the hyperbolic geometry's advantages of the exponential capacity and hierarchical awareness.

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 [32, 105, 138, 150] to process structural information. For example, TGAT [150] proposes a functional time encoding technique based on the classical Bochner's theorem from harmonic analysis, which enables the learned node embeddings to be inherently represented as a function of time. To obtain more fine-grained continuous node embeddings in dynamic graphs, some work further leverages neural interaction processes [19] and ordinary differential equation [61]. EvolveGCN [98] models the network evolution from a different perspective, which learns to evolve the parameters of graph convolutional networks instead of the node embeddings by RNNs. In this way, the model does not require the knowledge of a node in the full time span, and is applicable to the frequent change of the node set.

DyGNNs have been widely applied in real-world applications, including dynamic anomaly detection [16], event forecasting [33], dynamic recommendation [158], social character prediction [139], user modeling [72], temporal knowledge graph completion [142], entity linking [143], health care [166], *etc.* For example, DGEL [117] proposes a dynamic graph evolution learning framework for generating satisfying recommendations in dynamic environments, including three efficient real-time update learning methods for nodes from the perspectives of inherent interaction potential, time-decay neighbor augmentation and symbiotic local structure learning. DynShare [174] proposes a dynamic share recommendation model that is able to recommend a friend who would like to share a particular item at a certain timestamp for social-oriented e-commerce platforms. LLM4DyG [168] proposes to handle spatial-temporal problems on dynamic graphs from perspective of leveraging both advantages of large language models and graphs [165]. PTGCN [60] models the patterns between user-item interactions in sequential recommendation by defining a position-enhanced and time-aware graph convolution operation, demonstrating great potential for online session-based recommendation scenarios.

In this paper, we consider DyGNNs under spatio-temporal distribution shift, which remains unexplored in dynamic graph neural networks literature.

6.2 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 [109]. In particular, there might be uncontrollable distribution shifts between training and testing data distribution, which may lead to a sharp drop in model performance.

To solve this problem, Out-of-Distribution (OOD) generalization problem has recently become a central research topic in various areas [109, 128, 156]. As a representative work tackling OOD generalization problems, IRM [3] aims at learning an invariant predictor which minimizes the empirical risks for all training domains, so that the classifier and learned representations match for all environments and achieve out-of-distribution generalization. GroupDRO [106] minimizes worst-group risks across training domains by putting more weight on training domains with larger errors when minimizing empirical risk. VREx [71] reduces differences in risk across training domains to reduce the model's sensitivity to distribution shifts.

Recently, several works attempt to handle distribution shift on graphs [15, 20, 29, 38, 73, 76, 77, 79, 80, 82, 101, 152, 157, 167, 173], where the distribution shift can exist on graph topologies, e.g., graph sizes and other structural properties. For example, some work [11] assumes independence between cause and mechanism, and constructs a structural causal model to learn the graph representations that can extrapolate among different size distributions for graph classification tasks. Some work [48]

interpolates the node features and graph structure in embedding space as data augmentation to improve the model's OOD generalization abilities. EERM [144] proposes to utilize multiple context explorers that are adversarially trained to maximize the variance of risks from multiple virtual environments, so that the model can extrapolate from a single observed environment for node-level prediction. DIR [146] attempts to capture the causal rationales that are invariant under structural distribution shift and filter out the unstable spurious patterns. DR-GST [83] finds that high-confidence unlabeled nodes may introduce the distribution shift issue between the original labeled dataset and the augmented dataset in self-training, and proposes a framework to recover the distribution of the original labeled dataset. SR-GNN [176] adapts GNN models to tackle the distributional differences between biased training data and the graph's true inference distribution. GDN [44] discovers the structural distribution shifts in graph anomaly detection, that is, the heterophily and homophily can change across training and testing data. They solve the problem by teasing out the anomaly features, on which they constrain to mitigate the effect of heterophilous neighbors and make them invariant. GOOD-D [86] studies the problem of unsupervised graph out-of-distribution detection and creates a comprehensive benchmark to make comparisons of several state-of-the-art methods. Some works focus on the distribution shift problem in general recommendation. DESMIL [85], a multi-interest learning framework, can eliminate spurious interests and adapt to distribution shifts. AST [148] links unbiased recommendation with distribution shift and presents a novel adversarial self-training framework for unbiased recommendation. CausPref [52], a causal recommendation model, can handle the distribution shift problem by learning the causal relationships between users and items.

Another classic of OOD methods most related to our works handle distribution shifts on time-series data [81, 88, 126]. For example, some work [65] observes that statistical properties such as mean and variance often change over time in time series, and propose a reversible instance normalization method to remove and restore the statistical information for tackling the distribution shifts. AdaRNN [37] formulates the temporal covariate shift problem for time series forecasting and proposes to characterize the distribution information and reduce the distribution mismatch during the training of RNN-based prediction models. DROS [154] proposes a distributionally robust optimization mechanism with a distribution adaption paradigm to capture the dynamics of data distribution and explore the possible distribution shifts for sequential recommendation. Wild-Time [155] creates a benchmark of datasets that reflect the temporal distribution shifts arising in a variety of real-world time-series applications like patient prognosis, showing that current time-series and out-of-distribution methods still have limitations in tackling temporal distribution shifts. WOODS [43] is another benchmark for out-of-distribution generalization methods in time series tasks, including videos, brain recordings, smart device sensory signals, *etc.*

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 paper is the first study of spatio-temporal distribution shifts in dynamic graphs.

6.3 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 [7]. Besides its applications in computer vision [23–25, 27, 34, 55, 92, 122, 136] and recommendation [21, 74, 90, 91, 132–135, 164], several disentangled GNNs have proposed to generalize disentangled representation learning in graph data recently [170, 171]. DisenGCN [89] learns disentangled node representations by proposing a neighborhood routing mechanism in the graph convolution networks to identify the factors that may cause the links from the nodes to their neighbors. IPGDN [87] further encourages

the graph latent factors to be as independent as possible by minimizing the dependence among representations with a kernel-based measure. FactorGCN [153] decomposes the input graph into several interpretable factor graphs, and each of the factor graphs is fed into a different GCN so that different aspects of the graph can be modeled into factorized graph representations. DGCL [75] and IDGCL [78] aim to learn disentangled graph-level representations with self-supervision to reduce the potential negative effects of the bias brought by supervised signals. However, most of these methods are designed for static graphs and may not disentangle the factors with the consideration of the structural and temporal information on graphs. GRACES [101] designs a self-supervised disentangled graph encoder to characterize the invariant factors hidden in diverse graph structures, and thus facilitates the subsequent graph neural architecture search. Some other works factorize deep generative models based on node, edge, static, dynamic factors [163] or spatial, temporal, graph factors [36] to achieve interpretable dynamic graph generation. DisenCTR [141] proposes a disentangled graph representation module to extract diverse user interests and exploit the fluidity of user interests and model the temporal effect of historical behaviors using a mixture of Hawkes process. In this paper, we borrow the idea of disentangled representation learning, and disentangle the spatio-temporal patterns on dynamic graphs into invariant and variant parts for the subsequent invariant learning to enhance the model's generalization ability under distribution shifts.

6.4 Sequential Recommendation

Different from traditional recommendation systems [2, 26, 137, 140], sequential recommendation further leverages sequential information, aiming to predict the next item that a user will interact with based on the user's historical interactions.

With the development of deep learning, many deep learning-based methods have been proposed for sequential recommendation. GRU4Rec [53] is one of the early works that uses RNN to model the user-item interactions in the session-based recommendation. Besides, convolutional neural networks have also been widely used in sequential recommendation. Caser [118] incorporates convolution operations to effectively model high-order Markov chains. Furthermore, attention mechanism has been adopted in sequential recommendation as a powerful tool to capture the user's interests. SASRec [64] maximizes the utilization of self-attention mechanism and is one of the early adopters of transformers for sequential recommendation tasks. Diff4Rec [147] leverages diffusion process for sequential recommendation.

Graph neural networks have also been widely used in sequential recommendation because of their ability to leverage the graph structure information. There are some works [22, 145, 149, 160] that use graph neural networks to model the user-item interactions in sequential recommendation. SR-GNN [145] uses a graph neural network to model the user-item interactions in the session-based recommendation. A-PGNN [160] combine personalized GNN and attention mechanism to model the user-item interactions.

Although the GNN-based methods have achieved good performance in sequential recommendation, they are not designed to utilize the item relationship across different user-item interactions sequences. To address this issue, some works [113, 127, 130, 169] have been proposed. HyperRec [127] utilizes a hypergraph to effectively capture the high-order correlations between items within and across sequences. CSRM [130] takes into consideration neighboring sessions by evaluating the similarity between the current session and other sessions. DGRec [113] establishes explicit associations among different user sequences based on social relationships. DGSR [161] establishes connections between different user sequences using a dynamic graph structure, thereby exploring the interactive behavior of users and items with respect to time and order information.

However, these methods do not consider the spatio-temporal distribution shifts in sequential recommendation. In this paper, we propose a novel method to handle the spatio-temporal distribution shifts in sequential recommendation tasks.

7 CONCLUSION

In this paper, we propose Disentangled Intervention-based Dynamic Graph Attention Networks with Invariance Promotion (**I-DIDA**) to handle spatio-temporal distribution shifts in sequential recommendation. 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. Moreover, based on the invariant learning literature, we design a spatio-temporal environment inference to infer the latent environments of the nodes at different time, and propose an environment-level invariance loss to promote the invariance properties of the captured invariant patterns. Extensive experiments on one synthetic dataset and several real-world datasets demonstrate the superiority of our proposed method against state-of-the-art baselines to handle spatio-temporal distribution shifts. Experiments on sequential recommendation datasets also show our method can effectively perform accurate recommendations for sequential user-item systems under spatio-temporal distribution shifts.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China No. 2022ZD0115903, 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] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. 2020. Invariant risk minimization games. In *International Conference on Machine Learning*. PMLR, 145–155.

[2] Vineeta Anand and Ashish Kumar Maurya. 2024. A Survey on Recommender Systems using Graph Neural Network. *ACM Trans. Inf. Syst.* (Sept. 2024). https://doi.org/10.1145/3694784 Just Accepted.

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint* (2019).

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

[5] Qijie Bai, Changli Nie, Haiwei Zhang, Dongming Zhao, and Xiaojie Yuan. 2023. HGWaveNet: A Hyperbolic Graph Neural Network for Temporal Link Prediction. In *Proceedings of the ACM Web Conference 2023*. 523–532.

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

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

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

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

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

[11] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. 2021. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*. 837–851.

[12] M Boonekamp, A Dechambre, V Juranek, O Kepka, Murilo Rangel, Christophe Royon, and R Staszewski. 2011. FPMC: a generator for forward physics. *arXiv preprint arXiv:1102.2531* (2011).
[13] Stephen J Brown, William Goetzmann, Roger G Ibbotson, and Stephen A Ross. 1992. Survivorship bias in performance studies. *The Review of Financial Studies* 5, 4 (1992), 553–580.
[14] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. 2019. Rubi: Reducing unimodal biases for visual question answering. *Advances in neural information processing systems* (2019).
[15] Jie Cai, Xin Wang, Haoyang Li, Ziwei Zhang, and Wenwu Zhu. 2024. Multimodal Graph Neural Architecture Search under Distribution Shifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8227–8235.
[16] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. 2021. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *Proceedings of the 30th ACM international conference on Information & Knowledge Management*. 3747–3756.
[17] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.
[18] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *International Conference on Machine Learning*. PMLR, 1448–1458.
[19] Xiaofu Chang, Xuqin Liu, Jianfeng Wen, Shuang Li, Yanming Fang, Le Song, and Yuan Qi. 2020. Continuous-time dynamic graph learning via neural interaction processes. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 145–154.
[20] Cen Chen, Tiandi Ye, Li Wang, and Ming Gao. 2022. Learning to generalize in heterogeneous federated networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 159–168.
[21] Hong Chen, Yudong Chen, Xin Wang, Ruobing Xie, Rui Wang, Feng Xia, and Wenwu Zhu. 2021. Curriculum Disentangled Recommendation with Noisy Multi-feedback. *Advances in Neural Information Processing Systems* 34 (2021), 26924–26936.
[22] Hong Chen, Bin Huang, Xin Wang, Yuwei Zhou, and Wenwu Zhu. 2023. Global-Local GraphFormer: Towards Better Understanding of User Intentions in Sequential Recommendation. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia*. 1–7.
[23] Hong Chen, Xin Wang, Yipeng Zhang, Yuwei Zhou, Zeyang Zhang, Siao Tang, and Wenwu Zhu. 2024. DisenStudio: Customized Multi-subject Text-to-Video Generation with Disentangled Spatial Control. arXiv:2405.12796 [cs.CV]
[24] Hong Chen, Yipeng Zhang, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. 2024. DisenDreamer: Subject-Driven Text-to-Image Generation with Sample-aware Disentangled Tuning. *IEEE Transactions on Circuits and Systems for Video Technology* (2024).
[25] Hong Chen, Yipeng Zhang, Simin Wu, Xin Wang, Xuguang Duan, Yuwei Zhou, and Wenwu Zhu. 2023. Disenbooth: Identity-preserving disentangled tuning for subject-driven text-to-image generation. *arXiv preprint arXiv:2305.03374* (2023).
[26] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Trans. Inf. Syst.* 41, 3, Article 67 (Feb. 2023), 39 pages. https://doi.org/10.1145/3564284
[27] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems* 29 (2016).
[28] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.
[29] Yongqiang Chen, Yonggang Zhang, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022. Invariance Principle Meets Out-of-Distribution Generalization on Graphs. *arXiv preprint* (2022).
[30] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *EMNLP*.
[31] James S Coleman. 1994. *Foundations of social theory*. Harvard university press.
[32] Weilin Cong, Yanhong Wu, Yuandong Tian, Mengting Gu, Yinglong Xia, Mehrdad Mahdavi, and Chun-cheng Jason Chen. 2021. Dynamic Graph Representation Learning via Graph Transformer Networks. *arXiv preprint arXiv:2111.10447* (2021).
[33] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2020. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1585–1595.
[34] Emily L Denton et al. 2017. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems* 30 (2017).

[35] Mucong Ding, Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Micah Goldblum, David Wipf, Furong Huang, and Tom Goldstein. 2021. A Closer Look at Distribution Shifts and Out-of-Distribution Generalization on Graphs. (2021).

[36] Yuanqi Du, Xiaojie Guo, Hengning Cao, Yanfang Ye, and Liang Zhao. 2022. Disentangled Spatiotemporal Graph Generative Models. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 6541–6549.

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

[38] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. 2021. Generalizing Graph Neural Networks on Out-Of-Distribution Graphs. *arXiv preprint arXiv:2111.10657* (2021).

[39] Ziwei Fan, Zhiwei Liu, Shen Wang, Lei Zheng, and Philip S Yu. 2021. Modeling sequences as distributions with uncertainty for sequential recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 3019–3023.

[40] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM Web Conference 2022*. 2036–2047.

[41] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *ACM Trans. Inf. Syst.* 39, 1, Article 10 (nov 2020), 42 pages. https://doi.org/10.1145/3426723

[42] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[43] Jean-Christophe Gagnon-Audet, Kartik Ahuja, Mohammad-Javad Darvishi-Bayazi, Guillaume Dumas, and Irina Rish. 2022. WOODS: Benchmarks for Out-of-Distribution Generalization in Time Series Tasks. *arXiv preprint arXiv:2203.09978* (2022).

[44] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 357–365.

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

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

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

[48] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. 2022. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*. 8230–8248.

[49] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 161–169.

[50] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[51] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[52] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. 2022. Causpref: Causal preference learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. 410–421.

[53] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

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

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

[56] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[57] Hong Huang, Zixuan Fang, Xiao Wang, Youshan Miao, and Hai Jin. 2020. Motif-Preserving Temporal Network Embedding.. In *IJCAI*. 1237–1243.

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

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

[60] Liwei Huang, Yutao Ma, Yanbo Liu, Bohong Danny Du, Shuliang Wang, and Deyi Li. 2023. Position-enhanced and time-aware graph convolutional network for sequential recommendations. *ACM Transactions on Information Systems (TOIS)* 41, 1 (2023), 1–32.
[61] Zijie Huang, Yizhou Sun, and Wei Wang. 2021. Coupled Graph ODE for Learning Interacting System Dynamics.. In *KDD*. 705–715.
[62] Tian Jin, Qiong Wu, Xuan Ou, and Jianjun Yu. 2021. Community detection and co-author recommendation in co-author networks. *International Journal of Machine Learning and Cybernetics* 12, 2 (2021), 597–609.
[63] Mengyuan Jing, Yanmin Zhu, Tianzi Zang, and Ke Wang. 2023. Contrastive Self-supervised Learning in Recommender Systems: A Survey. *ACM Trans. Inf. Syst.* 42, 2, Article 59 (nov 2023), 39 pages. https://doi.org/10.1145/3627158
[64] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
[65] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations*.
[66] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.).
[67] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
[68] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations*. OpenReview.net.
[69] Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki. 2011. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment* 2011, 11 (2011), P11005.
[70] Lauri Kovanen, Kimmo Kaski, János Kertész, and Jari Saramäki. 2013. Temporal motifs reveal homophily, gender-specific patterns, and group talk in call sequences. *Proceedings of the National Academy of Sciences* 110, 45 (2013), 18070–18075.
[71] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*. 5815–5826.
[72] Haoyang Li, Peng Cui, Chengxi Zang, Tianyang Zhang, Wenwu Zhu, and Yishi Lin. 2019. Fates of Microscopic Social Ecosystems: Keep Alive or Dead?. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 668–676.
[73] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. 2024. Disentangled Graph Self-supervised Learning for Out-of-Distribution Generalization. In *Forty-first International Conference on Machine Learning*.
[74] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. 2021. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[75] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. 2021. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems* 34 (2021), 21872–21884.
[76] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* (2022).
[77] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Out-Of-Distribution Generalization on Graphs: A Survey. *arXiv preprint* (2022).
[78] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Disentangled Graph Contrastive Learning With Independence Promotion. *IEEE Transactions on Knowledge and Data Engineering* (2022).
[79] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Learning Invariant Graph Representations for Out-of-Distribution Generalization. In *Thirty-Sixth Conference on Neural Information Processing Systems*.
[80] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2023. Invariant Node Representation Learning under Distribution Shifts with Multiple Latent Environments. *ACM Transactions on Information Systems (TOIS)* (jun 2023). https://doi.org/10.1145/3604427 Just Accepted.
[81] Peiwen Li, Xin Wang, Zeyang Zhang, Yuan Meng, Fang Shen, Yue Li, Jialong Wang, Yang Li, and Wenweu Zhu. 2024. Llm-enhanced causal discovery in temporal domain from interventional data. *arXiv preprint arXiv:2404.14786* (2024).
[82] Peiwen Li, Xin Wang, Zeyang Zhang, Yijian Qin, Ziwei Zhang, Jialong Wang, Yang Li, and Wenwu Zhu. 2024. Causal-Aware Graph Neural Architecture Search under Distribution Shifts. *arXiv preprint arXiv:2405.16489* (2024).
[83] Hongrui Liu, Binbin Hu, Xiao Wang, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. 2022. Confidence may cheat: Self-training on graph neural networks under distribution shift. In *Proceedings of the ACM Web Conference 2022*. 1248–1258.
[84] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. 2021. Heterogeneous risk minimization. In *International Conference on Machine Learning*. PMLR, 6804–6814.

[85] Qiang Liu, Zhaocheng Liu, Zhenxi Zhu, Shu Wu, and Liang Wang. 2023. Deep stable multi-interest learning for out-of-distribution sequential recommendation. *arXiv preprint arXiv:2304.05615* (2023).
[86] Yixin Liu, Kaize Ding, Huan Liu, and Shirui Pan. 2023. Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 339–347.
[87] Yanbei Liu, Xiao Wang, Shu Wu, and Zhitao Xiao. 2020. Independence promoted graph disentangled networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4916–4923.
[88] Wang Lu, Jindong Wang, Yiqiang Chen, and Xinwei Sun. 2021. DIVERSIFY to Generalize: Learning Generalized Representations for Time Series Classification. *arXiv preprint* (2021).
[89] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *International conference on machine learning*. PMLR, 4212–4221.
[90] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems* 32 (2019).
[91] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.
[92] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. 2018. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99–108.
[93] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.).
[94] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).
[95] Jovana Mitrovic, Brian McWilliams, Jacob C. Walker, Lars Holger Buesing, and Charles Blundell. 2021. Representation Learning via Invariant Causal Mechanisms. In *9th International Conference on Learning Representations*. OpenReview.net.
[96] Diego C Nascimento, Bruno A Pimentel, Renata MCR Souza, Lilia Costa, Sandro Gonçalves, and Francisco Louzada. 2021. 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 (2021), 111440.
[97] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 601–610.
[98] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5363–5370.
[99] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* (2019).
[100] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19 (2000), 2.
[101] Yijian Qin, Xin Wang, Ziwei Zhang, Pengtao Xie, and Wenwu Zhu. 2022. Graph Neural Architecture Search Under Distribution Shifts. In *International Conference on Machine Learning*. 18083–18095.
[102] Zhenyu Qiu, Wenbin Hu, Jia Wu, Weiwei Liu, Bo Du, and Xiaohua Jia. 2020. Temporal network embedding with high-order nonlinear information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5436–5443.
[103] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
[104] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. 2021. The Risks of Invariant Risk Minimization. In *9th International Conference on Learning Representations*. OpenReview.net.
[105] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637* (2020).
[106] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. [n. d.]. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*.
[107] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 519–527.
[108] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*. Springer, 362–373.
[109] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624* (2021).
[110] Georg Simmel. 1950. *The sociology of georg simmel*. Vol. 92892. Simon and Schuster.

[111] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*. ACM, 243–246.

[112] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. 2021. Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey. *IEEE Access* (2021), 79143–79168.

[113] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining*. 555–563.

[114] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

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

[116] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. 2019. Learning to represent the evolution of dynamic graphs with recurrent models. In *Proceedings of the ACM Web Conference 2019*. 301–307.

[117] Haoran Tang, Shiqing Wu, Guandong Xu, and Qing Li. 2023. Dynamic Graph Evolution Learning for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1589–1598.

[118] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[119] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. 2012. Cross-domain Collaboration Recommendation. In *KDD'2012*.

[120] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *KDD'08*. 990–998.

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

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

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

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

[125] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. [n. d.]. Graph Attention Networks. In *International Conference on Learning Representations*.

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

[127] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1101–1110.

[128] Jindong Wang, Haoliang Li, Sinno Pan, and Xing Xie. 2023. A Tutorial on Domain Generalization. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1236–1239.

[129] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413.

[130] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten De Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 345–354.

[131] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *Proceedings of the ACM Web Conference 2022*. 3562–3571.

[132] Xin Wang, Hong Chen, Zirui Pan, Yuwei Zhou, Chaoyu Guan, Lifeng Sun, and Wenwu Zhu. 2024. Automated Disentangled Sequential Recommendation with Large Language Models. *ACM Transactions on Information Systems* (2024).

[133] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2022. Disentangled Representation Learning for Recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

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

[135] Xin Wang, Zirui Pan, Yuwei Zhou, Hong Chen, Chendi Ge, and Wenwu Zhu. 2023. Curriculum co-disentangled representation learning across multiple environments for social recommendation. In *International Conference on Machine Learning*. PMLR, 36174–36192.

[136] Xin Wang, Zihao Wu, Hong Chen, Xiaohan Lan, and Wenwu Zhu. 2023. Mixup-augmented temporally debiased video grounding with content-location disentanglement. In *Proceedings of the 31st ACM International Conference on Multimedia*. 4450–4459.

[137] Xin Wang, Wenwu Zhu, and Chenghao Liu. 2019. Social recommendation with optimal limited attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1518–1527.

[138] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. 2021. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. In *9th International Conference on Learning Representations*. OpenReview.net.

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

[140] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A Survey on the Fairness of Recommender Systems. *ACM Trans. Inf. Syst.* 41, 3, Article 52 (feb 2023), 43 pages. https://doi.org/10.1145/3547333

[141] Yifan Wang, Yifang Qin, Fang Sun, Bo Zhang, Xuyang Hou, Ke Hu, Jia Cheng, Jun Lei, and Ming Zhang. 2022. DisenCTR: Dynamic graph-based disentangled representation for click-through rate prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2314–2318.

[142] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L. Hamilton. 2020. TeMP: Temporal Message Passing for Temporal Knowledge Graph Completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 5730–5746.

[143] Junshuang Wu, Richong Zhang, Yongyi Mao, Hongyu Guo, Masoumeh Soflaei, and Jinpeng Huai. 2020. Dynamic graph convolutional networks for entity linking. In *Proceedings of The ACM Web Conference 2020*. 1149–1159.

[144] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. 2022. Handling Distribution Shifts on Graphs: An Invariance Perspective. *International Conference on Learning Representations* (2022).

[145] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[146] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *The Tenth International Conference on Learning Representations*. OpenReview.net.

[147] Zihao Wu, Xin Wang, Hong Chen, Kaidong Li, Yi Han, Lifeng Sun, and Wenwu Zhu. 2023. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9329–9335.

[148] Teng Xiao, Zhengyu Chen, and Suhang Wang. 2023. Reconsidering learning objectives in unbiased recommendation: A distribution shift perspective. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2764–2775.

[149] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph contextualized self-attention network for session-based recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

[150] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. In *8th International Conference on Learning Representations*. OpenReview.net.

[151] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. 2021. 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*. 1975–1985.

[152] Qiang Yang, Changsheng Ma, Qiannan Zhang, Xin Gao, Chuxu Zhang, and Xiangliang Zhang. 2023. Interpretable Research Interest Shift Detection with Temporal Heterogeneous Graphs. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 321–329.

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

[154] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. 2023. A Generic Learning Framework for Sequential Recommendation with Distribution Shifts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[155] Huaxiu Yao, Caroline Choi, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. 2022. 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*.

[156] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. 2022. Improving Out-of-Distribution Robustness via Selective Augmentation. In *Proceeding of the Thirty-ninth International Conference on*

Machine Learning.

[157] Yang Yao, Xin Wang, Yijian Qin, Ziwei Zhang, Wenwu Zhu, and Hong Mei. 2024. Data-augmented curriculum graph neural architecture search under distribution shifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 16433–16441.

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

[159] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2022. A Survey on Cross-domain Recommendation: Taxonomies, Methods, and Future Directions. *ACM Trans. Inf. Syst.* 41, 2, Article 42 (dec 2022), 39 pages. https://doi.org/10.1145/3548455

[160] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. 2020. Personalized graph neural networks with attention mechanism for session-aware recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3946–3957.

[161] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4741–4753.

[162] Shilei Zhang, Toyotaro Suzumura, and Li Zhang. 2021. DynGraphTrans: Dynamic Graph Embedding via Modified Universal Transformer Networks for Financial Transaction Data. In *2021 IEEE International Conference on Smart Data Services (SMDS)*. IEEE, 184–191.

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

[164] Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. 2023. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3434–3445.

[165] Ziwei Zhang, Haoyang Li, Zeyang Zhang, Yijian Qin, Xin Wang, and Wenwu Zhu. 2023. Graph meets llms: Towards large graph models. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*.

[166] Zeyang Zhang, Xingwang Li, Fei Teng, Ning Lin, Xueling Zhu, Xin Wang, and Wenwu Zhu. 2023. Out-of-Distribution Generalized Dynamic Graph Neural Network for Human Albumin Prediction. In *IEEE International Conference on Medical Artificial Intelligence*.

[167] Zeyang Zhang, Xin Wang, Yijian Qin, Hong Chen, Ziwei Zhang, Xu Chu, and Wenwu Zhu. 2024. Disentangled Continual Graph Neural Architecture Search with Invariant Modular Supernet. In *Forty-first International Conference on Machine Learning*.

[168] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, and Wenwu Zhu. 2024. LLM4DyG: can large language models solve spatial-temporal problems on dynamic graphs?. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4350–4361.

[169] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. 2022. Dynamic graph neural networks under spatio-temporal distribution shift. In *Advances in Neural Information Processing Systems*.

[170] Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. 2023. Spectral Invariant Learning for Dynamic Graphs under Distribution Shifts. In *Advances in Neural Information Processing Systems*.

[171] Zeyang Zhang, Xin Wang, Ziwei Zhang, Guangyao Shen, Shiqi Shen, and Wenwu Zhu. 2023. Unsupervised Graph Neural Architecture Search with Disentangled Self-supervision. In *Advances in Neural Information Processing Systems*.

[172] Zeyang Zhang, Ziwei Zhang, Xin Wang, Yijian Qin, Zhou Qin, and Wenwu Zhu. 2023. Dynamic Heterogeneous Graph Attention Neural Architecture Search. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*.

[173] Zeyang Zhang, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Learning to Solve Travelling Salesman Problem with Hardness-Adaptive Curriculum. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 9136–9144.

[174] Ziwei Zhao, Xi Zhu, Tong Xu, Aakas Lizhiyu, Yu Yu, Xueying Li, Zikai Yin, and Enhong Chen. 2023. Time-interval Aware Share Recommendation via Bi-directional Continuous Time Dynamic Graphs. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 822–831.

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

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

[177] Yuecai Zhu, Fuyuan Lyu, Chengming Hu, Xi Chen, and Xue Liu. 2022. Learnable Encoder-Decoder Architecture for Dynamic Graph: A Survey. *arXiv preprint arXiv:2203.10480* (2022).

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