Invariant Link Selector for Spatial-Temporal Out-of-Distribution Problem

Katherine Tieu
University of Illinois
Urbana-Champaign
kt42@illinois.edu

Dongqi Fu
Meta AI
dongqifu@meta.com

Jun Wu
Michigan State University
wujun4@msu.edu

Jingrui He
University of Illinois
Urbana-Champaign
jingrui@illinois.edu

Abstract

In the era of foundation models, Out-of-Distribution (OOD) problems, i.e., the data discrepancy between the training environments and testing environments, hinder AI generalization. Further, relational data like graphs disobeying the Independent and Identically Distributed (IID) condition makes the problem more challenging, especially much harder when it is associated with time. Motivated by this, to realize the robust invariant learning over temporal graphs, we want to investigate what components in temporal graphs are most invariant and representative with respect to labels. With the Information Bottleneck (IB) method, we propose an error-bounded Invariant Link Selector that can distinguish invariant components and variant components during the training process to make the deep learning model generalizable for different testing scenarios. Besides deriving a series of rigorous generalizable optimization functions, we also equip the training with task-specific loss functions, e.g., temporal link prediction, to make pretrained models solve real-world application tasks like citation recommendation and merchandise recommendation, as demonstrated in our experiments with state-of-the-art (SOTA) methods. Our code is available at https://github.com/kthrn22/OOD-Linker

1 Introduction

Recently, foundation models have revolutionized the field of artificial intelligence, demonstrating unprecedented performance across a wide range of tasks, such as natural language processing [48] and computer vision [3]. However, these efforts face significant insufficiencies if a discrepancy between the data distributions of training and testing environments, e.g., hallucination of Large Language Models (LLMs) [38, 23, 35]. This cause is also referred to as Out-of-Distribution (or OOD) problem [57, 32, 42], which mismatch poses a substantial obstacle to the generalization capabilities of AI systems, limiting their effectiveness in real-world applications.

The challenge of OOD generalization becomes even more complicated when dealing with relational data structures, particularly graphs. Unlike traditional data types, graph data inherently violates the Independent and Identically Distributed (IID) condition [51], a fundamental assumption in many machine learning algorithms. This violation originates from the interconnected nature of graph data, where each node's features and labels are influenced by its neighbors, creating complex dependencies that are difficult to model and generalize [67, 36, 60, 8, 50, 55, 65, 66]. Furthermore, the introduction of a temporal dimension to graph data exponentially increases the complexity of the problem [7, 56, 6, 21, 46, 22, 20]. Spatial-temporal graphs, which represent evolving relationships and dynamic structures over time, present a unique set of challenges for machine learning models. The temporal aspect introduces additional variability and dependencies that must be accounted for, making the task of identifying invariant and generalizable features even more challenging.

According to the recent survey [29], it suggests that the effective graph data required by the graph foundation models is not about the size (e.g., number of nodes and edges) but the density of different subgraph

Proceedings of the 28$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

patterns; Also, there is currently no viable graph foundation models for temporal link prediction tasks, i.e., whether the link exists or not between two entities in the graph [4, 61]. Motivated by the above analysis, our research focuses on realizing robust invariant learning over spatial-temporal graphs. The central question we aim to address is: **What components in spatial-temporal graphs are most invariant and representative with respect to labels across different domains and time periods?**

Answering this question and extracting those invariant components for utilization is crucial for developing models that can generalize effectively to unseen environments and future timestamps. To this end, we leverage the Information Bottleneck (IB) method, a powerful framework for extracting relevant information from complex data structures. Building upon the IB principle, we propose a novel approach: an error-bounded **Invariant Link Selector**. This innovative method is designed to distinguish between invariant and variant components during the training process, enabling deep learning models to focus on the most stable and informative features of temporal graphs. Our Invariant Link Selector operates by:

- Identifying and prioritizing graph components that remain consistent across different domains and time periods.

- Minimizing the influence of variant components that may lead to overfitting or poor generalization.

- Adaptively adjusting the selection process based on error bounds, ensuring robustness and reliability.

In developing this approach, we derive a series of rigorous generalizable optimization processes. These processes form the theoretical foundation of our method, providing a principled way to balance the trade-off between compressing input information and preserving relevant features for the task.

Recognizing the importance of practical applicability, we augment our method with task-specific loss functions. This integration allows our pre-trained models to be fine-tuned for real-world applications such as temporal link prediction. To validate the effectiveness of our proposed method, we conduct extensive domain-shift experiments comparing our approach with state-of-the-art (SOTA) methods. These experiments are designed to evaluate not only the overall performance of our model but also its ability to generalize across different domains and time periods.

2 Preliminary

In this section, we introduce some necessary techniques and notations for paving the way for the derivation of our method.

2.1 Information Bottleneck method

The Information Bottleneck (IB) method [47] presents a principled approach to extract and compress the representation of relevant information from complex data structures.

Mathematically, the IB method seeks to find a compressed representation of an input variable that retains maximal information about a target variable. Formally, given a joint distribution $p(x, y)$ over the input variable X and target variable Y, the IB method aims to find a compressed representation T, which (1) compresses the input X as much as possible; and (2) preserves as much relevant information as possible about the target Y. Let $I(T; X)$ denote the mutual information between T and X, and let $I(T; Y)$ denote the mutual information between T and Y, the IB objective is to maximize:

$$\mathcal{L}[p(t|x)] = I(T; Y) - \beta I(T; X) \tag{1}$$

where β is a Lagrange multiplier that controls the trade-off between compression and preservation of IB.

The optimization problem in Equation 1 leads to a set of self-consistent equations:

$$\begin{aligned} p(t|x) &= Kp(t)\exp(-\beta D_{KL}[p(y|x)||p(y|t)]) \\ p(t) &= \sum_x p(x)p(t|x),\ p(y|t) = \sum_x p(y|x)p(x|t) \end{aligned} \tag{2}$$

where K is a scalar normalization and D_{KL} is the Kullback-Leibler divergence or KL-divergence [13].

2.2 Spatial-Temporal Graphs Modeling

In this paper, a spatial-temporal graph can be represented by a sequence of temporal edges between two nodes, e.g., the connection between u and v at timestamp t is $e^t_{u,v}$ =(u, v, t) and $t \in \{1, \ldots, T\}$.

Also, we consider the spatial-temporal graph is attributed, i.e., nodes and edges have time-evolving features, **bold vector** $\mathbf{s}^t_u$ denotes the node feature of node u at time t, and **bold vector** $\mathbf{e}^t_{u,v}$ denotes the edge feature of the edge $e^t_{u,v}$ at time t.

For the clear notation when deriving the theoretical analysis, we denote all connections that happen at time t as G^t, and we use spatial-temporal graph and temporal graph interchangeably.

3 Proposed Method: OOD-Linker

In this section, we first formally define the problem we target to solve with concrete examples in Section 3.1. Then, we dive into the proposed **Invariant Link Selector** with its modeling and detailed optimization procedures in Section 3.2. Given that selector, in Section 3.3, we introduce the capable neural architecture for **OOD-Linker** to achieve pre-training and finish the link prediction in the new domain-shifted setting. The theoretical analysis of all proposed techniques is placed in Section 4. Please refer to Appendix A for detailed proofs of variational bounds.

3.1 Problem Setting

Suppose we can observe the historical behavior of a temporal graph, how can we know whether a future link somewhere will appear or not, especially if we are not able to assume any condition held for the future environment, e.g., the topological structure changes, and new features and new labels emerge.

For example, in the paper citation network, after 2006, "Data Mining" papers emerge, can we use the "Information Theory" and "Database" papers interactions before 2006 to predict the citation among "Data Mining" papers?

Problem 1 (Out-of-Distribution Generalization for Temporal Link prediction). *For a* ***query link*** $(u, v, T+1)$*, we need to decide whether it will happen or not at time* $T+1$*, i.e., label* $Y_{T+1} = 1$ *means the query link exists at time* $T+1$ *and* 0 *vice versa.*

Motivated by the above question, we need to solve: **Given the unpredictable domain shift possibilities, can we disentangle the historical temporal graphs before time T and identify the invariant subgraphs that are directly related to the query link and wisely use this knowledge to make the prediction?**

3.2 Principle of Invariant Link Selector

Suppose the label of a link is determined by an invariant subgraph (e.g., the cow prediction vs. camel is determined by the shape of the object, not the color of the background [1]).

Here, we introduce how the invariant links (i.e., subgraph) are selected. In brief, to select the most label-relevant subgraph to make predictions, our proposed selector is constrained by the mutual information between the optimal invariant subgraph and the original input graph and to remove spurious correlation towards labels, like the background color green or yellow for the cow or camel prediction.

Hence, before involving time dimension, we can first denote the most label-relevant invariant subgraph as G_{inv}; then we would want to maximize the mutual information between $I(G_{inv}, Y)$ while constraining the mutual information between the original input graph G and G_{inv}, i.e., $I(G, G_{inv}) \leq \alpha$. Mathematically, we formulate our objective as follows:

$$\arg\max_{G_{inv} \subseteq G} I(G_{inv}, Y), \text{ s.t. } I(G, G_{inv}) \leq \alpha \tag{3}$$

If we can obtain a good selector function f_{inv} that extracts the optimal subgraph, satisfying the constraint in Eq. 3, then with the introduction of Lagrange multiplier β, we can re-write Eq. 3 as follows:

$$\arg\max_{f_{inv}} I(G', Y) - \beta I(G, G'), \text{s.t. } G' = f_{inv}(G), G' \subseteq G \tag{4}$$

Then, we can involve the time. Leveraging the sequential nature of temporal graphs as we discussed in the preliminary, we re-formulate the objective as:

$$\arg\max_{e_1,\ldots,e_T} I(\{e\}_1^T; Y_{T+1}) - \beta I(\{e\}_1^T; \{G\}_1^T) \tag{5}$$

where $\{e\}_1^T = \{e^1, \ldots, e^T\}, \{G\}_1^T = \{G^1, \ldots, G^T\}$. For the notation clarity, e^t means a bunch of selected edges that appeared at time t, omitting the node index in the subscript.

However, directly optimizing Eq. 5 is intractable [2], so we introduce approximated variational bounds that allow us to achieve Eq. 5 with trainable neural architectures (details in Section 3.3). Specifically, optimizing Eq. 5 is equivalent to:

$$\arg\min_{e_1,\ldots,e_T} -I(\{e\}_1^T; Y_{T+1}) + \beta I(\{e\}_1^T; \{G\}_1^T) \tag{6}$$

Then, we can achieve Eq. 6 by establishing an upper bound and minimizing this upper bound with trainable neural architectures.

Next, we elaborate on how to obtain the upper bound for each component in Eq. 6 in the following two subsections.

3.2.1 Minimizing $-I(\{e\}_1^T; Y_{T+1})$

In Eq. 6, we can have the first term as $-I(\{e\}_1^T; Y_{T+1})$.

Then, it is easy to show that $-I(\{e\}_1^T; Y_{T+1}) \leq -\log(q_{\phi_1}(Y_{T+1}|\{e\}_1^T))$ (*proof in Appendix*), where $q_{\phi_1}(Y_{T+1}|\{e\}_1^T)$ is the variational approximation of the probability $p(Y_{T+1}|\{e\}_1^T)$.

Thus, if we parameterize q_{ϕ_1} with a neural architecture, we could minimizing $-\log(q_{\phi_1}(Y_{T+1}|\{e\}_1^T))$ by incorporating this term into the model's loss function, and thus minimizing $-I(\{e\}_1^T; Y_{T+1})$.

3.2.2 Minimizing $\beta I(\{e\}_1^T; \{G\}_1^T)$

In Eq. 6, we can have the second term as $\beta I(\{e\}_1^T; \{G\}_1^T)$.

Also, we can show that $\beta I(\{e\}_1^T; \{G\}_1^T) \leq \beta \sum_{t=1}^{T} \mathcal{D}_{KL}(p_{\phi_2}(e_t|G_t, \{e\}_1^{t-1})||q_{\phi_3}(e_t|\{e\}_1^{t-1}))$ (*proof in Appendix*), where $\mathcal{D}_{KL}$ denotes the KL-divergence.

Again, we parameterize the variational approximation of $p(e_t|G_t, \{e\}_1^{t-1})$ and the prior $p(e_t|E^{t-1})$ with p_{ϕ_2}, q_{ϕ_3}, respectively. In this way, we can integrate the upper bound into the model's loss function and minimize it, thus minimizing $\beta I(\{e\}_1^T; \{G\}_1^T)$.

3.3 Selection Process and Trainable Neural Architectures

Here, we introduce how to realize the invariant link selection (or invariant subgraph construction) through a learnable manner, such that OOD-Linker can achieve generalizable and adaptive invariant learning and serve for effective specific tasks like temporal link prediction.

3.3.1 Query Link and its Computational Subgraph

To extract the invariant subgraph for a query link, we first need to define the scope for the selection. Thus, we define a computational subgraph $\widetilde{G}_{u,v}$ for a query link $(u, v, T+1)$ as follows.

The computational graph $\widetilde{G}_{u,v}$ should be close and thus is supposed to be the L-hop neighborhood graph contains links that lie within L-hop from edge (u, v) at any time before the query time, i.e., any link within L-hop of edge (u, v, t) in the time window, $t \in \{1, \ldots, T\}$. However, pre-defining a L for all nodes across time and structure is not feasible or adaptive for the OOD setting. Therefore, we propose to adaptively learn this computational graph.

Thus, we specify the neural architecture (i.e., parameterization) for p_{ϕ_1}, p_{ϕ_2}, and q_{ϕ_3}. We first introduce p_{ϕ_2}, q_{ϕ_3} for the invariant generalization and then p_{ϕ_1} for the task-specific label prediction, i.e., temporal link prediction task.

3.3.2 Parameterize p_{ϕ_2}, q_{ϕ_3}

Firstly, we model p_{ϕ_2} as the process of choosing a link at a certain time t as an invariant link.

Formally, for a certain timestamp t, constructing the invariant subgraph at time t can be modeled as the iterative process of modeling $p(e_t|\widetilde{G}^t, \{e\}_1^{t-1})$, i.e., the probability of choosing an edge as an invariant link, using neural network p_{ϕ_2}, given the current interactions in computational graph $\widetilde{G}^t$ and previous invariant subgraph $\{e\}_1^{t-1}$.

In detail, for an arbitrary link $(a, b, t) \in \widetilde{G}_{u,v}$, p_{ϕ_2} first maps the link to a latent representation, and then outputs the probability that (a, b, t) is chosen as an invariant link. We start by obtaining the node representation for nodes a, b, as Eq 11, by firstly aggregating information from previous invariant links and from current neighbor links, then further processing this representation with MLP transformations to obtain the probability of choosing link (a, b, t) as an invariant link. Mathematically, the computational process could be described as follows.

$$\begin{aligned}\widehat{\mathbf{h}}^t_{a,N,\phi_2} &= \sum_{(w,t')\in\mathcal{N}(a)|t'<t} p_{(a,w,t')}[\mathbf{s}^{t'}_w||f_{\text{time}}(t-t')||\mathbf{e}^{t'}_{(w,a)}] \\ &\quad + \sum_{(w,t)\in\mathcal{N}(a)} [\mathbf{s}^t_w||f_{\text{time}}(0)||\mathbf{e}^t_{(w,a)}], \text{ i.e.,} \\ \widetilde{\mathbf{h}}^t_{a,N,\phi_2} &= \mathbf{W}^{(2)}_{(agg),\phi_2}\Bigg(\text{RELU}\Bigg(\mathbf{W}^{(1)}_{(agg),\phi_2}\widehat{\mathbf{h}}^t_{a,N,\phi_2}\Bigg)\Bigg)\end{aligned} \tag{7}$$

where $[.||.]$ denotes concatenation, f_{time} is a time encoding function to obtain the vector representation of time t [54], $\mathcal{N}(a)$ denotes the neighbors of node a, $\mathbf{s}^t_w$ denotes the node feature of node w at time t, $\mathbf{e}^t_{(w,a)}$ denotes the edge feature of edge (w, a) at time t, and $p_{(a,w,t')}$ is the probability that (a, w, t') is chosen as an invariant link, modeled by p_{ϕ_2}.

After deriving the neighborhood aggregated representation for node a, we can obtain the node representation for a as:

$$\mathbf{h}^t_{a,N,\phi_2} = \mathbf{s}^t_a + \tanh\Bigg(\widetilde{\mathbf{h}}^t_{a,N,\phi_2} + \mathbf{W}_{\phi_2}\mathbf{s}^t_a\Bigg) \tag{8}$$

In the same way, we can obtain the node representation for node b, i.e., $\mathbf{h}^t_{b,N}$.

Next, we derive the probability that (a, b, t) is chosen as an invariant link by further applying MLP transformations on the node representations $\mathbf{h}^t_{a,N}$ and $\mathbf{h}^t_{b,N}$:

$$\begin{aligned}&\widehat{p}_{\phi_2}\big((a, b, t)\big) = \\ &\mathbf{W}^{(3)}_{\phi_2}\text{RELU}\Bigg(\mathbf{W}^{(2)}_{\phi_2}\text{RELU}\Bigg(\mathbf{W}^{(1)}_{\phi_2}[\mathbf{h}^t_{a,N,\phi_2}||\mathbf{h}^t_{b,N,\phi_2}]\Bigg)\Bigg)\end{aligned} \tag{9}$$

where $\mathbf{W}^{(3)}_{\phi_2}$, $\mathbf{W}^{(2)}_{\phi_2}$, and $\mathbf{W}^{(1)}_{\phi_2}$ are MLPs.

After obtaining the logits $\widehat{p}(\cdot)$, we apply the Sigmoid function and, inspired by the Gumbel-Softmax parameterization trick, we incorporate the coefficient to control how "soft" the probability is. Ideally, we would want the probability to be "hard", i.e., close to 0 or 1, which is equivalent to choosing the link or discarding the link, as later on, we only want our model to aggregate information along invariant links. Thus, the probability is refined as follows.

$$\begin{aligned} &p_{\phi_2}((a,b,t) \in \text{invariant subgraph}) = \\ &\text{SIGMOID}(\widehat{p}_{\phi_2}((a,b,t))/\tau) \end{aligned} \tag{10}$$

where τ is the control coefficient, the lower τ is the "harder" the probability distribution is.

Secondly, we elaborate how we can model the prior $q(e_t|\{e\}^1_{t-1})$ by parameterized the distribution with a neural architecture q_{ϕ_3}. Intuitively $q(e_t|\{e\}^t_{t-1})$ tells us about the probability of choosing an invariant link given we know that invariant link at previous timestamps. Therefore, similar to the parameterization p_{ϕ_2}, we would derive node representation by neighborhood aggregation and derive a probability indicating that whether an edge contributes to the invariant subgraph or not. We formulate the computational process as follows.

$$\begin{aligned} \widehat{\mathbf{h}}^t_{a,N,\phi_3} &= \sum_{(w,t')\in\mathcal{N}(a)|t'<t} p_{(a,w,t')}[\mathbf{s}^{t'}_w||f_{\text{time}}(t-t')||\mathbf{e}^{t'}_{(w,a)}] \\ \widetilde{\mathbf{h}}^t_{a,N,\phi_3} &= \mathbf{W}^{(2)}_{(agg),\phi_3}\Big(\text{RELU}\Big(\mathbf{W}^{(1)}_{(agg),\phi_3}\widehat{\mathbf{h}}^t_{a,N,\phi_3}\Big)\Big) \\ \mathbf{h}^t_{a,N,\phi_2} &= \mathbf{s}^t_a + \tanh\Big(\widetilde{\mathbf{h}}^t_{a,N,\phi_3} + \mathbf{W}_{\phi_3}\mathbf{s}^t_a\Big) \end{aligned} \tag{11}$$

Similar to modeling p_{ϕ_2} through Eq. 9 and Eq. 10, we obtain the probability with q_{ϕ_3} as follows:

$$\begin{aligned} &\widehat{p}_{\phi_3}\big((a,b,t)\big) = \\ &\mathbf{W}^{(3)}_{\phi_2}\text{RELU}\Big(\mathbf{W}^{(2)}_{\phi_2}\text{RELU}\Big(\mathbf{W}^{(1)}_{\phi_1}[\mathbf{h}^t_{a,N,\phi_2}||\mathbf{h}^t_{b,N,\phi_2}]\Big)\Big) \\ &\text{and} \\ &q_{\phi_3}((a,b,t) \in \text{invariant subgraph}) = \\ &\text{SIGMOID}(\widehat{p}_{\phi_3}((a,b,t)/\tau) \end{aligned} \tag{12}$$

3.3.3 Parameterize p_{ϕ_1}

Finally, we specify the neural architecture for the link predictor, $p_{\phi_1}(Y_{T+1}|\{e\}^1_T)$. Intuitively, we develop a neural architecture that predicts the link occurrence $(u, v, T+1)$ based on ALL invariant links in its computational graph.

Thus, we first derive the node representation for node u as follows:

$$\begin{aligned} \widehat{\mathbf{h}}^T_{a,N,\phi_1} &= \sum_{(w,t')\in\mathcal{N}(a)|t'<T} p_{(a,w,t')}[\mathbf{s}^{t'}_w||f_{\text{time}}(t-t')||\mathbf{e}^{t'}_{(w,a)}] \\ &\quad + \sum_{(w,t)\in\mathcal{N}(a)} [\mathbf{s}^t_w||f_{\text{time}}(0)||\mathbf{e}^t_{(w,a)}] \\ \widetilde{\mathbf{h}}^t_{a,N,\phi_1} &= \mathbf{W}^{(2)}_{(agg),\phi_1}\Big(\text{RELU}\Big(\mathbf{W}^{(1)}_{(agg),\phi_1}\widehat{\mathbf{h}}^t_{a,N}\Big)\Big) \\ \mathbf{h}^t_{a,N,\phi_1} &= \mathbf{s}^t_a + \tanh\Big(\widetilde{\mathbf{h}}^t_{a,N,\phi_1} + \mathbf{W}_{\phi_1}\mathbf{s}^t_a\Big) \end{aligned} \tag{13}$$

and the link prediction is made by

$$\widehat{y} = \text{sigmoid}(\mathbf{W}^{(2)}\text{RELU}(\mathbf{W}^{(1)}[(\mathbf{h}^t_u)^\top \mid\mid (\mathbf{h}^t_v)^\top]^\top)) \tag{14}$$

where we also employ the sigmoid function as we are performing binary classification.

3.4 Optimization of OOD-Linker

As discussed above, we first minimize the information bottleneck objective by minimizing the component's upper bound and then minimize the temporal link prediction risk. Therefore, we can now introduce the entire loss function as follows. Given N query links, we derive the loss function as:

$$\begin{aligned} \mathcal{L} &= \frac{1}{N}\sum_{i=1}^{N} -\log(p_{\phi_1}(Y_{i,T+1}|\{e\}^T_{1,i})) \\ &+ \sum_{i=1}^{N}\sum_{t=1}^{T} \mathcal{D}_{KL}(p_{\phi_2}(e_t|\widetilde{G}^t_i, \{e\}^{t-1}_{1,i})||q_{\phi_3}(e_t|\{e\}^{t-1}_{1,i})) \end{aligned} \tag{15}$$

where the first item is for temporal link prediction, and the second item is for invariant learning.

Moreover, we denote $Y_{i,T+1}, \widetilde{G}^t_i, \{e\}^t_{1,i}$ as the label, computational graph, and invariant links corresponding to the i-th query link, respectively.

More comprehensive algorithmic training procedures are presented in with pseudo-code in Appendix F.

4 Theoretical Analysis

In this section, we establish an upper bound for the error difference between applying OOD-Linker on a training distribution μ and testing distribution ν over $\mathcal{G} \times \mathcal{Y}$, where $\mathcal{G}, \mathcal{Y}$ is the space of computational graphs and labels (i.e., we can draw the computational graph of a query link and the label indicating the link occurrence from μ, ν) as follows.

Theorem 1. *Given N query links and their respective computational graphs, $\widetilde{G}_1, \ldots, \widetilde{G}_N$, and an $\alpha-$ Lipschitz and $\sigma-$sub-Gaussian loss function ℓ, then with probability at least $1-\delta > 0$, we have*

$$\left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \mathbb{E}_\nu[\ell(f(\widetilde{G}')), Y']\right| \le \mathcal{O}\Bigg(\frac{1}{N}\sum_{i=1}^{N}\sqrt{2\sigma^2 I(\phi(\widetilde{G}_i), \widetilde{G}_i) + D_{KL}(\mu||\nu)} + \sqrt{\frac{\log(1/\delta)}{N}}\Bigg) \tag{16}$$

where f is our neural architecture OOD-Linkerwith $\phi(G_i)$ denoting the invariant subgraph extracted by OOD-Linker (Proof in Appendix).

In Eq. 16, the expectations on the left hand side are taken over all $(\widetilde{G}, Y)$ drawn from μ and $(\widetilde{G}', Y')$ drawn from ν. Notably, our OOD-Linker seeks to extract a predictive subgraph, while constraining the mutual information between the invariant subgraph and the original computational graph, which is equivalent to constraining $I(\phi(\widetilde{G}_i), \widetilde{G}_i)$, and thus constraining the error difference between two distributions. Detailed proof for Theorem 1 can be founded in Appendix B.

Moreover, complexity analysis for our OOD-Linker, and comparison between our and other method's time complexity are presented in Appendix. E.

5 Experiments

In this section, we present the performance of OOD-Linker on the temporal link prediction task under several different distribution shift settings, i.e., shifts in edge attributes and shifts in node attributes; then, we conduct an ablation study to demonstrate the robustness of the Invariant Link Selector. We provide the reproducibility details like hyperparameters, and computing resources for OOD-Linker in the Appendix D.4.

5.1 Experimental Settings

Datasets and Baselines. We examine the ability of OOD-Linker in performing temporal link prediction with 3 classic real-world OOD datasets, COLLAB [44], ACT [14], and Aminer [45, 43] (data statistics are presented in Appendix D.1), and compare OOD-Linker against a range of baselines: (1) Static GNNs, including GAE [10], VGAE [10], are GCN[11]-based autoencoders for static graphs; (2) Dynamic GNNs, including GCRN [41], EvolveGCN [33], DySAT [40]; (3) OOD Generalization methods, including IRM [1], V-REx [12], GroupDRO [39]; (4) Dynamic Graph OOD Generalization methods, including DIDA [62], EAGLE [59], SILD [64], I-DIDA [63].

Table 1: Temporal Link Prediction (ROC) in the Edge OOD Setting.

Dataset	COLLAB	ACT
GAE	74.04 ± 0.75	60.27 ± 0.41
VGAE	74.95 ± 1.25	66.29 ± 1.33
GCRN	69.72 ± 0.45	64.35 ± 1.24
EvolveGCN	76.15 ± 0.91	63.17 ± 1.05
DySAT	76.59 ± 0.20	66.55 ± 1.21
IRM	75.42 ± 0.87	69.19 ± 1.35
V-REx	76.24 ± 0.77	70.15 ± 1.09
GroupDRO	76.33 ± 0.29	74.35 ± 1.62
DIDA	81.87 ± 0.40	78.64 ± 0.97
EAGLE	<u>84.41 ± 0.87</u>	<u>82.70 ± 0.72</u>
OOD-Linker (OURS)	**85.30 ± 0.31**	**85.98 ± 1.00**

Empirical evaluation details. Our experimental environment strictly follows the standard of SOTA OOD baselines [62, 59, 64]. We define the out-of-distribution dataset as follows. Each dataset of COLLAB and ACT has edge attributes, so the *out-of-distribution testing environment* is obtained by filtering out one certain link attribute of the original testing set. Then, the original unfiltered testing set forms an *in-distribution testing environment.* Further, for the originally given input graph, we can obtain the train/validate/test set by performing chronological splits (10/1/5 for COLLAB and 20/2/8 for ACT). To be more specific, the data splits are based on the number of distinct timestamps of the graph. For example, 10/1/5 for COLLAB indicates that we retrieve the temporal graph snapshots of the first 10 timestamps, and the temporal graph snapshot corresponding to the 11−th timestamp is the validation set. Then, we train and validate the models and select the models with the best validation score for testing them on both out-of-distribution and in-distribution testing sets. We elaborate more details on obtaining the out-of-distribution data and illustrate their distribution shifts in the Appendix D.2, AppendixD.3.

5.2 Link Prediction with Edge Attribute Shift

In Table 1, we report the average metric score, ROC, and the standard deviation on the testing out-of-distribution dataset. We obtain the average and standard deviation by evaluating OOD-Linker on 5 different runs. Best OOD testing ROC score is emphasize with **bold**, and the second-best is highlighted with <u>underline</u>. As suggested by Table 1, OOD-Linker achieves the best performance on the OOD testing set, and especially for the ACT dataset, OOD-Linker yields substantial improvements, compared to the second-best baseline, EAGLE [59]. Additionally, we report the performance comparison between our OOD-Linkerand other Dynamic Graph OOD Generalization methods, including

Table 2: Temporal Link Prediction (ROC) in the Node OOD Setting.

Dataset	COLLAB ($\bar{p} = 0.4$)	COLLAB ($\bar{p} = 0.6$)	COLLAB ($\bar{p} = 0.8$)
GCRN	70.24 ± 1.26	64.01 ± 0.19	62.19 ± 0.39
IRM	69.40 ± 0.09	63.97 ± 0.37	62.66 ± 0.33
V-REx	70.44 ± 1.08	63.99 ± 0.21	62.21 ± 0.40
GroupDRO	70.30 ± 1.23	64.05 ± 0.21	62.13 ± 0.35
DIDA	85.20 ± 0.84	82.89 ± 0.23	72.59 ± 3.31
EAGLE	**88.32 ± 0.61**	**87.29 ± 0.71**	**82.30 ± 0.75**
SILD	85.95 ± 0.18	84.69 ± 1.18	78.01 ± 0.71
I-DIDA	85.27 ± 0.06	83.00 ± 1.08	74.87 ± 1.59
OOD-Linker (OURS)	85.58 ± 1.54	83.09 ± 1.82	79.83 ± 1.69

DIDA [62], EAGLE [59], and SILD [64] for Aminer in Table 6 in Appendix D.5.

5.3 Link Prediction with Node Attribute Shift

To obtain the Node OOD setting, we follow the same pre-processing as SOTA baselines [62], [59], [64], which are modifications from COLLAB for exhibiting node features shift. More details on how to obtain the synthetic data are provided in the Appendix.

We report the average ROC score and the standard deviation on each synthetic testing dataset in Table 2. We highlighted the best and second-best results with **bold** and underline, respectively. As Table 2 suggests, our model achieves competitive results under distribution shifts of node features, as we have the second-best result on COLLAB ($\bar{p} = 0.8$), and have a close gap compared to the second-best results on COLLAB ($\bar{p} = 0.4$) and COLLAB ($\bar{p} = 0.6$).

5.4 Ablation Study of Invariant Link Selector

In this section, we provide insights into the generability of OOD-Linker under distribution shifts by comparing the link prediction task loss values between (1) OOD-Linker (i.e., using selected invariant links) and (2) using all links in the computational graph (i.e., omitting the invariant links discovery process).

Specifically, we present the trend of the link prediction task loss of using selected invariant links and using all links on the training and edge OOD validation sets of COLLAB and ACT in Figure 1 and Figure 2, respectively. In the figures, the x-axis represents the number of training epochs, and the y-axis shows the epoch-respective link prediction task loss.

As suggested by Figure 1 and Figure 2, the task loss of both methods on the training set decreases as the training proceeds, suggesting both models improve on prediction links in the training dataset. However, from the plot of task loss on the edge OOD validation set, we can see the gap, especially a substantial one from COLLAB, between 2 methods. Notably, OOD-Linker yields lower task loss on most of the epochs, suggesting that making predictions with invariant links is more robust to distribution shifts, while using all links hurts the performance in the dataset with distribution shift.

6 Related Work

Out-of-Distribution generalization on graphs is a critical problem in graph machine learning where model performance degrades due to distribution shifts between testing and training graph data [18, 9, 24, 49, 68, 53]. In [62], the authors propose a dynamic graph neural network called DIDA to handle spatio-temporal distribution shifts in dynamic graphs. To discover variant and invariant spatio-temporal patterns in dynamic graphs, DIDA first uses a disentangled spatio-temporal attention network to encode variant and invariant patterns. It then proposes a spatio-temporal intervention mechanism to create multiple intervened distributions by sampling and reassembling variant patterns across neighborhoods and timestamps. Finally, an invariance regularization term is used to minimize the variance of predictions in the intervened distributions, enabling the model to focus on invariant patterns with stable predictive abilities to handle distribution shifts. Motivated by DIDA [62], I-DIDA [63] is proposed. Additionally, I-DIDA further promotes the invariance property by inferring latent spatio-temporal environments and minimizing prediction variance among them.

Very recently, some new efforts have been shown to address the out-of-distribution problems in the dynamic graph deep learning community. A method called Spectral Invariant Learning for Dynamic Graphs under Distribution Shifts (SILD) [64] is proposed from the spectral domain. SILD addresses two key challenges: capturing different graph patterns driven by various frequency components in the spectral domain, and handling distribution shifts with the discovered spectral patterns. Environment-Aware Dynamic Graph Learning (EAGLE) [59] framework is also designed for

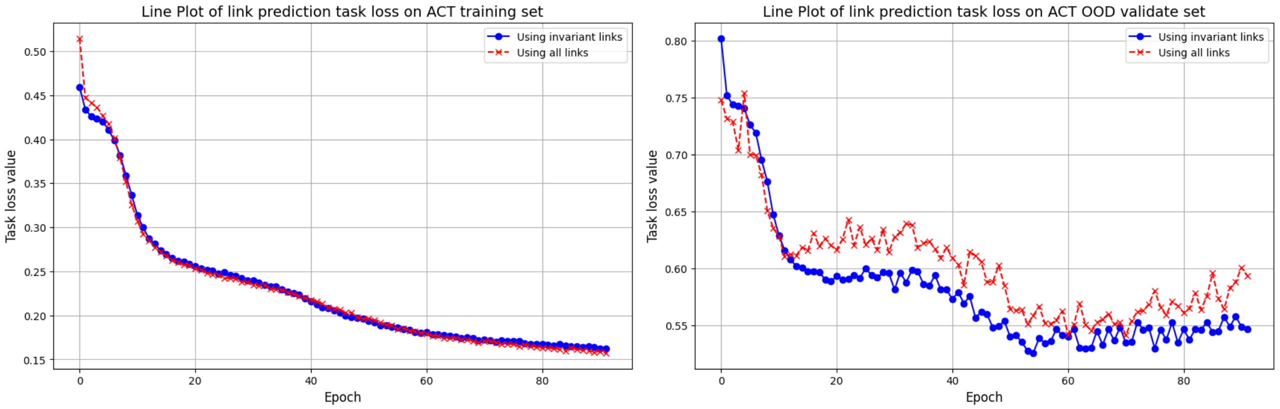

Figure 1: Comparison of link prediction task loss on the training and edge OOD settings of ACT.

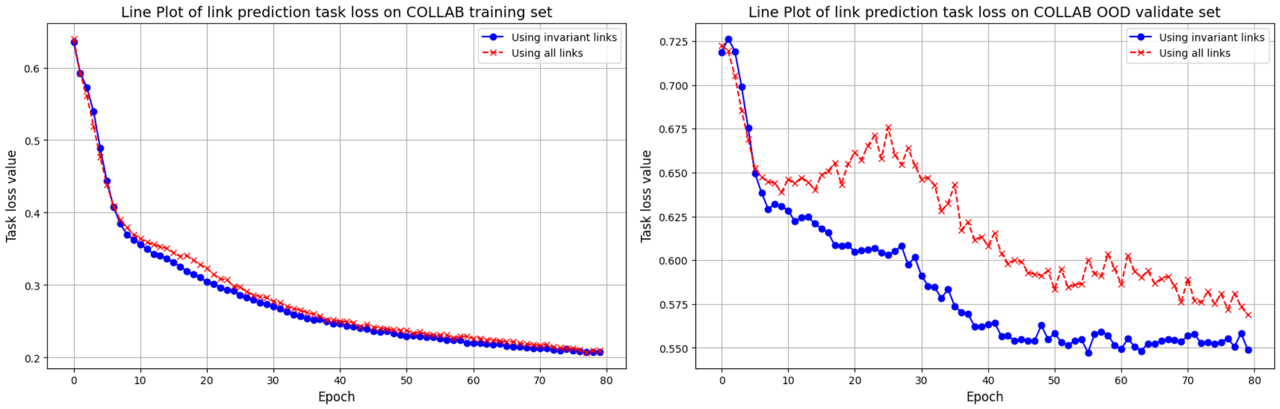

Figure 2: Comparison of link prediction task loss on the training and edge OOD settings of COLLAB.

out-of-distribution generalization on dynamic graphs. EAGLE addresses how to model and infer the complex environments on dynamic graphs with distribution shifts, and how to discover invariant patterns given inferred spatio-temporal environments. EAGLE contains an environment-aware graph neural network to model environments by multi-channel environments disentangling. Then EAGLE uses an environment instantiation mechanism for environment diversification with inferred distributions. Finally, EAGLE performs fine-grained causal interventions node-wisely with a mixture of instantiated environment samples to generalize to diverse environments. Different from previous methods on Dynamic Graphs OOD Generalization, our OOD-Linker effectively extracts the representations that are robust to distribution shifts by looking back into the historical interactions, while other previous methods obtain invariant representations by causal intervention, which is prone to be computationally expensive. Moreover, we also establish a provable error bound, theoretically justifying OOD-Linker's robustness under distribution shifts. To the best of our knowledge, we are the first effort to establish such a guarantee on Dynamic Graphs OOD Generalization. More detailed discussion on Invariant Learning and Information Bottleneck-based methods can be found in Appendix C.

7 Conclusion

In this work, we aim to advance the field of temporal graph learning and contribute to the development of more robust and generalizable deep learning models capable of handling the complex, dynamic nature of real-world relational data, i.e., spatial-temporal graphs. Hence, we propose the error-bounded Invariant Link Selector technique based on the Information Bottleneck method. Moreover, we incorporate the theoretical contribution into a concrete framework OOD-Linker for dealing with temporal link prediction tasks in domain-shift settings, evaluated by extensive verification scenarios with SOTA baseline algorithms.

Acknowledgements

This work is supported by National Science Foundation under Award No. IIS-2117902, and the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-08-00. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

[1] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 3438–3450, 2021.

[2] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[3] Muhammad Awais, Muzammal Naseer, Salman H. Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundational models defining a new era in vision: A survey and outlook. *CoRR*, abs/2307.13721, 2023.

[4] Yikun Ban, Jiaru Zou, Zihao Li, Yunzhe Qi, Dongqi Fu, Jian Kang, Hanghang Tong, and Jingrui He. Pagerank bandits for link prediction. In *NeurIPS*, 2024.

[5] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[6] Dongqi Fu, Liri Fang, Ross Maciejewski, Vetle I. Torvik, and Jingrui He. Meta-learned metrics over multi-evolution temporal graphs. In Aidong Zhang and Huzefa Rangwala, editors, *KDD*, 2022.

[7] Dongqi Fu and Jingrui He. SDG: A simplified and dynamic graph neural network. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai, editors, *SIGIR*, 2021.

[8] Dongqi Fu, Zhigang Hua, Yan Xie, Jin Fang, Si Zhang, Kaan Sancak, Hao Wu, Andrey Malevich, Jingrui He, and Bo Long. Vcr-graphormer: A mini-batch graph transformer via virtual connections. In *ICLR*, 2024.

[9] Wenzhao Jiang, Hao Liu, and Hui Xiong. When graph neural network meets causality: Opportunities, methodologies and an outlook, 2024.

[10] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

[11] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[12] David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5815–5826. PMLR, 2021.

[13] Solomon Kullback. Kullback-leibler divergence, 1951.

[14] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1269–1278. ACM, 2019.

[15] Vitaly Kuznetsov and Mehryar Mohri. Time series prediction and online learning. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 1190–1213. JMLR.org, 2016.

[16] Vitaly Kuznetsov and Mehryar Mohri. Discrepancy-based theory and algorithms for forecasting non-stationary time series. *Ann. Math. Artif. Intell.*, 88(4):367–399, 2020.

[17] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. Disentangled graph self-supervised learning for out-of-distribution generalization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[18] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *CoRR*, abs/2202.07987, 2022.

[19] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[20] Zihao Li, Dongqi Fu, Mengting Ai, and Jingrui He. Apex^2: Adaptive and extreme summarization for personalized knowledge graphs. *CoRR*, 2024.

[21] Zihao Li, Dongqi Fu, and Jingrui He. Everything evolves in personalized pagerank. In *WWW*, 2023.

[22] Xiao Lin, Zhining Liu, Dongqi Fu, Ruizhong Qiu, and Hanghang Tong. Backtime: Backdoor attacks on multivariate time series forecasting. In *NeurIPS*, 2024.

[23] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023.

[24] Shuhan Liu and Kaize Ding. Beyond generalization: A survey of out-of-distribution adaptation on graphs. *CoRR*, abs/2402.11153, 2024.

[25] Wenliang Liu, Guanding Yu, Lele Wang, and Renjie Liao. An information-theoretic framework for out-of-distribution generalization. In *IEEE International Symposium on Information Theory, ISIT 2024, Athens, Greece, July 7-12, 2024*, pages 2670–2675. IEEE, 2024.

[26] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4212–4221. PMLR, 2019.

[27] J Macqueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.

[28] Mehrdad Mahdavi. Exploiting smoothness in statistical learning, sequential prediction, and stochastic optimization. *CoRR*, abs/1407.5908, 2014.

[29] Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[30] Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15524–15543. PMLR, 2022.

[31] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[32] Atsuyuki Miyai, Jingkang Yang, Jingyang Zhang, Yifei Ming, Yueqian Lin, Qing Yu, Go Irie, Shafiq Joty, Yixuan Li, Hai Li, Ziwei Liu, Toshihiko Yamasaki, and Kiyoharu Aizawa. Generalized out-of-distribution detection and beyond in vision language model era: A survey. *CoRR*, abs/2407.21794, 2024.

[33] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *The Thirty-Fourth*

AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 5363–5370. AAAI Press, 2020.

[34] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96 – 146, 2009.

[35] Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *CoRR*, abs/2408.08921, 2024.

[36] Yunzhe Qi, Yikun Ban, and Jingrui He. Graph neural bandits. In *KDD*, 2023.

[37] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning via sequential complexities. *J. Mach. Learn. Res.*, 16:155–186, 2015.

[38] Vipula Rawte, Amit P. Sheth, and Amitava Das. A survey of hallucination in large foundation models. *CoRR*, abs/2309.05922, 2023.

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

[40] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang, editors, *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 519–527. ACM, 2020.

[41] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In Long Cheng, Andrew Chi-Sing Leung, and Seiichi Ozawa, editors, *Neural Information Processing - 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I*, volume 11301 of *Lecture Notes in Computer Science*, pages 362–373. Springer, 2018.

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

[43] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (MAS) and applications. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, pages 243–246. ACM, 2015.

[44] Jie Tang, Sen Wu, Jimeng Sun, and Hang Su. Cross-domain collaboration recommendation. In Qiang Yang, Deepak Agarwal, and Jian Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1285–1293. ACM, 2012.

[45] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 990–998. ACM, 2008.

[46] Katherine Tieu, Dongqi Fu, Yada Zhu, Hendrik F. Hamann, and Jingrui He. Temporal graph neural tangent kernel with graphon-guaranteed. In *NeurIPS*, 2024.

[47] Naftali Tishby, Fernando C. N. Pereira, and William Bialek. The information bottleneck method. *CoRR*, physics/0004057, 2000.

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.

[49] Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. STONE: A spatio-temporal OOD learning framework kills both spatial and temporal shifts. In Ricardo Baeza-Yates and Francesco Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 2948–2959. ACM, 2024.

[50] Limei Wang, Kaveh Hassani, Si Zhang, Dongqi Fu, Baichuan Yuan, Weilin Cong, Zhigang Hua, Hao Wu, Ning Yao, and Bo Long. Learning graph quantized tokenizers for transformers. *CoRR*, 2024.

[51] Jun Wu, Jingrui He, and Elizabeth A. Ainsworth. Non-iid transfer learning on graphs. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *AAAI*, 2023.

[52] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[53] Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[54] Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[55] Zhe Xu, Kaveh Hassani, Si Zhang, Hanqing Zeng, Michihiro Yasunaga, Limei Wang, Dongqi Fu, Ning Yao, Bo Long, and Hanghang Tong. Language models are graph learners. *CoRR*, 2024.

[56] Yuchen Yan, Lihui Liu, Yikun Ban, Baoyu Jing, and Hanghang Tong. Dynamic knowledge graph alignment. In *AAAI*, 2021.

[57] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *CoRR*, abs/2110.11334, 2021.

[58] Haonan Yuan, Qingyun Sun, Xingcheng Fu, Cheng Ji, and Jianxin Li. Dynamic graph information bottleneck. In Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee, editors, *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 469–480. ACM, 2024.

[59] Haonan Yuan, Qingyun Sun, Xingcheng Fu, Ziwei Zhang, Cheng Ji, Hao Peng, and Jianxin Li. Environment-aware dynamic graph learning for out-of-distribution generalization. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[60] Zhichen Zeng, Si Zhang, Yinglong Xia, and Hanghang Tong. PARROT: position-aware regularized optimal transport for network alignment. In *WWW*, 2023.

[61] Zhichen Zeng, Ruike Zhu, Yinglong Xia, Hanqing Zeng, and Hanghang Tong. Generative graph dictionary learning. In *ICML*, 2023.

[62] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[63] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, and Wenwu Zhu. Out-of-distribution generalized dynamic graph neural network with disentangled intervention and invariance promotion. *CoRR*, abs/2311.14255, 2023.

[64] Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. Spectral invariant learning for dynamic graphs under distribution shifts. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[65] Lecheng Zheng, Dongqi Fu, Ross Maciejewski, and Jingrui He. Drgnn: Deep residual graph neural network with contrastive learning. In *TMLR*, 2024.

[66] Lecheng Zheng, Baoyu Jing, Zihao Li, Zhichen Zeng, Tianxin Wei, Mengting Ai, Xinrui He, Lihui Liu, Dongqi Fu, Jiaxuan You, Hanghang Tong, and Jingrui He. Pyg-ssl: A graph self-supervised learning toolkit. *CoRR*, 2024.

[67] Dawei Zhou, Lecheng Zheng, Dongqi Fu, Jiawei Han, and Jingrui He. Mentorgnn: Deriving curriculum for pre-training gnns. In Mohammad Al Hasan and Li Xiong, editors, *CIKM*, 2022.

[68] Zhengyang Zhou, Qihe Huang, Kuo Yang, Kun Wang, Xu Wang, Yudong Zhang, Yuxuan Liang,

and Yang Wang. Maintaining the status quo: Capturing invariant relations for OOD spatiotemporal learning. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye, editors, *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 3603–3614. ACM, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Proofs for Variational Bounds and Derivation of Loss function

A.1 Minimizing $-I(\{e\}_1^T; Y_{T+1})$

For an arbitrary $t \leq T-1$, we have

$$\begin{aligned} I(\{e\}_1^t; Y_{T+1}) &= I(e_t, \{e\}_1^{t-1}; Y_{T+1}) \\ &= I(e_t; Y_{T+1} \mid \{e\}_1^{t-1}) + I(\{e\}_1^{t-1}; Y_{T+1}) \end{aligned} \tag{17}$$

Thus, we have

$$\begin{aligned} I(\{e\}_1^T; Y_{T+1}) &= I(e_T, \{e\}_1^{T-1}; Y_{T+1}) \\ &= I(e_T; Y_{T+1} \mid \{e\}_1^{T-1}) + I(\{e\}_1^{T-1}; Y_{T+1}) \\ &= I(e_T; Y_{T+1} \mid \{e\}_1^{T-1}) + I(e_{T-1}; Y_{T+1} \mid \{e\}_1^{T-2}) + I(\{e\}_1^{T-2}; Y_{T+1}) \\ &= \dots \\ &= I(e_T; Y_{T+1} \mid \{e\}_1^{T-1}) + I(e_{T-1}; Y_{T+1} \mid \{e\}_1^{T-2}) + \dots + I(\{e\}_1^1; Y_{T+1}) \end{aligned} \tag{18}$$

Moreover, for an arbitrary t, we have

$$\begin{aligned} &I(e_t; Y_{T+1} \mid \{e\}_1^{t-1}) = \iiint P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \log \frac{P(e_t, Y_{T+1} \mid \{e\}_1^{t-1})}{P(e_t \mid \{e\}_1^{t-1}) P(Y_{T+1} \mid \{e\}_1^{t-1})} \\ &= \iiint P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \log \frac{P(Y_{T+1} \mid e_t, \{e\}_1^{t-1}) P(e_t \mid \{e\}_1^{t-1})}{P(e_t \mid \{e\}_1^{t-1}) P(Y_{T+1} \mid \{e\}_1^{t-1})} \\ &= \iiint P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \log \frac{P(Y_{T+1} \mid e_t, \{e\}_1^{t-1})}{P(Y_{T+1} \mid \{e\}_1^{t-1})} \\ &= \iiint P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \log \Big(P(Y_{T+1} \mid e_t, \{e\}_1^{t-1}) \Big) - \iiint P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^{t-1}) \Big) \\ &= \iiint P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^{t}) \Big) - \iint \log \Big(P(Y_{T+1} \mid \{e\}_1^{t-1}) \Big) \Big(\int P(e_t, \{e\}_1^{t-1}, Y_{T+1}) \Big) \\ &= \iint P(\{e\}_1^t, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^t) \Big) - \iint P(\{e\}_1^{t-1}, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^{t-1}) \Big) \end{aligned} \tag{19}$$

Thus the right-hand side of Eq. 18 is equivalent to

$$\begin{aligned} &I(e_T; Y_{T+1} \mid \{e\}_1^{T-1}) + I(e_{T-1}; Y_{T+1} \mid \{e\}_1^{T-2}) + \dots + I(\{e\}_1^1; Y_{T+1}) = \\ &= \iint P(\{e\}_1^T, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^T) \Big) - \iint P(\{e\}_1^1, Y_{T+1}) \log \Big(P(Y_{T+1}) \Big) \\ &= \iint P(\{e\}_1^T, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^T) \Big) - \int P(Y_{T+1}) \log \Big(P(Y_{T+1}) \Big) \\ &= \iint P(\{e\}_1^T, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^T) \Big) + H(Y_{T+1}) \\ &\geq \iint P(\{e\}_1^T, Y_{T+1}) \log \Big(P(Y_{T+1} \mid \{e\}_1^T) \Big) \end{aligned} \tag{20}$$

where $H(.)$ is the entropy.

However, since $P(Y_{T+1} \mid \{e\}_1^T)$ is intractable [2], so let $q_{\phi_1}(Y_{T+1} \mid \{e\}_1^T)$ be a variational approximation to $p(Y_{T+1} \mid \{e\}_1^{T+1})$. We have

$$\begin{aligned}
&D_{KL}(p(Y_{T+1} \mid \{e\}_1^T) \mid\mid q_{\phi_1}(Y_T \mid \{e\}_1^T)) \geq 0 \\
&\iint p(\{e\}_1^T, Y_{T+1}) \log\Big(p(Y_{T+1} \mid \{e\}_1^T)\Big) \geq \iint p(\{e\}_1^T, Y_{T+1}) \log\Big(q_{\phi_1}(Y_{T+1} \mid \{e\}_1^T)\Big)
\end{aligned} \tag{21}$$

Therefore, we obtain the variational lower bound for $I(\{e\}_1^T; Y_{T+1})$ as follows:

$$\begin{aligned}
I(\{e\}_1^T; Y_{T+1}) &\geq \iint p(\{e\}_1^T, Y_{T+1}) \log\Big(p(Y_{T+1} \mid \{e\}_1^T)\Big) \\
&\geq \iint p(\{e\}_1^T, Y_{T+1}) \log\Big(q_{\phi_1}(Y_{T+1} \mid \{e\}_1^T)\Big)
\end{aligned} \tag{22}$$

Thus, $-I(\{e\}_1^T; Y_{T+1}) \leq -\iint p(\{e\}_1^T, Y_{T+1}) \log\Big(q_{\phi_1}(Y_{T+1} \mid \{e\}_1^T)\Big)$

Moreover, given N query links and their corresponding invariant subgraphs, $\{e\}_{1,1}^T, \ldots, \{e\}_{1,N}^T$, we can approximate the distribution $p(\{e\}_1^T, Y_{T+1})$ as $\sum_{i=1}^N \frac{1}{N} \delta_{Y_{i,T+1}} \delta_{\{e\}_{1,i}^T}$, where $Y_{i,T+1}$ is the ground-truth label of the $i-$ th query link. Thus

$$\iint p(\{e\}_1^T, Y_{T+1}) \log\Big(q_{\phi_1}(Y_{T+1} \mid \{e\}_1^T)\Big) \approx \frac{1}{N} \sum_{i=1}^N \log\Big(q_{\phi_1}(Y_{i,T+1} \mid \{e\}_1^T)\Big) \tag{23}$$

Therefore, given that we parameterize q_{ϕ_1} with a neural architecture, we can integrate $-\frac{1}{N}\sum_{i=1}^N \log\Big(q_{\phi_1}(Y_{i,T+1} \mid \{e\}_{1,i}^T)\Big)$ into the model's loss function, and thus minimizing this term leads to minimization of $-I(\{e\}_1^T; Y_{T+1})$, i.e., maximization of $I(\{e\}_1^T; Y_{T+1})$. Moreover, we refer to $-\frac{1}{N}\sum_{i=1}^N \log\Big(q_{\phi_1}(Y_{i,T+1} \mid \{e\}_{1,i}^T)\Big)$ as the link prediction task loss, and as link prediction is a binary classification task, we employ Binary Cross Entropy to compute this component.

A.2 Minimizing $\beta I(\{e\}_1^T; \{G\}_1^T)$

For arbitrary t, we have

$$\begin{aligned}
I(e_t; \{G\}_1^t \mid \{e\}_1^{t-1}) &= I(e_t; G^t, \{G\}_1^{t-1} \mid E_1^{t-1}) \\
&= I(e_t; G^t \mid \{e\}_1^{t-1}) + I(e_t; \{G\}_1^{t-1} \mid G^t, \{e\}_1^{t-1}) \\
&= I(e_t; G^t \mid \{e\}_1^{t-1}) \\
I(\{e\}_1^{t-1}; \{G\}_1^t) &= I(\{e\}_1^{t-1}; G^t, \{G\}_1^{t-1}) \\
&= I(\{e\}_1^{t-1}; \{G\}_1^{t-1}) + I(\{e\}_1^{t-1}; G^t \mid \{G\}_1^{t-1}) \\
&= I(\{e\}_1^{t-1}; \{G\}_1^{t-1})
\end{aligned} \tag{24}$$

As e_t is a subset of G^t, so e_t could be regarded as the result of a noisy function of G^t, i.e $e^t = f(G^t, \epsilon)$, with some noise ϵ. So when G^t is observed, e^t becomes conditionally independent with other variables, so $I(e_t; \{G\}_1^{t-1} \mid G^t, \{e\}_1^{t-1}) = 0$. The same reasoning applies for $I(\{e\}_1^{t-1}; G^t \mid \{G\}_1^{t-1}) = 0$, as we can consider $\{e\}_1^{t-1}$ as the result of a noisy function of $\{G\}_1^{t-1}$.

Moreover, for an arbitrary t, we obtain the upper bound for $I(e_t; G^t \mid \{e\}_1^{t-1})$ as follows. Firstly, we have

$$\begin{aligned}
I(e_t; G^t \mid \{e\}_1^{t-1}) &= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(\frac{p(e_t, G^t \mid \{e\}_1^{t-1})}{p(e_t \mid \{e\}_1^{t-1}) p(G^t \mid \{e\}_1^{t-1})}\right) \\
&= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(\frac{p(e_t \mid G^t, \{e\}_1^{t-1}) p(G^t \mid \{e\}_1^{t-1})}{p(e_t \mid \{e\}_1^{t-1}) p(G^t \mid \{e\}_1^{t-1})}\right) \\
&= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(\frac{p(e_t \mid G^t, \{e\}_1^{t-1})}{p(e_t \mid \{e\}_1^{t-1})}\right)
\end{aligned} \tag{25}$$

Let $q_{\phi_3}(e_t \mid \{e\}_1^{t-1})$ be a variational approximation to $p(e_t \mid \{e\}_1^{t-1})$, we have

$$\begin{aligned}
&D_{KL}(p(e_t \mid \{e\}_1^{t-1}) \mid\mid q_{\phi_3}(e_t \mid \{e\}_1^{t-1})) \geq 0 \\
\Leftrightarrow &\iint p(e_t, \{e\}_1^{t-1}) \log\left(p(e_t \mid \{e\}_1^{t-1})\right) \geq \iint p(e_t, \{e\}_1^{t-1}) \log\left(q_{\phi_3}(e_t \mid \{e\}_1^{t-1})\right)
\end{aligned} \tag{26}$$

Therefore, we derive the upper bound for the left-hand side of Eq. 25 as follows:

$$\begin{aligned}
&\iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(\frac{p(e_t \mid G^t, \{e\}_1^{t-1})}{p(e_t \mid \{e\}_1^{t-1})}\right) = \\
&= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(p(e_t \mid G^t, \{e\}_1^{t-1})\right) - \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(p(e_t \mid \{e\}_1^{t-1})\right) \\
&= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(p(e_t \mid G^t, \{e\}_1^{t-1})\right) - \iint p(e_t, \{e\}_1^{t-1}) \log\left(p(e_t \mid \{e\}_1^{t-1})\right) \\
&\leq \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(p(e_t \mid G^t, \{e\}_1^{t-1})\right) - \iint p(e_t, \{e\}_1^{t-1}) \log\left(q_{\phi_3}(e_t \mid \{e\}_1^{t-1})\right) \\
&= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(p(e_t \mid G^t, \{e\}_1^{t-1})\right) - \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(q_{\phi_3}(e_t \mid \{e\}_1^{t-1})\right) \\
&= \iiint p(e_t, G^t, \{e\}_1^{t-1}) \log\left(\frac{p(e_t \mid G^t, \{e\}_1^{t-1})}{q_{\phi_3}(e_t \mid \{e\}_1^{t-1})}\right)
\end{aligned} \tag{27}$$

Moreover, as we propose to parameterize $p(e_t \mid G^t, \{e\}_1^{t-1})$ with $q_{\phi_2}(e_t \mid G^t, \{e\}_1^{t-1})$, so we have $I(e_t; G^t \mid \{e\}_1^{t-1}) \leq D_{KL}(q_{\phi_2}(e_t \mid G^t, \{e\}_1^{t-1}) \mid\mid q_{\phi_3}(e_t \mid \{e\}_1^{t-1}))$.

Putting everything together, we obtain the upper bound for $I(\{e\}_1^T; \{G\}_1^T)$ as follows:

$$\begin{aligned}
I(\{e\}_1^T; \{G\}_1^T) &= I(e_T; \{G\}_1^T \mid \{e\}_1^{T-1}) + I(\{e\}_1^{T-1}; \{G\}_1^T) \\
&= I(e_T; G^T \mid \{e\}_1^{T-1}) + I(\{e\}_1^{T-1}; \{G\}_1^{T-1}) \\
&= \dots \\
&= I(e_T; G^T \mid \{e\}_1^{T-1}) + I(e_{T-1}; G^{T-1} \mid \{e\}_1^{T-2}) + \cdots + I(e_1; G^1) \\
&\leq \sum_{t=1}^{T} D_{KL}(q_{\phi_2}(e_t \mid G^t, \{e\}_1^{t-1}) \mid\mid q_{\phi_3}(e_t \mid \{e\}_1^{t-1}))
\end{aligned} \tag{28}$$

In this way, we can minimize $\beta I(\{e\}_1^T; \{G\}_1^T)$ by minimizing $\beta \sum_{t=1}^{T} D_{KL}(q_{\phi_2}(e_t \mid G^t, \{e\}_1^{t-1}) \mid\mid q_{\phi_3}(e_t \mid \{e\}_1^{t-1}))$, which could be achieved by incorporating this term into the model's loss function.

A.3 Optimization

Given that we have obtained the variational upper bounds, we can derive the loss function for OOD-Linker's optimization process as follows. Suppose we have N query links and their corresponding computational graphs

$G_1, \ldots, G_N$ and their respective invariant subgraphs $\{e\}_{1,1}^T, \ldots \{e\}_{1,N}^T$, then the loss function for our model would be

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} -\log\left(q_{\phi_1}(Y_{i,T+1} \mid \{e\}_{1,i}^T)\right) + \beta\sum_{t=1}^{T} D_{KL}(q_{\phi_2}(e_{t,i} \mid G^t, \{e\}_{1,i}^{t-1}) \mid\mid q_{\phi_3}(e_{t,i} \mid \{e\}_{1,i}^{t-1})) \tag{29}$$

Therefore, by minimizing $\mathcal{L}$, we achieve the maximization of $I(\{e\}_1^T; Y_{T+1})$ and minimization of $\beta I(\{e\}_1^T; \{G\}_1^T)$.

B Proof for Error Bound

In this section, we present the proof for the error bound stated in Theorem 1 in the main paper.

Proof. We start by decomposing the error difference into 2 terms and present the proof for bounding these terms as follows.

$$\begin{aligned}
&\left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \mathbb{E}_\nu[\ell(f(\widetilde{G}'), Y')]\right| = \\
&= \left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i) + \left(\frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i) - \mathbb{E}_\nu[\ell(f(\widetilde{G}'), Y')]\right)\right| \\
&\leq \left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i)\right| + \left|\frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i) - \mathbb{E}_\nu[\ell(f(\widetilde{G}'), Y')]\right|
\end{aligned} \tag{30}$$

The inequality holds due to Triangle Inequality. Next, we focus on bounding the two terms.

$$\left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i)\right| \text{ and } \left|\frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i) - \mathbb{E}_\nu[\ell(f(\widetilde{G}'), Y')]\right| \tag{31}$$

Firstly, we present how to upper-bound $\left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i)\right|$. Overall, we aim to bound this term by the Sequential Rademacher Complexity [15, 37] of the function class containing functions such as f, and further bound this term by known variables.

We can rewrite $E_\mu[\ell(f(\widetilde{G}), Y)]$ as $E_\mu[\ell(f(\widetilde{G}^{T+1}), Y_{T+1})|\widetilde{G}^1, \ldots \widetilde{G}^T]$ due to the sequential nature of temporal graphs. With probability at least $1-\delta$, we have

$$
\begin{aligned}
&\left|E_\mu[\ell(f(\widetilde{G}^{T+1}), Y_{T+1})|\widetilde{G}^1, \dots \widetilde{G}^T] - \frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i, Y_i))\right| \\
&= \left|E_\mu[\ell(f(\widetilde{G}^{T+1}), Y_{T+1})|\widetilde{G}^1, \dots \widetilde{G}^T] - \frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i, Y_i))\right| \\
&= \left|E_\mu[\ell(f(\widetilde{G}^{T+1}), Y_{T+1})|\widetilde{G}^1, \dots \widetilde{G}^T] - \frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{T}\ell(f(\widetilde{G}_i^t, Y_i))\right| \\
&\leq \mathcal{O}\left(\frac{1}{\sqrt{NT}} + M\mathfrak{R}_T^{\mathrm{seq}}(\ell \circ \mathcal{F}) + \frac{M}{\sqrt{NT}}\sqrt{\log(1/\delta)}\right) \\
&\leq \mathcal{O}\left(\alpha(\log T)^3\mathfrak{R}_T^{seq}(\mathcal{F}) + \frac{M\sqrt{\log(1/\delta)}+1}{\sqrt{NT}}\right) \\
&\leq \mathcal{O}\left(\alpha R(\log T)^3\sqrt{\frac{(\log T)^3}{T}} + \frac{M\sqrt{\log(1/\delta)}+1}{\sqrt{NT}}\right) \\
&= \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right)
\end{aligned} \tag{32}
$$

where $\mathcal{F}$ is the class containing functions such as f, and M is the upper bound of the loss function, R is the Lipschitz constant of MLPs transformations in neural architecture model f. The first inequality holds by applying Corollary 2 from [16], the second inequality holds by as $\mathfrak{R}_T^{seq}(\ell \circ \mathcal{F}) \leq \mathcal{O}\left(\alpha(\log T)^3\right)\mathfrak{R}_T^{seq}(\mathcal{F})$ as proven in [28], and the third inequality holds due to $\mathfrak{R}_T^{seq}(\mathcal{F}) \leq \mathcal{O}\left(R\sqrt{\frac{(\log T)^3}{T}}\right)$ as proven in [28].

Then, we obtain the upper bound for $\left|\frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i) - \mathbb{E}_\nu[\ell(f(\widetilde{G}'), Y')]\right|$ by directly apply results from [25] as follows:

$$
\left|\frac{1}{N}\sum_{i=1}^{N}\ell(f(\widetilde{G}_i), Y_i) - \mathbb{E}_\nu[\ell(f(\widetilde{G}), Y)]\right| \leq \mathcal{O}\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{2\sigma^2 I(\phi(\widetilde{G}_i), \widetilde{G}_i) + D_{KL}(\mu||\nu)}\right) \tag{33}
$$

Combining the 2 upper bounds, we derive the error bound as:

$$
\left|\mathbb{E}_\mu[\ell(f(\widetilde{G}), Y)] - \mathbb{E}_\nu[\ell(f(\widetilde{G}'), Y')]\right| \leq \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}} + \frac{1}{N}\sum_{i=1}^{N}\sqrt{2\sigma^2 I(\phi(\widetilde{G}_i), \widetilde{G}_i) + D_{KL}(\mu||\nu)}\right) \tag{34}
$$

□

C More detailed discussion on related works

Here, we provide discussion on more related works in Invariant Learning and comparison with methods leveraging Information Bottleneck (IB) for OOD Generalization on Graphs as follows.

Graph Invariant Learning methods aim to extract invariant graph patterns that could stabilize predictions under distribution shifts. For example, GIL [19] proposes to identify invariant subgraphs and infer environment labels for variant subgraphs through joint learning, GSAT [30] learns task-relevant subgraphs by constraining mutual information with stochasticity using attention weights.

Supported by Causal Inference theory, causal-based methods utilize Structural Causal Model (SCM) [34] to identify and filter out spurious correlations. For instance, DIR [52] filters out subgraphs that have spurious correlations with graph labels by leveraging a learnable mask, and then perform intervention with do-calculus to enhance model's ability in recognizing invariant rationales. CIGA [5] firsts identifying the causal subgraph using informative invariant features (FIIF) or partially informative invariant features (PIIF), and then achieves OOD generalization by identifying the causal subgraph that maximally preserves the intra-class information across different training environments.

Rooted in Disentanglement theories, disentangle-based methods aim at separating factors of variations in data, then proceed with learning representations by distinguishing invariant factors of the graph data. For example, DisenGCN [26] learns disentangled node representations by clustering neighbors into subspaces corresponding to distinct latent factors, allowing each channel to extract factor-specific features, and thus enhancing OOD Generalization. Most recently, OOD-GCL [17], a self-supervised disentangled graph contrastive learning model that achieves OOD generalization without leveraging graph labels, proposed a tailored invariant self-supervised learning module to distinguish invariant and variant factors.

Regarding notable **IB-based methods**, GSAT [30] and DGIB [58], we elaborate on the differences between our OOD-Linker and aforementioned methods as follows. Although GSAT [30] also extracts invariant subgraphs to obtain stable predictions under distribution shifts, GSAT [30] does not consider temporal dependencies and sequential nature of dynamic graphs. Specifically, GSAT [30]'s marginal distribution of the invariant subgraph is characterized by Bernoulli distribution as follows: the probability of an edge being included in the invariant subgraph is drawn from the Bernoulli distribution with some parameter r. On the contrary, our OOD-Linker considers the sequential nature of dynamic graphs, i.e., the occurrence of links at a certain time is affected by the link occurrences at previous timestamp. From the temporal setting, we reason that the invariant links at a certain time are also affected by the invariant links at previous timestamps. Therefore, for the minimization of $\beta I(\{e\}_1^T; \{G\}_1^T)$ (Section 3.2.2), the marginal distribution (or prior), $q_{\theta_3}(.)$, of obtaining invariant links at time t, e_t, is a conditional probability, given the observed invariant links at previous timestamps, $\{e\}_i^{t-1}$. In this way, the sequential nature of dynamic graphs could be taken into account. Moreover, the probability distribution of including invariant links, is also conditioned on previous invariant links. In summary, compared to GSAT [30], designed to operate on static graphs, our IB principle and derivation of variational bounds face additional challenges posed by dynamic graphs: temporal dependencies and sequential nature of dynamic graphs. The entanglement of temporal dependencies is also shown in our rigorous proof for obtaining the variational bounds in Appendix 1 (Supplementary Materials). Moreover, in terms of downstream tasks, GSAT [30] is evaluated on Graph Classification, while our OOD-Linker is assessed on Temporal Link Prediction.

DGIB, a work for dynamic graphs, but different from GSAT [30] and our OOD-Linker, which is not targeting the invariant subgraphs. Specifically, DGIB [58] maximizes the mutual information (MI) between the learned representation at the current timestamp $T+1$ with the queried link label (i.e., 1 for existing link occurrence, and 0 otherwise) (the first term of Eq. 6 in DGIB [58] paper) and constrains the MI between the learned representation at the current timestamp $T+1$ with the (the second term of Eq. 6 in DGIB [58] paper) raw historical data occurred before T. However, our OOD-Linker maximizes the MI between invariant links at all timestamps in the queried link's computational subgraph and its label (first term in Eq. 5), while constrains the MI between these invariant links and the raw historical data (second term in Eq. 5). In summary, DGIB [58] seeks to utilize IB principle to obtain the learned representation at a single timestamp, while OOD-Linker aims to extract invariant links occurring at multiple timestamps of the query link's computational subgraph and entangles the temporal dependencies between invarian links occurring at different timestamps. Also, in terms of empirical evaluation, DGIB [58]'s robustness is designed for the adversarial attacks (e.g., structure perturbation and feature perturbation), while our OOD-Linker evaluates the model's robustness under distribution shifts (e.g., new unseen domain in the training process like the Data Mining papers' citation pattern and features for the training set consisting of Medical Informatics, Theory, and Visualization papers).

D Experimental Details

D.1 Dataset details

In this section, we report the dataset statistics and illustrate the distribution shifts in 2 real-world datasets used in our empirical evaluation: COLLAB [44] and ACT [14]. Specifically, we report the dataset statistics in Table 3. In Table 3, # Nodes, # Links, # Unique Timestamps, denote the number of nodes, number of edges, and number of unique timestamps of the dataset temporal graph. The dataset that yields the most challenging setting is COLLAB, which has the longest time span but has the coarsest temporal granularity, and more substantial

Table 3: Datasets Statistics

Dataset	# Nodes	# Links	# Unique Timestamps	Granularity
COLLAB	23,035	151,790	16	year
ACT	20,408	202,339	30	day

Table 4: Datasets Statistics for in-distribution and out-of-distribution datasets

Dataset	# Nodes	# Links	# Unique Timestamps	Granularity
COLLAB (in-distribution)	19,806	136,996	16	year
COLLAB (out-of-distribution)	3,235	14,792	16	year
ACT (in-distribution)	15,274	184,190	30	day
ACT (out-of-distribution)	2,338	18,149	22	day

difference in links' properties, compared to ACT.

D.2 Out-of-distribution data for distribution shift in edge attributes

Next, we elaborate more details on how out-of-distribution data is obtained. Following previous works, DIDA [62], EAGLE [59], and SILD [64], we obtain the out-of-distribution dataset based on the field information of edges as follows.

- COLLAB [44] is an academic collaboration dataset consisting of papers published during the time window 1990 - 2006. Nodes denote authors, and edges denote co-authorship. There are 5 possible attributes for a co-authorship, "Data Mining", "Database", "Medical Informatics", "Theory", and "Visualization". Edges associated with "Data Mining" are filtered out for representing the out-of-distribution dataset, while edges with other 4 field information are merged together, forming the in-distribution dataset. For the input node features of the graph, Word2Vec [31] is employed to construct 32-dimensional node features based on paper abstracts.

- ACT [14] is a dynamic graph demonstrating student actions on a MOOC platform in a month. Nodes represent students or targets of actions, and edges represent actions, respectively. K-Means [27] is leveraged to cluster the action features into 5 categories, and a certain category is randomly selected to act as the shifted attribute. Then, each student or target is assigned the action features, and these features are further expanded to 32-dimensional features with a linear function.

We report the dataset statistics for in-distribution and out-of-distribution datasets of COLLAB and ACT in Table 4.

D.3 Synthetic data for distribution shift in node features

Here, we elaborate more on how to obtain synthetic data that exhibits distribution shifts in node attributes. Following previous work on OOD Generalization for Dynamic graphs, DIDA [62], EAGLE [59], SILD [64], we interpolate node features of the COLLAB dataset as follows. Firstly, we sample $p(t)|\mathcal{E}^{t+1}|$ (where $\mathcal{E}^{t+1}$ is the number of links in the next timestamp) positive links and $(1-p(t))|\mathcal{E}^{t+1}|$ negative links, then these links are further factorized into shifted features $X^t \in \mathbb{R}^{|\mathcal{V}|\times d}$, which is the node features for graph snapshot at time t. The sampling probability, $p(t)$, is varied with $\bar{p}$: $p(t) = \bar{p} + \sigma \cos(t)$. In this way, node features with higher $p(t)$ will have stronger spurious correlations with future graph snapshots. $\bar{p} = 0.1$ is used for the testing set, while for the training dataset, $\bar{p}$ is varied from $\{0.4, 0.6, 0.8\}$, corresponding to the 3 datasets $\text{COLLAB}(\bar{p} = 0.4), \text{COLLAB}(\bar{p} = 0.6), \text{COLLAB}(\bar{p} = 0.8)$ in Table 2 of the main paper.

D.4 Hyperparamter details

For the purpose of reproducibility, we report the configuration and hyper-parameters in this section. Firstly, we present the hyper-parameters that are unchanged for all datasets:

Table 5: Hyper-parameters across all datasets

Dataset	Node features dimension	Edge features dimension	Time Encoding dimension
COLLAB	32	8	9
ACT	32	8	16
COLLAB ($\bar{p} = 0.4$)	64	1	9
COLLAB ($\bar{p} = 0.6$)	64	1	9
COLLAB ($\bar{p} = 0.8$)	64	1	9

- Temperature, $\tau = 1.0$
- Learning rate: 0.0005
- Batch size: 400

Next, we report the detailed hyper-parameters across all datasets in Table 5. The experiments are implemented by Python and executed on a Linux machine with a single NVIDIA Tesla V100 32GB GPU. The code will be released upon the paper's publication.

D.5 Additional real-world dataset performance comparison

Here, we report additional empirical results on Aminer dataset [45, 43] for our OOD-Linker, and other Dynamic Graph OOD Generalization methods, including DIDA [62], EAGLE [59], and SILD [64], in Table 6. Compared to other baselines for Dynamic Graph OOD Generalization, our OOD-Linkerachieves the best ROC score on Aminer.

Table 6: Temporal Link Prediction (ROC) in the Edge OOD Setting.

Dataset	Aminer
DIDA	94.06 ± 1.16
EAGLE	96.59 ± 0.07
SILD	94.45 ± 0.47
OOD-Linker (OURS)	**98.62 ± 0.06**

E Complexity Analysis

In this section, we provide the runtime complexity analysis for OOD-Linker. Given that we have N query links, we aim to obtain link predictions with OOD-Linker. For each query link, we would need to traverse all nodes in each T discrete snapshots of its computational graphs to obtain temporal graph representations, resulting in $\mathcal{O}(|\mathcal{V}|T)$, where $|\mathcal{V}|$ is the number of nodes. Then, we compute the invariant edge weight for each link in each discrete snapshot, leading to additional $\mathcal{O}(\sum_{t=1}^{T} |\mathcal{E}^t|) \leq \mathcal{O}(|\mathcal{E}|)$ cost, where $|\mathcal{E}^t|$ is the number of edges in a snapshot at time t, and $|\mathcal{E}|$ is the total number of edges in the temporal graphs. Thus, the total computational complexity for obtaining predictions for N query links is $\mathcal{O}(N(|V|T + |E|))$, which scales linearly with the number of nodes and edges in the graph.

F Pseudo-code

In this section, we provide the pseudo-code to illustrate OOD-Linker's training procedure.

Algorithm 1 Pseudo-code for OOD-Linker's training procedure

```
Require: $N$ query links
Ensure: Link Predictions
  for epoch in [1, . . . , number of epochs] do
      for each query link $(u, v, T+1)$ do
          Obtain computational graphs $\widetilde{G}_{u,v}$
          for each time $t$ in $[1, \ldots, T]$ do
              for each link $(a, b, t) \in \widetilde{G}^t_{u,v}$ do
                  Compute $p_{\phi_2}((a, b, t) \in$ invariant subgraph$)$ as in Eq. 7, 8, 9, and 10 of the main paper
                  Compute $q_{\phi_3}((a, b, t) \in$ invariant subgraph$)$ as in Eq. 11 and 12 of the main paper
              end for
          end for
          Temporal node representations for $u, v$: $\mathbf{h}^T_{u,N,\phi_1}, \mathbf{h}^T_{v,N,\phi_1} \leftarrow$ Eq. 13 in the main paper, using invariant edge weights $p_{\phi_2}((a, b, t)), \forall (a, b, t) \in \widetilde{G}^t_{u,v}, \forall t$
          Link prediction $\hat{y} \leftarrow$ Eq. 14 in the main paper
      end for
      $\mathcal{L} \leftarrow$ Eq. 15 in the main paper, applying BCE loss for link predictions $\hat{y}$, and invariant edge weights computed by $\phi_2, \phi_3$: $p_{\phi_2}((a, b, t)), q_{\phi_3}((a, b, t)), \forall (a, b, t) \in \widetilde{G}^t_{u,v}, \forall t$
      Backpropagate loss and update the model's weights
  end for
```