Learning Divergence Fields for Shift-Robust Graph Representations

Qitian Wu [1] **Fan Nie** [1] **Chenxiao Yang** [1] **Junchi Yan** [1]

Abstract

Real-world data generation often involves certain geometries (e.g., graphs) that induce instance-level interdependence. This characteristic makes the generalization of learning models more difficult due to the intricate interdependent patterns that impact data-generative distributions and can vary from training to testing. In this work, we propose a geometric diffusion model with learnable divergence fields for the challenging generalization problem with interdependent data. We generalize the diffusion equation with stochastic diffusivity at each time step, which aims to capture the multi-faceted information flows among interdependent data. Furthermore, we derive a new learning objective through causal inference, which can guide the model to learn generalizable patterns of interdependence that are insensitive across domains. Regarding practical implementation, we introduce three model instantiations that can be considered as the generalized versions of GCN, GAT, and Transformers, respectively, which possess advanced robustness against distribution shifts. We demonstrate their promising efficacy for out-of-distribution generalization on diverse real-world datasets. Source codes are available at https://github.com/fannie1208/GLIND.

1. Introduction

Learning from data involving certain geometries is a fundamental challenge in machine learning (Tenenbaum et al., 2000; Belkin & Niyogi, 2003; Belkin et al., 2006). One common scenario entails explicit graph structures, where the observed edges create interdependence among data points; another more challenging scenario involves implicit structures, where interdependence also exists but is not directly observable from data. In both scenarios, due to the interdependence of data points, the commonly used i.i.d. assumption for modeling becomes invalid. Moreover, given the dynamic environment where the model interacts in the open-world setting, the training and testing data are often generated from different distributions, necessitating generalization under distribution shifts (Mansour et al., 2009; Blanchard et al., 2011; Muandet et al., 2013; Gong et al., 2016; Arjovsky et al., 2019).

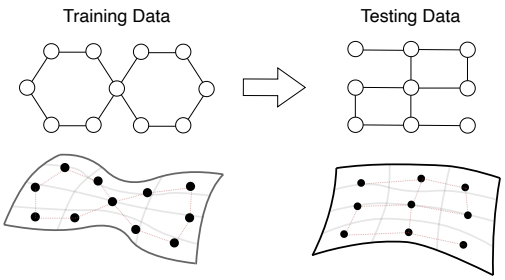


Figure 1: The challenge of generalization with interdependent data involves distribution shifts regarding the underlying manifolds that define the proximity among data samples.

However, unlike standard learning settings (e.g., for image and text data) where each data sample can be treated as independent (Koh et al., 2021; Ye et al., 2021), the interdependence among data points (e.g., nodes in an observed graph or other geometries with implicit structures) significantly increases the difficulty of generalization (Ma et al., 2021; Wu et al., 2022; Zhao et al., 2020; Stadler et al., 2021). Particularly, as illustrated in Fig. 1, since the training and testing data lie on different underlying manifolds that induce proximity among data points, the model solely optimized on training data with specific geometry may not generalize well to testing data generated from a different manifold. In contrast, an ideal model should be able to capture the *generalizable* patterns of data interdependence that can transfer across domains (or interchangeably, environment).

While generalization across potentially different environments is desirable in practice, the challenge lies in how to model the generalizable patterns of the interdependence among data points, which can be abstract and convoluted in nature. Given the data interdependence, the label of each instance depends on both the instance itself and other

[1]School of Artificial Intelligence & Department of Computer Science and Engineering & MoE Lab of AI, Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Junchi Yan <yanjunchi@sjtu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

instances, and such input-label dependency, which is important for prediction, would vary accordingly when distribution shifts exist (between training and testing data). Thereby, for generalization, the model needs to learn causal predictive relations from the inputs of interdependent data with certain geometries to their labels, which stays insensitive to distribution shifts.

In this paper, we commence with diffusion equations on manifolds (Weickert et al., 1998; Romeny, 2013) as the foundation for learning generalizable predictive relations with interdependent data. We propose a geometric diffusion model with learnable divergence fields as an effective means to model the complex interdependent patterns among data points. In particular, we generalize the *diffusivity* function, which measures the rate of information flows between two points on the manifold, as random samples from a variational distribution over a set of diffusivity hypotheses, inducing stochastic evolutionary directions at each step. In this way, the model can accommodate and capture the multi-faceted information flows at each diffusion step.

To facilitate the generalization, we harness deconfounded learning, a technique from causal inference, and derive a step-wise re-weighting regularization approach for the inferred latent diffusivity. This gives rise to a new optimization objective that guides the model to learn the causal relation induced by the diffusion dynamics trajectory, which captures the generalizable relations from initial states (x) to output states (y) that are insensitive to data distribution shifts. In terms of practical implementation, we present three instantiations based on our model formulation, which can be treated as generalized versions of GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), and DIFFormer (Wu et al., 2023c), respectively, for out-of-distribution generalization in situations where the data geometries are observed (e.g., as graphs) or unobserved.

We evaluate the practical efficacy of our models on diverse experimental datasets. The results show that the proposed models can effectively handle various distribution shifts. The contributions are summarized below.

○ We propose a geometric diffusion model with learnable divergence fields that accommodates multi-faceted information flows among data points. The diffusivity at each step is estimated via a variational distribution conditioned on the diffusion dynamics trajectories.

○ We design a causal regularization approach for learning the generalizable patterns of interdependence. Our scheme serves as the first attempt to unlock the potential of diffusion models for generalization with interdependent data.

○ We introduce three practical versions of model implementation and demonstrate their superiority through extensive experiments, particularly in handling various distribution shifts with observed and unobserved data geometries.

2. Background

In this section, we review some technical background as building blocks of our proposed model.

Diffusion on Manifolds. The diffusion process over some abstract domain Ω composed of N points describes the evolution of particular signal $z(u,t)$, a scalar-valued function on $\Omega \times [0,\infty)$, for each point $u \in \Omega$ at arbitrary time t. In specific, the evolution of $z(u,t)$ by a (heat) diffusion process can be described via a partial differential equation (PDE) with boundary conditions (Freidlin & Wentzell, 1993; Medvedev, 2014; Romeny, 2013):

$$\frac{\partial z(u,t)}{\partial t} = \nabla^* \left(D(u,t) \odot \nabla z(u,t) \right), \tag{1}$$

with initial conditions $z(u,0) = z_0(u), t \geq 0, u \in \Omega$. Here ∇, ∇^* and $D(u,t)$ reflect some spatial characteristics and are associated with the structure of Ω. Without loss of generality, we assume Ω as a Riemannian manifold and define $\mathcal{Z}(\Omega)$ and $\mathcal{Z}(T\Omega)$ as the scalar and (tangent) vector fields on Ω, respectively. Then the gradient operator $\nabla : \mathcal{Z}(\Omega) \to \mathcal{Z}(T\Omega)$ returns a vector field $\nabla z(u,t)$ that provides the direction of the steepest change of z at the point u. The divergence operator $\nabla^* : \mathcal{Z}(T\Omega) \to \mathcal{Z}(\Omega)$ is a scalar field that reflects the flow of $\nabla z(u,t)$ through an infinitesimal volume around u. The $D(u,t)$ in Eqn. 1 is the *diffusivity* that describes some thermal conductance properties of Ω, i.e., the measure of the rate at which heat can spread over the space (Rosenberg & Steven, 1997). For homogeneous system, the diffusivity $D(u,t)$ remains a constant for arbitrary u, while $D(u,t)$ is position-dependent for inhomogeneous system. In the latter case, $D(u,t)$ can be scalar-valued (isotropic diffusion) or matrix-valued (anisotropic diffusion) (Weickert et al., 1998).

Causal Deconfounded Learning. The principle of deconfounded learning is originally rooted in the study of Causality (Pearl et al., 2016; Rubin, 2019), which pursuits the causal (a.k.a. stable) mappings between input x (cause) and output y (effect) and eliminates the influence from a latent confounder c, i.e., the common cause of x and y. In machine learning tasks, the domain context in a dataset can be a common confounder between input features x and labels y. For example, in image classification, the background often associates with both x and y, e.g., a horse often appears on the grass. The domain context would mislead the model to learn non-stable correlation between inputs and labels, e.g., a shortcut between the 'green' background and the 'horse' (Beery et al., 2018; Geirhos et al., 2018). Such correlation however does not hold on other datasets collected with different contexts and may impair the model generalization. Deconfounded learning aims to remove the

confounding bias from domain context by intervention on x and maximize $p(y|do(x))$ where the do-operation cuts off the dependence from c and x. The probability with the do operator can be computed by randomized controlled trial using data recollection or statistically estimated by leveraging backdoor adjustment (Pearl et al., 2016; Christakopoulou et al., 2020; Roberts et al., 2020; Yang et al., 2022a).

3. Model Formulation

We consider N data samples (a.k.a. instances) $\{\mathbf{x}_u, y_u\}_{u=1}^N$ that are partially labeled and the labeled portion is $\{\mathbf{x}_u, y_u\}_{u=1}^M$ where $M < N$. There exist interconnecting structures among the N data points, reflected by $\mathcal{G} = \{a_{uv}\}_{N\times N}$ where $a_{uv} = 1$ indicates the connectivity between instance u and v and 0 for non-connectivity. In this situation, each instance u can be considered as the node in the graph $\mathcal{G}$. Furthermore, without loss of generality, the structures can also be unobserved, in which case we can assume $a_{uv} = 1$ for $\forall u, v$ suggesting that there can exist potential interactions between every instance pair. We introduce $\mathbf{Z} = [\mathbf{z}_u]_{u=1}^N$ where $\mathbf{z}_u \in \mathbb{R}^d$ denotes the representation (a.k.a. embedding) of instance u.

3.1. Geometric Diffusion Model

Diffusion Equation over Discrete Space. We first characterize a neural message passing model induced by generic diffusion process Eqn. 1 defined over a discrete space consisting of N points as locations on the manifold. Through treating the instance representations as signals of locations in the discrete space, we can define the gradient and divergence operators according to the discretization of the continuous notions. The gradient operator ∇ measures the difference between source and target locations, i.e., $(\nabla\mathbf{Z}(t))_{uv} = \mathbf{z}_u(t) - \mathbf{z}_v(t)$. The divergence operator ∇^* aggregates the information flows at a point, i.e., $(\nabla^*)_u = \sum_{v,a_{uv}=1} \mathbf{d}_v(\mathbf{Z}(t), u, t)\,(\nabla\mathbf{Z}(t))_{uv}$, where $\mathbf{d}(\mathbf{Z}(t), u, t)$ determines the diffusivity of the location u at time t. We can then obtain the diffusion equation that describes the evolution of instance representations over the observed discrete structures among N data points:

$$\frac{\partial \mathbf{z}_u(t)}{\partial t} = \sum_{v,a_{uv}=1} \mathbf{d}_v(\mathbf{Z}(t), u, t)\left(\mathbf{z}_v(t) - \mathbf{z}_u(t)\right), \quad (2)$$

with initial conditions $\mathbf{Z}(0) = [\mathbf{x}_u]_{u=1}^N$ and $t \geq 0$. The above equation can be essentially considered as a continuous version of neural message passing, commonly used by graph neural networks (Chamberlain et al., 2021a) and Transformers (Wu et al., 2023c), while the latter models uses discrete layers as the approximation of time. On top of this connection, recent works harness diffusion equations on graphs as a principled perspective for justifying architectural choices (Chamberlain et al., 2021b; Thorpe et al., 2022), analyzing GNNs' behaviors (Wu et al., 2023b) and guiding model designs for difficult problems, e.g., geometric knowledge distillation (Yang et al., 2022b).

Branching-Structured Divergence Fields. The above diffusion system assumes that at each time step the information flows between connected locations are determined by a deterministic diffusivity function $\mathbf{d}(\mathbf{Z}(t), u, t)$. The diffusivity essentially measures the rate of information flows between any pair of locations on the manifold. Based on the analogy between diffusion on manifolds and message passing among interconnected data, the diffusivity can be treated as quantification of pairwise influence among data points. The latter, however, could be driven by multiple criteria along with uncertainty, instead of remaining to be deterministic. In light of this observation, we extend Eqn. 2 to accommodate the heterogeneity and stochasticity of the diffusivity, which is assumed to be sampled from a generative distribution $p(\mathbf{d}^{(t)}|\mathbf{Z}(t), u, t)$, over a set of hypothesis, dependent on the current states $\mathbf{Z}(t)$, position u and time t:

$$\frac{\partial \mathbf{z}_u(t)}{\partial t} = \sum_{v,a_{uv}=1} d_{uv}^{(t)} \cdot \left(\mathbf{z}_v(t) - \mathbf{z}_u(t)\right), \quad (3)$$

where $[d_{uv}^{(t)}]_{v=1}^N = \mathbf{d}_u^{(t)} \sim p(\mathbf{d}^{(t)}|\mathbf{Z}(t), u, t)$. Through the diffusion process defined by Eqn. 3, the change of instance representation at infinitesimal time is equal to the *divergence* at each point u induced by the current diffusivity $\mathbf{d}_u^{(t)}$. The evolutionary direction of instance representation at time t is stochastic according to $p(\mathbf{d}^{(t)}|\mathbf{Z}(t), u, t)$ that yields multiple branches of subsequent diffusion dynamics from the current time (as illustrated in Fig. 2(d)). This formulation further paves the way for generalization by enabling the model to learn the stable diffusion dynamics as we will discuss in later subsections.

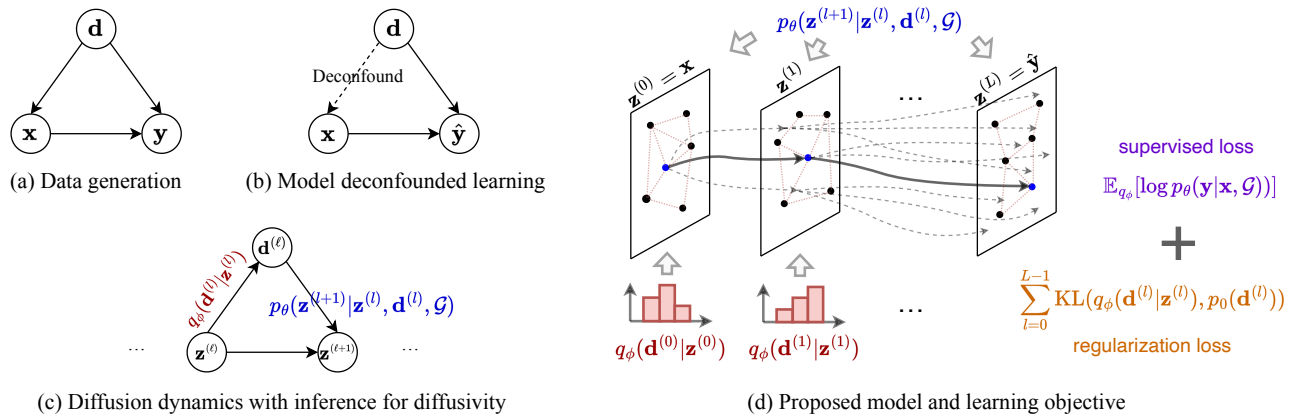
Diffusion Trajectories with Stochastic Diffusivity. The differential equation of Eqn. 3 can be solved through numerical methods by using discrete time steps $\{l\}$ to approximate the continuous time t. In particular, with step size α, Eqn. 3 induces a trajectory of instance embeddings:

$$\mathbf{z}_u^{(l+1)} = \mathbf{z}_u^{(l)} + \alpha \sum_{v,a_{uv}=1} d_{uv}^{(l)} \cdot \left(\mathbf{z}_v^{(l)} - \mathbf{z}_u^{(l)}\right), \quad (4)$$

where $[d_{uv}^{(t)}]_{v=1}^N = \mathbf{d}_u^{(l)} \sim p(\mathbf{d}^{(l)}|\mathbf{z}_u^{(l)})$. The diffusion model Eqn. 4 with L layers yields a trajectory of embeddings for each instance u: $\mathbf{x}_u = \mathbf{z}_u^{(0)} \to \mathbf{z}_u^{(1)} \to \cdots \to \mathbf{z}_u^{(L)} = \hat{\mathbf{y}}_u$, where $\hat{\mathbf{y}}_u$ denotes the predicted label for u.

3.2. Optimization with Causal Regularization

For generalization, we next discuss on how to guide the geometric diffusion model to learn causal relations from inputs to outputs that are insensitive to distribution shifts. Consider



(a) Data generation (b) Model deconfounded learning

(c) Diffusion dynamics with inference for diffusivity (d) Proposed model and learning objective

Figure 2: (a)$\sim$ (c) Dependence among random variables of interest (here we omit $\mathcal{G}$ for brevity since everything can be treated as conditioned on $\mathcal{G}$). (a) Causal dependence for data generation where the diffusivity $\mathbf{d}$ is the common cause of $\mathbf{x}$ and $\mathbf{y}$. (b) Deconfounded learning $p_\theta(\mathbf{y}|do(\mathbf{x}), \mathcal{G})$ which aims to cut off the dependence path from $\mathbf{d}$ to $\mathbf{x}$ in order to learn causal (a.k.a. stable) relation between $\mathbf{x}$ and $\mathbf{y}$ for generalization. (c) Our diffusion model whose feed-forward dynamics $\mathbf{x}_u = \mathbf{z}_u^{(0)} \to \mathbf{z}_u^{(1)} \to \cdots \to \mathbf{z}_u^{(L)} = \hat{\mathbf{y}}_u$ is given by a predictive distribution $p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})$ and a variational distribution $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$. (d) The geometric diffusion model that is optimized by a new learning objective (comprised of a supervised term and a regularization term) which can achieve the goal of deconfounded learning $p_\theta(\mathbf{y}|do(\mathbf{x}), \mathcal{G})$.

the instance embedding $\mathbf{z}_u^{(l)}$ of an arbitrary given layer as a random variable, we define by $p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})$ the (delta) predictive distribution induced by the l-th layer's updating of Eqn. 4, where the condition on $\mathcal{G}$ stems from the message passing among instances. Then the generative distribution at the l-th layer can be expressed as

$$p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathcal{G}) = \mathbb{E}_{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}[p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})],$$

where the expectation marginalizes over all the possible results of $\mathbf{d}^{(l)}$. The log-likelihood $\log p_\theta(\mathbf{y}|\mathbf{x}, \mathcal{G})$ for each data sample can be computed by $\log \prod_{l=0}^{L-1} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathcal{G}) =$

$$\sum_{l=0}^{L-1} \log \mathbb{E}_{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}[p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})]. \quad (5)$$

Inference for Diffusivity. Due to the intractable integration over $\mathbf{d}^{(l)}$ in the above objective, we turn to the evidence lower bound of Eqn. 5 by introducing a variational distribution $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$ (where ϕ denotes its parameterization):

$$\sum_{l=0}^{L-1} \mathbb{E}_{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \left[\log p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G}) \frac{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \right]. \quad (6)$$

With some re-arranging, the evidence lower bound of the log-likelihood $\log p_\theta(\mathbf{y}|\mathbf{x}, \mathcal{G})$ induces the following objective (see derivation in Appendix A.1)

$$\mathbb{E}_{q_\phi(\mathbf{d}^{(0)}|\mathbf{z}^{(0)}), \cdots, q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} \Big[\log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(L-1)}, \mathcal{G}) \Big] - \sum_{l=0}^{L-1} \mathrm{KL}(q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}), p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})), \quad (7)$$

where the first term corresponds to the supervised loss by computing the difference between the ground-truth label $\mathbf{y}_u$ and the prediction $\hat{\mathbf{y}}_u$, and the second term has a regularization effect on the inferred latent diffusivity. Eqn. 7 is a tractable objective that approximates the log-likelihood.

However, the generative distribution $p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$ is a model-based prior for diffusivity over a hypothesis space that expresses how plausible the model thought the ideal diffusivity were in light of the previous diffusion dynamics. Though it incorporates the information of $\mathbf{z}^{(l)}$ for estimating the adaptive diffusivity at the current step, the model learning would make the estimation of diffusivity biased towards training data. In this way, the predictive relations from $\mathbf{z}^{(l)}$ (resp. $\mathbf{x}$) to $\mathbf{z}^{(l+1)}$ (resp. $\mathbf{y}$) learned by the model would be spuriously associated by the latent confounder, the diffusivity $\mathbf{d}^{(l)}$ (resp. $\{\mathbf{d}^{(l)}\}_{l=0}^{L-1}$), as illustrated by Fig. 2(a). The confounding effect of the diffusivity, which corresponds to the context information that differs across domains, would hamper the generalization of the model when transferring from the training distribution to a testing one where the data lies on another manifold with disparate proximity context.

Step-wise Re-weighting Regularization To facilitate generalization, the ideal solution is to achieve deconfounded learning, i.e., $p_\theta(\mathbf{y}|do(\mathbf{x}), \mathcal{G})$ that cuts off the dependence path from $\mathbf{d}$ to $\mathbf{x}$ (as illustrated in Fig. 2(b)), in which situation the model can learn the causal (a.k.a. stable) relation between $\mathbf{x}$ and $\mathbf{y}$ that is generalizable across domains. While the technical difficulty lies in how to compute $p_\theta(\mathbf{y}|do(\mathbf{x}), \mathcal{G})$ in a feasible manner, the next theorem presents a tractable surrogate that approximates the target.

Theorem 3.1. *For any given diffusion model* $p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})$*, we have a lower bound of the deconfounded learning objective:* $\log p_\theta(\mathbf{y}|do(\mathbf{x}), \mathcal{G}) \geq$

$$\sum_{l=0}^{L-1} \mathbb{E}_{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \left[\log p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G}) \frac{p_0(\mathbf{d}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}\right], \tag{8}$$

where $p_0(\mathbf{d}^{(l)})$ *is a model-free prior distribution. In particular, the equality holds for Eqn. 8 iff* $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}) = p_\theta(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}, \mathbf{z}^{(l+1)}, \mathcal{G}) \cdot \frac{p_0(\mathbf{d}^{(l)})}{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}$.

The proof is deferred to Appendix A.2. We can optimize Eqn. 8 as a surrogate objective by joint learning of q_ϕ and p_θ that has the effect of optimizing the deconfounded learning objective. As another perspective for understanding the effect of Eqn. 8, it can be seen as multiplying a re-weighting term $\frac{p_0(\mathbf{d}^{(l)})}{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}$ with Eqn. 6, which can down-weight the frequent diffusivity components and up-weight the infrequent ones to eliminate the observation bias caused by limited training data. This can facilitate learning the stable relation from $\mathbf{z}^{(l)}$ (resp. $\mathbf{x}$) to $\mathbf{z}^{(l+1)}$ (resp. $\mathbf{y}$) that holds across domains, instead of the non-stable correlation pertaining to specific context in training data.

Based on Eqn. 8, we can obtain the final learning objective (see derivation in Appendix A.3 which is similar to Eqn. 7):

$$\mathbb{E}_{q_\phi(\mathbf{d}^{(0)}|\mathbf{z}^{(0)}),\cdots,q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} \Big[\log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(L-1)}, \mathcal{G})\Big] - \sum_{l=0}^{L-1} \mathrm{KL}(q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}), p_0(\mathbf{d}^{(l)})). \tag{9}$$

Fig. 2(d) provides an illustration for the model. The computation of the regularization term depends on specific form of $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$, which we will discuss in the next section.

4. Model Instantiations

In this section, we delve into the model instantiations on the basis of the formulation of our diffusion model in Sec. 3. We call our model **GLIND**, short for diffusion-induced **G**eneralizable **L**earning with **IN**terdependent **D**ata. For simplicity, we assume the inference distribution for diffusivity $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$ as a multinomial distribution: at each time step l for arbitrary instance u, the diffusivity $\mathbf{d}_u^{(l)}$ is sampled from a limited hypothesis set $\{\mathbf{d}_u^{(l,k)}\}_{k=1}^K$ with probabilities $\{\pi_u^{(l,k)}\}_{k=1}^K$ for each component, where $\sum_{k=1}^K \pi_u^{(l,k)} = 1$. We can further assume $\alpha = 1$, and then the diffusion trajectory of Eqn. 4 induces a feed-forward propagation layer:

$$\mathbf{z}_u^{(l+1)} = \mathbf{z}_u^{(l)} + \sum_{k=1}^K h_{u,k}^{(l)} \sum_{v, a_{uv}=1} d_{uv}^{(l,k)} (\mathbf{z}_v^{(l)} - \mathbf{z}_u^{(l)}),$$

where $\mathbf{h}_u^{(l)} \sim \mathcal{M}(\boldsymbol{\pi}_u^{(l)})$ is a one-hot vector where the entry '1' indicates the sampled result from the multinomial distribution $\mathcal{M}(\boldsymbol{\pi}_u^{(l)})$. Here $\boldsymbol{\pi}_u^{(l)}$ and $\mathbf{d}_u^{(l,k)}$ determine the divergence (the updating signals) at the current time step l and control the evolutionary direction of the diffusion process, and both of them can be parameterized.

Parameterization. We adopt a feed-forward layer with Softmax to model the probability of K diffusivity components and leverage the Gumbel trick (Jang et al., 2017) to approximate the sampling process:

$$h_{u,k}^{(l)} = \frac{\exp\left(\left(\pi_u^{(l,k)} + g_k\right)/\tau\right)}{\sum_{k'} \exp((\pi_u^{(l,k')} + g_{k'})/\tau)}, \quad g_k \sim \mathrm{Gumbel}(0,1),$$

where $[\pi_u^{(l,k)}]_{k=1}^K = \boldsymbol{\pi}_u^{(l)} = \mathrm{Softmax}(\mathbf{W}_L^{(l)} \mathbf{z}_u^{(l)})$ and $\mathbf{W}_L^{(l)} \in \mathbb{R}^{K \times d}$ is a trainable weight matrix. The message passing is driven by the diffusivity hypothesis set $\{\mathbf{d}_u^{(l,k)}\}_{k=1}^K$. We next introduce three model instantiations that can be considered as the generalized implementations of GCN, GAT and Transformers, respectively.

GLIND-GCN. We first consider a model version that uses the aggregated embeddings of connected instances to update the representation of the target instance. Inspired by the anisotropic diffusion which is position- and direction-dependent, we assume each branch k converts the embeddings for the target instance (resp. other connected instances) through $\mathbf{W}_S^{(l,k)} \in \mathbb{R}^{d \times d}$ (resp. $\mathbf{W}_D^{(l,k)} \in \mathbb{R}^{d \times d}$) and the gradient of signals is scaled by the number of connected instances $\tilde{d}_u = \sum_v a_{uv}$:

$$\mathbf{z}_u^{(l+1)} = \mathbf{z}_u^{(l)} + \sum_{k=1}^K h_{u,k}^{(l)} \left(\sum_{v, a_{uv}=1} \frac{1}{\tilde{d}_u} \mathbf{W}_D^{(l,k)} \mathbf{z}_v^{(l)} + \mathbf{W}_S^{(l,k)} \mathbf{z}_u^{(l)} \right),$$

where $\{\mathbf{W}_S^{(l,k)}\}_{k=1}^K$ and $\{\mathbf{W}_D^{(l,k)}\}_{k=1}^K$ are trainable weights at the l-th layer. This model can be seen as a generalized implementation of the GCN architecture (Kipf & Welling, 2017), where the approximate one-hot vector $\mathbf{h}_u^{(l)}$ dynamically selects the convolution filter in each layer.

GLIND-GAT. We can further harness an attention network for each branch to model the intensity of the pairwise influence between connected nodes, by extending the spirit of GAT (Velickovic et al., 2018):

$$\mathbf{z}_u^{(l+1)} = \mathbf{z}_u^{(l)} + \sum_{k=1}^K h_{u,k}^{(l)} \left(\sum_{v, a_{uv}=1} w_{uv}^{(l,k)} \mathbf{W}_D^{(l,k)} \mathbf{z}_v^{(l)} + \mathbf{W}_S^{(l,k)} \mathbf{z}_u^{(l)} \right),$$

$$w_{uv}^{(l,k)} = \frac{\delta((\mathbf{c}^{(l,k)})^\top [\mathbf{W}_A^{(l,k)} \mathbf{z}_u^{(l)} \| \mathbf{W}_A^{(l,k)} \mathbf{z}_v^{(l)}])}{\sum_{w, a_{uw}=1} \delta(\mathbf{c}^{(l,k)})^\top [\mathbf{W}_A^{(l,k)} \mathbf{z}_u^{(l)} \| \mathbf{W}_A^{(l,k)} \mathbf{z}_w^{(l)}])},$$

where δ is instantiated as LeakyReLU and $\mathbf{W}_A^{(l,k)} \in \mathbb{R}^{d \times d}$ and $\mathbf{c}^{(l,k)} \in \mathbb{R}^{2d}$ are both trainable parameters. Here, $\mathbf{h}_u^{(l)}$ adaptively selects the path for attentive propagation.

GLIND-TRANS. The preceding two models resort to message passing over observed structures. There also exist

scenarios where the data geometries are partially observed or unobserved. To accommodate the potential interactions among arbitrary instance pairs, we extend the attention-based propagation to a latent topology as is done by Transformer (Vaswani et al., 2017):

$$\mathbf{z}_u^{(l+1)} = \mathbf{z}_u^{(l)} + \sum_{k=1}^{K} h_{u,k}^{(l)} \left(\mathbf{W}_D^{(l,k)} \mathbf{b}_u^{(l,k)} + \mathbf{W}_S^{(l,k)} \mathbf{z}_u^{(l)} \right),$$

$$\mathbf{b}_u^{(l,k)} = \sum_v \frac{\eta(\mathbf{W}_K^{(l,k)} \mathbf{z}_v^{(l)}, \mathbf{W}_Q^{(l,k)} \mathbf{k}_u^{(l)})}{\sum_{w=1}^N \eta(\mathbf{W}_K^{(l,k)} \mathbf{z}_w^{(l)}, \mathbf{W}_Q^{(l,k)} \mathbf{k}_u^{(l)})} \cdot \mathbf{z}_v^{(l)},$$

where $\mathbf{W}_K^{(l,k)}$ and $\mathbf{W}_Q^{(l,k)}$ are trainable weights at the l-th layer, and η denotes a certain similarity function. However, due to the all-pair global attention, the above computation requires quadratic complexity w.r.t. N, which hinders the scalability to large numbers of instances. We thus turn to a linearly scalable and numerically stable version of all-pair attention proposed by DIFFormer (Wu et al., 2023c), which specifically instantiates η as $\eta(\mathbf{a}, \mathbf{b}) = 1 + (\frac{\mathbf{a}}{\|\mathbf{a}\|_2})^\top \frac{\mathbf{b}}{\|\mathbf{b}\|_2}$. Then assuming $\mathbf{k}_u^{(l)} = \frac{\mathbf{W}_K^{(l,k)} \mathbf{z}_u^{(l)}}{\|\mathbf{W}_K^{(l,k)} \mathbf{z}_u^{(l)}\|_2}$ and $\mathbf{q}_u^{(l)} = \frac{\mathbf{W}_Q^{(l,k)} \mathbf{z}_u^{(l)}}{\|\mathbf{W}_Q^{(l,k)} \mathbf{z}_u^{(l)}\|_2}$, we can efficiently compute $\mathbf{b}_{u,v}^{(l,k)}$ by

$$\mathbf{b}_u^{(l,k)} = \frac{\sum_{v=1}^N \mathbf{z}_v^{(l)} + \left(\sum_{v=1}^N (\mathbf{k}_v^{(l)})(\mathbf{z}_v^{(l)})^\top \right) (\mathbf{q}_u^{(l)})}{N + (\mathbf{q}_u^{(l)})^\top (\sum_{v=1}^N \mathbf{k}_v^{(l)})}.$$

The above computation for updating N instances' embeddings in each layer can be achieved within $O(N)$ complexity since the three summation terms are shared by all instances and only requires once computation in practice. Moreover, if the observed structures are available, we can incorporate the aggregated embeddings of connected instances to the updating signals: $\mathbf{b}_u^{(l,k)} \leftarrow \frac{1}{2}(\mathbf{b}_u^{(l,k)} + \sum_{v, a_{uv}=1} \frac{1}{\tilde{d}_u} \mathbf{z}_v^{(l)})$.

Data-Driven Prior via Mixture of Posteriors. A proper setting of the prior distribution $p_0(\mathbf{d}^{(l)})$ is important but non-trivial. One simple solution is to assume the prior as some pre-defined trivial forms (Dinh et al., 2015), which however may potentially lead to over-regularization (Burda et al., 2015). Alternatively, one can estimate the prior by the average of the variational posterior (Hoffman & Johnson, 2016), i.e., $p_0(\mathbf{d}^{(l)}) \approx 1/N \sum_{u=1}^N q(\mathbf{d}^{(l)} | \mathbf{z}^{(l)} = \mathbf{z}_u^{(l)})$. Nevertheless, such an approach can be computationally expensive (Tomczak & Welling, 2018) and would lead to biased estimation given limited training data. In our case, inspired by (Tomczak & Welling, 2018) using mixture of Gaussian as a learnable prior, we propose to use a mixture of pseudo variational posteriors as a flexible prior estimation:

$$p_0(\mathbf{d}^{(l)}) = \frac{1}{T} \sum_{t=1}^{T} q(\mathbf{d}^{(l)} | \mathbf{z}^{(l)} = \tilde{\mathbf{z}}_t^{(l)}). \tag{10}$$

where $\tilde{\mathbf{z}}_t^{(l)}$ is the embedding of instance t in the generated pseudo dataset $\{\tilde{\mathbf{x}}_t, \tilde{y}_t\}_{t=1}^T$. The latter is constructed by randomly sampling T instances from the N instances in the observed dataset, and the structures of the pseudo dataset are randomly generated with a pre-defined probability for each potential edge. We set $T \ll N$ to reduce the computational cost. In this way, the prior estimation by Eqn. (10) plays as a general reflection of how the model recognizes each diffusivity hypothesis given uninformative inputs and is learned with the model in a fully data-driven manner.

5. Experiments

The goal of our experiments is to evaluate the generalization ability of the model under distribution shifts with interdependent data. We consider various real-world datasets that involve observed and unobserved data geometries, and the distribution shifts between training and testing sets are led by distinct contexts. Following common practice, we use the prediction accuracy on the out-of-distribution testing data for measuring the generalization performance.

5.1. Experiment Setup

Datasets. We adopt five datasets `Twitch`, `Arxiv`, `DPPIN`, `STL` and `CIFAR` for evaluation. Detailed information about these datasets are deferred to Appendix B. Since the properties of these datasets vary case by case, we consider different ways to split the observed data into multiple domains or construct out-of-distribution data with new unseen domains (where the data from different domains are considered as samples from different distributions). We next describe the evaluation protocol in terms of how we split the training and testing sets with distribution shifts in each case.

○ `Twitch` is a multi-graph dataset (Rozemberczki & Sarkar, 2021) where nodes of graphs are instances whose interconnectivity is reflected by observed structures. Each subgraph is comprised of a social network of users from a particular region, so one can use the nodes in different subgraphs as samples from different domains, since these subgraphs have obviously different topological features (Wu et al., 2022). In particular, we use the nodes from three subgraphs as training data (where we hold out 25% for validation), and the nodes from the other three subgraphs as testing data. The task is a binary classification for users' genders, and the performance is measured by ROC-AUC.

○ `Arxiv` is a temporal network (Hu et al., 2020b) which evolves with time. Each node (i.e., a paper) as an instance has a time label indicating its publication year. Since the citation relationships among papers can significantly vary with different time-sensitive contexts, the distribution shifts naturally exist for the network data collected within different time windows. We thus use the papers published

Table 1: Testing (mean±standard deviation) Accuracy (%) for `Arxiv` and ROC-AUC (%) for `Twitch` on different subsets of out-of-distribution data (determined by publication times and subgraphs, respectively). The missing results of EERM on `Arxiv` stem from the out-of-memory issue. We mark the **first**/**second**/**third** methods with the top performance.

Method	Arxiv			Twitch		
	2014-2016	**2016-2018**	**2018-2020**	**ES**	**FR**	**EN**
ERM-GCN	56.33 ± 0.17	53.53 ± 0.44	45.83 ± 0.47	66.07 ± 0.14	52.62 ± 0.01	63.15 ± 0.08
IRM-GCN	55.92 ± 0.24	53.25 ± 0.49	45.66 ± 0.83	66.95 ± 0.27	52.53 ± 0.02	62.91 ± 0.08
GroupDRO-GCN	56.52 ± 0.27	53.40 ± 0.29	45.76 ± 0.59	66.82 ± 0.26	52.69 ± 0.02	62.95 ± 0.11
DANN-GCN	56.35 ± 0.11	53.81 ± 0.33	45.89 ± 0.37	66.15 ± 0.13	52.66 ± 0.02	63.20 ± 0.06
Mixup-GCN	56.67 ± 0.46	54.02 ± 0.51	46.09 ± 0.58	65.76 ± 0.30	52.78 ± 0.04	63.15 ± 0.08
EERM-GCN	-	-	-	67.50 ± 0.74	51.88 ± 0.07	62.56 ± 0.02
GLIND-GCN	59.42 ± 0.33	56.84 ± 0.54	57.06 ± 1.21	67.72 ± 0.10	53.16 ± 0.08	64.18 ± 0.03
ERM-GAT	57.15 ± 0.25	55.07 ± 0.58	46.22 ± 0.82	65.67 ± 0.02	52.00 ± 0.10	61.85 ± 0.05
IRM-GAT	56.55 ± 0.18	54.53 ± 0.32	46.01 ± 0.33	67.27 ± 0.19	52.85 ± 0.15	62.40 ± 0.24
GroupDRO-GAT	56.69 ± 0.27	54.51 ± 0.49	46.00 ± 0.59	67.41 ± 0.04	52.99 ± 0.08	62.29 ± 0.03
DANN-GAT	57.23 ± 0.18	55.13 ± 0.46	46.61 ± 0.57	66.59 ± 0.38	52.88 ± 0.12	62.47 ± 0.32
Mixup-GAT	57.17 ± 0.33	55.33 ± 0.37	47.17 ± 0.84	65.58 ± 0.13	52.04 ± 0.04	61.75 ± 0.13
EERM-GAT	-	-	-	66.80 ± 0.46	52.39 ± 0.20	62.07 ± 0.68
GLIND-GAT	60.36 ± 0.36	58.98 ± 0.43	59.71 ± 0.53	67.82 ± 0.10	54.50 ± 0.12	64.32 ± 0.12

Table 2: Testing RMSE for `DPPIN` on different domains (determined by protein identification methods).

Method	Hazbun	Krogan (LCMS)	Krogan (MALDI)	Lambert	Tarassov	Uetz	Yu
ERM-Trans	1.82 ± 0.17	1.63 ± 0.04	1.57 ± 0.03	1.49 ± 0.07	1.62 ± 0.03	1.52 ± 0.04	1.51 ± 0.04
IRM-Trans	1.66 ± 0.14	1.86 ± 0.04	1.84 ± 0.04	1.52 ± 0.07	1.76 ± 0.03	1.66 ± 0.05	1.66 ± 0.04
DANN-Trans	1.69 ± 0.11	1.66 ± 0.02	1.62 ± 0.03	1.39 ± 0.05	1.63 ± 0.01	1.49 ± 0.01	1.50 ± 0.01
GroupDRO-Trans	1.65 ± 0.13	1.68 ± 0.02	1.65 ± 0.02	1.48 ± 0.03	1.72 ± 0.01	1.53 ± 0.04	1.53 ± 0.01
Mixup-Trans	1.46 ± 0.13	1.79 ± 0.05	1.76 ± 0.04	1.50 ± 0.06	1.70 ± 0.05	1.56 ± 0.06	1.59 ± 0.06
EERM-Trans	1.68 ± 0.47	1.91 ± 0.23	1.92 ± 0.09	1.47 ± 0.05	1.79 ± 0.11	1.67 ± 0.07	1.65 ± 0.08
GLIND-TRANS	1.02 ± 0.07	1.38 ± 0.07	1.33 ± 0.05	1.08 ± 0.04	1.40 ± 0.04	1.20 ± 0.04	1.20 ± 0.04

before 2014 as training data (where we hold out 25% for validation), and the papers published after 2014 as testing data. The task is to predict the subareas of papers, and the performance is measured by Accuracy.

∘ `DPPIN` consists of multiple datasets of biological protein interactions (Fu & He, 2022). Each dataset is comprised of protein instances corresponding to a particular protein identification method (that can be treated as the domain). Each protein has a time-evolving scalar representing the gene expression value and the proteins within each domain have certain interactions exposed by the co-expressed activities. The interaction networks at one time, reflecting the partially observed data geometries, can be noisy and incomplete given the dynamical evolution of the co-expression levels. We use the proteins of four datasets for training, one dataset for validation and the other seven datasets for testing. We consider the regression task for protein's gene expression values, and the performance is measured by RMSE.

∘ `STL-10` is an image dataset where each instance is an image. Since there are no observed structures interconnecting the instances, we create the inter-instance relations via k-nearest-neighbor (kNN) with the Euclidean distance between two instances' input features. We consider using different k's for constructing the data from different domains. To be specific, the value of k is set as $\{2, 3, 4, 8, 9, 10\}$ which results in six domains where each domain corresponds to a particular kNN graph interconnecting the fix set of instances. We split all the image instances into training and testing sets with the ratio 1:1. Then we use the training instances with the structures of the first three domains as training data (where we hold out 25% for validation), and the testing instances with the structures of the other three domains as testing data. In this way, the training and testing data can be seen as samples with disparate underlying geometries. The task is to predict the image labels, and the performance is measured by Accuracy.

∘ `CIFAR-10` is another image dataset. Similarly, we use kNN to construct the inter-instance relations. Differently, we fix $k = 5$ and consider cosine similarity as the distance function. To introduce distribution shifts, we add angle bias to the cosine similarity to construct data from different domains. To be specific, we denote by s_i the distance function used for kNN in the i-th domain, and particularly we consider $s_i(\mathbf{x}_1, \mathbf{x}_2) = 1 - \cos(\angle(\mathbf{x}_1, \mathbf{x}_2) + \theta_i)$ where $\angle$ denotes the angle of input feature vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. We set $\theta_i \in \{0^o, 30^o, 90^o, 150^o, 160^o, 170^o\}$, resulting in six

domains. We adopt the same way for data splitting as STL, resulting in training data from three domains and testing data from the other three domains. The angle bias causes the interconnecting structures of training and testing data to have different proximity patterns.

Competitors. We compare with a set of learning algorithms including empirical risk minimization (ERM), which is the most commonly used supervised learning objective, and state-of-the-art approaches for improving generalization such as the adversarial learning approach DANN (Liu et al., 2020), the distributionally robust optimization Group-DRO (Sagawa et al., 2019), the data augmentation approach Mixup (Zhang et al., 2017), and invariant risk minimization approaches IRM (Arjovsky et al., 2019) and EERM (Wu et al., 2022). These models are agnostic to encoder architectures. For fair comparison, we use GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018) and DIFFormer (Wu et al., 2023c) as their encoder backbones to compare with our models GLIND-GCN, GLIND-GAT and GLIND-TRANS, respectively. Due to space limit, we defer implementation details and hyper-parameter searching space to Appendix C.

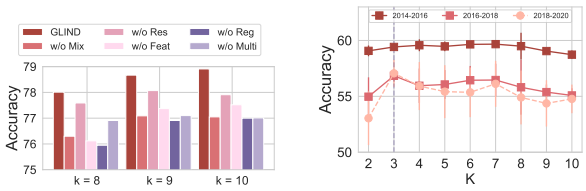
5.2. Comparative Results

Generalization with Observed Geometries. We first test the models on data with observed structures where we compare GLIND-GCN and GLIND-GAT with the competitors using GCN and GAT as the encoder backbones, respectively. Table 1 reports the performance on testing data of Arxiv and Twitch. In particular, for Arxiv, we further divide the testing data into three subsets according to the publication years of the papers to closely check the generalization performance on testing data with different levels of distribution shifts. As we can see, the accuracy produced by all methods exhibits an overall decrease as the time gap (between training and testing data) enlarges, while GLIND-GCN and GLIND-GAT significantly alleviate the performance drop. Notably, for testing data within 2018-2020, GLIND-GCN and GLIND-GAT achieve 23.8% and 26.6% improvement of the accuracy over the runner-ups, respectively. Furthermore, for Twitch, we separately report the ROC-AUC on three testing subgraphs to compare the generalization performance on specific cases. The results show that GLIND-GCN and GLIND-GAT achieve overall superior ROC-AUCs over these strong competitors, respectively.

Generalization with Partially Observed Geometries. We next consider the case where the data geometries are partially observed. This situation can be commonly encountered in practice, as the observed relational structures can be noisy and incomplete. To accommodate the potential interactions, we use our model GLIND-TRANS and compare it with the competitors using DIFFormer as the encoder. Table 2 reports the performance on testing data of DPPIN. We found that our model GLIND-TRANS consistently achieves the first-ranking results (the lowest RMSE) in seven testing domains, with an averaged reduction of 21.6% on RMSE over the runner-up. This demonstrates the practical efficacy of the proposed model for handling generalization with partially observed data geometries.

Generalization with Unobserved Geometries. The last scenario we study involves data geometries that are unobserved. We use the image datasets for demonstration where the inter-instance structures are synthetically constructed through the k-nearest-neighbor. We adopt the three model versions for comparison, where GLIND-GCN and GLIND-GAT purely rely on the input structures and GLIND-TRANS estimates the potential interactions among arbitrary instance pairs. Table 3 reports the performance on testing data of CIFAR and STL. In contrast with the competitors, our model yields the highest averaged accuracy on two datasets, demonstrating the superiority of the proposed model. Comparing the models using different backbones, we found that the Transformer-style architecture contributes to more stable performance (with lower variance across different testing sets). This is probably because the Transformer-style architecture can capture the latent interactions that are more insensitive to distribution shifts of structures than the GNNs that purely rely on observed structures.

5.3. Further Discussions



(a) Accuracy on testing sets (b) Accuracy on testing sets

Figure 3: (a) Ablation studies for GLIND on STL. (b) Performance of GLIND on Arxiv with different K's.

Ablation Studies. To further verify the effectiveness of GLIND, we next dissect the efficacy of the specific proposed components in our model. We conduct ablation studies w.r.t. model architectures and learning objectives, respectively, and report the comparison results on STL in Fig. 5(a) (more results are deferred to Appendix D). For the model architecture, we compare with three variants: 1) *w/o Res* removes the residual link ($\mathbf{z}_u^{(l)}$ added to each layer); 2) *w/o Feat* removes the self-loop feature aggregation ($\mathbf{W}_S^{(l,k)}\mathbf{z}_u^{(l)}$); 3) *w/o Multi* sets $K = 1$ that removes the multiple branches at each layer. We found that all of these components contribute to consistent performance improvements in the three testing domains, which verifies the efficacy of these designs and

Table 3: Testing Accuracy (%) for CIFAR and STL on out-of-distribution data with different domains. We use the k-nearest-neighbor to construct the inter-instance relations in each domain. For STL (resp. CIFAR), different k's (resp. angle bias for computing the cosine similarity) are used for creating distinct domains.

Method	CIFAR			STL		
	150°	160°	170°	$k = 8$	$k = 9$	$k = 10$
ERM-GCN	72.30 ± 0.22	73.16 ± 0.05	72.55 ± 0.30	72.89 ± 0.34	73.27 ± 0.40	74.18 ± 0.23
IRM-GCN	$\mathbf{72.96 \pm 0.32}$	$\mathbf{73.97 \pm 0.44}$	$\mathbf{73.33 \pm 0.08}$	$\mathbf{72.96 \pm 0.38}$	$\mathbf{73.67 \pm 0.22}$	74.28 ± 0.40
GroupDRO-GCN	$\mathbf{72.73 \pm 0.45}$	73.40 ± 0.37	$\mathbf{72.85 \pm 0.24}$	72.79 ± 0.39	73.31 ± 0.19	$\mathbf{74.44 \pm 0.67}$
DANN-GCN	72.19 ± 0.34	73.12 ± 0.19	72.34 ± 0.37	72.77 ± 0.22	73.48 ± 0.56	74.38 ± 0.73
Mixup-GCN	72.66 ± 0.51	$\mathbf{73.49 \pm 0.38}$	72.60 ± 0.49	$\mathbf{73.04 \pm 0.09}$	$\mathbf{73.99 \pm 0.19}$	$\mathbf{74.68 \pm 0.57}$
EERM-GCN	71.01 ± 0.95	72.29 ± 1.35	71.16 ± 1.36	72.14 ± 1.30	72.11 ± 1.73	72.07 ± 1.41
GLIND-GCN	$\mathbf{79.24 \pm 0.53}$	$\mathbf{80.26 \pm 0.60}$	$\mathbf{79.39 \pm 0.89}$	$\mathbf{78.00 \pm 0.34}$	$\mathbf{78.63 \pm 0.29}$	$\mathbf{78.17 \pm 0.29}$
ERM-GAT	72.92 ± 0.22	$\mathbf{74.39 \pm 0.18}$	73.22 ± 0.14	73.29 ± 0.36	73.38 ± 0.58	74.15 ± 0.74
IRM-GAT	$\mathbf{72.96 \pm 0.17}$	73.97 ± 0.38	$\mathbf{73.33 \pm 0.10}$	72.30 ± 1.42	73.16 ± 0.63	74.22 ± 1.24
GroupDRO-GAT	72.87 ± 0.18	74.07 ± 0.09	73.21 ± 0.11	$\mathbf{73.38 \pm 0.19}$	$\mathbf{73.70 \pm 0.12}$	74.61 ± 0.51
DANN-GAT	72.81 ± 0.38	74.16 ± 0.15	73.23 ± 0.45	$\mathbf{73.45 \pm 0.40}$	$\mathbf{73.70 \pm 0.19}$	$\mathbf{74.90 \pm 0.54}$
Mixup-GAT	$\mathbf{72.98 \pm 0.17}$	$\mathbf{74.41 \pm 0.19}$	$\mathbf{73.65 \pm 0.13}$	73.36 ± 0.21	$\mathbf{74.28 \pm 0.17}$	$\mathbf{74.75 \pm 0.01}$
EERM-GAT	71.03 ± 1.69	72.82 ± 1.87	71.84 ± 1.14	71.31 ± 1.01	72.65 ± 1.89	71.96 ± 1.55
GLIND-GAT	$\mathbf{78.23 \pm 0.80}$	$\mathbf{79.01 \pm 0.63}$	$\mathbf{77.98 \pm 0.88}$	$\mathbf{77.91 \pm 0.84}$	$\mathbf{79.17 \pm 0.23}$	$\mathbf{78.81 \pm 0.66}$
ERM-Trans	76.88 ± 0.11	77.51 ± 0.25	76.35 ± 0.28	76.53 ± 0.25	77.10 ± 0.65	77.90 ± 0.22
IRM-Trans	76.53 ± 0.03	77.11 ± 0.05	76.42 ± 0.31	76.95 ± 0.14	77.49 ± 0.25	78.02 ± 0.35
GroupDRO-Trans	76.94 ± 0.65	76.99 ± 0.31	76.37 ± 0.53	$\mathbf{77.81 \pm 0.59}$	78.01 ± 0.54	78.10 ± 0.27
DANN-Trans	76.91 ± 0.17	77.13 ± 0.37	76.61 ± 0.30	77.64 ± 0.13	78.29 ± 0.54	$\mathbf{78.19 \pm 0.35}$
Mixup-Trans	$\mathbf{77.49 \pm 0.39}$	$\mathbf{77.91 \pm 0.14}$	$\mathbf{77.45 \pm 0.34}$	77.76 ± 0.30	$\mathbf{78.32 \pm 0.57}$	$\mathbf{78.73 \pm 0.76}$
EERM-Trans	$\mathbf{79.68 \pm 0.51}$	$\mathbf{79.89 \pm 0.32}$	$\mathbf{78.82 \pm 0.54}$	$\mathbf{77.92 \pm 0.93}$	$\mathbf{78.58 \pm 0.20}$	78.18 ± 0.38
GLIND-TRANS	$\mathbf{80.72 \pm 0.39}$	$\mathbf{81.06 \pm 0.32}$	$\mathbf{80.24 \pm 0.38}$	$\mathbf{78.06 \pm 0.46}$	$\mathbf{79.39 \pm 0.28}$	$\mathbf{78.41 \pm 0.57}$

our model architecture induced by the principled diffusion framework. For the learning objective, we compare with two variants: 1) *w/o Reg* removes the regularization term (the KL loss in Eqn. 9); 2) *w/o Mix* replaces the prior distribution in Eqn. 10 with the average of variational posterior $p_0(\mathbf{d}^{(l)}) \approx 1/N \sum_{u=1}^{N} q(\mathbf{d}^{(l)}|\mathbf{z}^{(l)} = \mathbf{z}_u^{(l)})$. We found that these two variants lead to some performance drops, which verifies the effectiveness of our learning objective.

Impact of Numbers of Diffusivity Hypothesis. We proceed to study the impact of the hyper-parameter K, i.e., the number of diffusivity hypothesis at each message passing layer, on the model performance. Fig. 5(b) presents the testing accuracy of GLIND-GCN on Arxiv with K ranging from 2 to 10 (more results are deferred to Appendix D). We found that setting K as a moderate value can bring up overall the optimal performance. If K is too small, the model would have insufficient capacity for learning multi-faceted data interactions from complex underlying generative mechanisms; if K is too large, the model would become overly complicated which might impair generalization.

6. Conclusion

This paper has explored a geometric diffusion framework for generalization with interdependent data. The model is generalized from the diffusion process on manifolds with stochastic diffusivity function generated from a variational distribution, conditioned on the diffusion dynamics trajectories. We further develop a re-weighting regularization approach to guide the diffusion model to learn causal relations from inputs to outputs throughout the diffusion process that facilitates generalization. Experiments on diverse real-world datasets with different kinds of distribution shifts verify the effectiveness of the proposed models.

Future Work. There can be several promising directions for future investigation. For example, apart from the experimental scenarios and datasets considered in this paper, the proposed methodology and general ideas can be potentially applied and extended to other tasks where out-of-distribution generalization matters, e.g., molecule representations (Yang et al., 2022c). Another problem closely related to out-of-distribution generalization is out-of-distribution detection (Wu et al., 2023a; Li et al., 2022), whose goal is to enable the model to identify testing data from different distributions than training. It also remains under-explored how the diffusion framework explored in this paper can be used for guiding the methodology in the context of out-of-distribution detection with interdependent data.

Impact Statement

This work presents an attempt to improve the generalization for handling interdependent data. Researching the generalization of machine learning models holds profound implications for advancing the reliability and applicability of artificial intelligence systems. Understanding and enhancing the generalization capabilities of models is paramount for deploying them across diverse real-world scenarios. Improved generalization not only enhances model performance on unseen data but also addresses ethical concerns related to bias and fairness. Furthermore, research in this domain contributes to the creation of models that are transferable across different domains, fostering broader adoption of machine learning solutions. As the impact of AI continues to grow across various sectors, unraveling the intricacies of generalization becomes instrumental in ensuring the responsible and equitable deployment of machine learning technologies.

Acknowledgement

The work was in part supported by NSFC (92370201, 62222607, 72342023).

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Belkin, M. and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11), 2006.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2178–2186, 2011.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv:1509.00519*, 2015.

Chamberlain, B., Rowbottom, J., Gorinova, M. I., Bronstein, M. M., Webb, S., and Rossi, E. GRAND: graph neural diffusion. In *International Conference on Machine Learning (ICML)*, pp. 1407–1418, 2021a.

Chamberlain, B. P., Rowbottom, J., Eynard, D., Giovanni, F. D., Dong, X., and Bronstein, M. M. Beltrami flow and neural diffusion on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020.

Christakopoulou, K., Traverse, M., Potter, T., Marriott, E., Li, D., Haulk, C., Chi, E. H., and Chen, M. Deconfounding user satisfaction estimation from response rate bias. In *RecSys*, pp. 450–455, 2020.

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *ICLR*, 2015.

Freidlin, M. I. and Wentzell, A. D. Diffusion processes on graphs and the averaging principle. *The Annals of probability*, pp. 2215–2245, 1993.

Fu, D. and He, J. Dppin: A biological repository of dynamic protein-protein interaction network data. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 5269–5277. IEEE, 2022.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning (ICML)*, pp. 2839–2848, 2016.

Hoffman, M. D. and Johnson, M. J. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NeurIPS*, 2016.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *NeurIPS 2020*, 2020a.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020b.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, pp. 5637–5664, 2021.

Li, Z., Wu, Q., Nie, F., and Yan, J. Graphde: A generative framework for debiased learning and out-of-distribution detection on graphs. In *Advances in Neural Information Processing Systems*, 2022.

Liu, M., Gao, H., and Ji, S. Towards deeper graph neural networks. In Gupta, R., Liu, Y., Tang, J., and Prakash, B. A. (eds.), *KDD*, pp. 338–348. ACM, 2020.

Ma, J., Deng, J., and Mei, Q. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems*, 34:1048–1061, 2021.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*, 2009.

Medvedev, G. S. The nonlinear heat equation on dense graphs and graph limits. *SIAM Journal on Mathematical Analysis*, 46(4):2743–2766, 2014.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, pp. 10–18, 2013.

Pearl, J., Glymour, M., and Jewell, N. P. Causal inference in statistics: A primer. *John Wiley & Sons*, 2016.

Roberts, M. E., Stewart, B. M., and Nielsen, R. A. Adjusting for confounding with text matching. *American Journal of Political Science*, 64(4):887–903, 2020.

Romeny, B. M. H. *Geometry-driven diffusion in computer vision*, volume 1. Springer Science & Business Media, 2013.

Rosenberg, S. and Steven, R. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. Number 31. Cambridge University Press, 1997.

Rozemberczki, B. and Sarkar, R. Twitch gamers: a dataset for evaluating proximity preserving and structural role-based node embeddings, 2021.

Rubin, D. B. Essential concepts of causal inference: a remarkable history and an intriguing future. *Biostatistics & Epidemiology*, 3:140–155, 2019.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Stadler, M., Charpentier, B., Geisler, S., Zügner, D., and Günnemann, S. Graph posterior network: Bayesian predictive uncertainty for node classification. *Advances in Neural Information Processing Systems*, 34:18033–18048, 2021.

Tenenbaum, J. B., Silva, V. d., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Thorpe, M., Xia, H., Nguyen, T., Strohmer, T., Bertozzi, A. L., Osher, S. J., and Wang, B. GRAND++: graph neural diffusion with a source term. In *International Conference on Learning Representations (ICLR)*, 2022.

Tomczak, J. and Welling, M. Vae with a vampprior. In *AISTATS*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Weickert, J. et al. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

Wu, Q., Zhang, H., Yan, J., and Wipf, D. Handling distribution shifts on graphs: An invariance perspective. In *International Conference on Learning Representations*, 2022.

Wu, Q., Chen, Y., Yang, C., and Yan, J. Energy-based out-of-distribution detection for graph neural networks. In *International Conference on Learning Representations*, 2023a.

Wu, Q., Yang, C., Zeng, K., Nie, F., Bronstein, M., and Yan, J. Advective diffusion transformers for topological generalization in graph learning. *CoRR*, abs/2310.06417, 2023b.

Wu, Q., Yang, C., Zhao, W., He, Y., Wipf, D., and Yan, J. DIFFormer: Scalable (graph) transformers induced by energy constrained diffusion. In *International Conference on Learning Representations*, 2023c.

Yang, C., Wu, Q., Wen, Q., Zhou, Z., Sun, L., and Yan, J. Towards out-of-distribution sequential event prediction: A causal treatment. In *Advances in Neural Information Processing Systems*, 2022a.

Yang, C., Wu, Q., and Yan, J. Geometric knowledge distillation: Topology compression for graph neural networks. In *Advances in Neural Information Processing Systems*, 2022b.

Yang, N., Zeng, K., Wu, Q., Jia, X., and Yan, J. Learning substructure invariance for out-of-distribution molecular representations. In *Advances in Neural Information Processing Systems*, 2022c.

Ye, N., Li, K., Hong, L., Bai, H., Chen, Y., Zhou, F., and Li, Z. Ood-bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *CoRR*, abs/2106.03721, 2021.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhao, X., Chen, F., Hu, S., and Cho, J.-H. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems*, 33:12827–12836, 2020.

Appendix

A. Derivation and Proof

A.1. Derivation for Eqn. 7

The log-likelihood for observed data $\log p_\theta(\mathbf{y}|\mathbf{x}, \mathcal{G})$ can be computed by

$$\begin{aligned}
\log p_\theta(\mathbf{y}|\mathbf{x}, \mathcal{G}) &= \log \sum_{\mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(L-1)}} \prod_{l=0}^{L-1} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathcal{G}) \\
&= \sum_{l=0}^{L-2} \log \sum_{\mathbf{z}^{(l+1)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathcal{G}) + \log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)}, \mathcal{G}) \\
&= \sum_{l=0}^{L-2} \log \sum_{\mathbf{z}^{(l+1)}} \mathbb{E}_{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}[p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})] \\
&+ \log \mathbb{E}_{p(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}[p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)}, \mathbf{d}^{(L-1)}, \mathcal{G})],
\end{aligned} \tag{11}$$

where the marginalization over $\mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(L-1)}$ (which is omitted in Eqn. 5 for brevity) is through the delta predictive distributions $p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})$. Due to the intractable integration over $\mathbf{d}^{(l)}$, we consider the evidence lower bound of Eqn. 11 by introducing $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$:

$$\begin{aligned}
&\sum_{l=0}^{L-2} \mathbb{E}_{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \left[\log \sum_{\mathbf{z}^{(l+1)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G}) \frac{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \right] \\
&+ \mathbb{E}_{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} \left[\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)}, \mathbf{d}^{(L-1)}, \mathcal{G}) \frac{p(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} \right] \\
&= \sum_{l=0}^{L-2} \mathbb{E}_{q_\phi(\mathbf{d}^{(0)}|\mathbf{z}^{(0)}), \cdots, q_\phi(\mathbf{d}^{(L-2)}|\mathbf{z}^{(L-2)})} [\log \sum_{\mathbf{z}^{(l+1)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})] \\
&+ \mathbb{E}_{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} [\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)}, \mathbf{d}^{(L-1)}, \mathcal{G})] + \sum_{l=0}^{L-1} \mathbb{E}_{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \left[\log \frac{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \right] \\
&= \mathbb{E}_{q_\phi(\mathbf{d}^{(0)}|\mathbf{z}^{(0)}), \cdots, q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} [\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(0)}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(L-1)}, \mathcal{G})] \\
&- \sum_{l=0}^{L-1} \mathrm{KL}(q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}), p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})).
\end{aligned} \tag{12}$$

This gives rise to the objective of Eqn. 7.

A.2. Proof for Theorem 3.1

Before the proof, we first introduce two fundamental rules of *do*-calculus (Pearl et al., 2016) which will be used as the building blocks later. Consider a causal directed acyclic graph $\mathcal{A}$ with three nodes: B, D and E. We denote $\mathcal{A}_{\overline{B}}$ as the intervened causal graph by cutting off all arrows coming into B, and $\mathcal{A}_{\underline{B}}$ as the graph by cutting off all arrows going out from B. For any interventional distribution compatible with $\mathcal{A}$, the *do*-calculus induces the following two fundamental rules.

i) Action/observation exchange:

$$P(d|do(b), do(e)) = P(d|do(b), e), \text{ if } (D \perp\!\!\!\perp E|B)_{\mathcal{A}_{\overline{B}\underline{E}}}.$$

ii) Insertion/deletion of actions:

$$P(d|do(b), do(e)) = P(d|do(b)), \text{ if } (D \perp\!\!\!\perp E|B)_{\mathcal{A}_{\overline{BE}}}.$$

Back to our case where we study the causal graph comprised of $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{d}$. We have

$$\begin{aligned} p_\theta(\mathbf{y}|do(\mathbf{x}),\mathcal{G}) &= \sum_{\mathbf{d}} p_\theta(\mathbf{y}|do(\mathbf{x}),\mathbf{d},\mathcal{G})p(\mathbf{d}|do(\mathbf{x})) \\ &= \sum_{\mathbf{d}} p_\theta(\mathbf{y}|\mathbf{x},\mathbf{d},\mathcal{G})p(\mathbf{d}|do(\mathbf{x})) \\ &= \sum_{\mathbf{d}} p_\theta(\mathbf{y}|\mathbf{x},\mathbf{d},\mathcal{G})p_0(\mathbf{d}), \end{aligned} \tag{13}$$

where the first step is given by the law of total probability, the second step is according to the first rule (since $\mathbf{y} \perp\!\!\!\perp \mathbf{x}|\mathbf{d}$ in $\mathcal{A}_{\underline{\mathbf{x}}}$), and the third step is due to the second rule (since we have $\mathbf{d} \perp\!\!\!\perp \mathbf{x}$ in $\mathcal{A}_{\overline{\mathbf{x}}}$). The above derivation shows that $p_\theta(\mathbf{y}|do(\mathbf{x}),\mathcal{G}) = \mathbb{E}_{p_0(\mathbf{d})}[p_\theta(\mathbf{y}|\mathbf{x},\mathbf{d},\mathcal{G})]$ where p_0 is the prior distribution of the diffusivity.

Now consider the diffusion model that induces a trajectory $\mathbf{x} = \mathbf{z}^{(0)} \to \mathbf{z}^{(1)} \to \cdots \to \mathbf{z}^{(L)} = \mathbf{y}$. By treating $\{\mathbf{d}^{(l)}\}_{l=0}^{L-1}$ as a whole random variable, we can extend the result of Eqn. 13 and obtain

$$p_\theta(\mathbf{y}|do(\mathbf{x}),\mathcal{G}) = \sum_{\mathbf{d}^{(0)},\cdots,\mathbf{d}^{(L-1)}} p_\theta(\mathbf{y}|\mathbf{x},\mathbf{d}^{(0)},\cdots,\mathbf{d}^{(L-1)},\mathcal{G})p_0(\mathbf{d}^{(0)},\cdots,\mathbf{d}^{(L-1)}). \tag{14}$$

Notice that $p_0(\mathbf{d}^{(0)},\cdots,\mathbf{d}^{(L-1)}) = \prod_{l=0}^{L-1} p_0(\mathbf{d}^{(l)})$. By inserting the delta predictive distribution $p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G})$ induced by the diffusion model, we can derive the log-likelihood:

$$\begin{aligned} &\log p_\theta(\mathbf{y}|do(\mathbf{x}),\mathcal{G}) \\ =& \log \sum_{\mathbf{z}^{(1)},\cdots,\mathbf{z}^{(L-1)}} \sum_{\mathbf{d}^{(0)},\cdots,\mathbf{d}^{(L-1)}} \prod_{l=0}^{L-1} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G})p_0(\mathbf{d}^{(l)}) \\ =& \sum_{l=0}^{L-2} \log \sum_{\mathbf{z}^{(l+1)}} \sum_{\mathbf{d}^{(l)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G})p_0(\mathbf{d}^{(l)}) + \log \sum_{\mathbf{d}^{(L-1)}} p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)},\mathbf{d}^{(L-1)},\mathcal{G})p_0(\mathbf{d}^{(L-1)}) \\ =& \sum_{l=0}^{L-2} \log \sum_{\mathbf{z}^{(l+1)}} \sum_{\mathbf{d}^{(l)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G})p_0(\mathbf{d}^{(l)}) + \log \sum_{\mathbf{d}^{(L-1)}} p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)},\mathbf{d}^{(L-1)},\mathcal{G})p_0(\mathbf{d}^{(L-1)}) \end{aligned} \tag{15}$$

We next separately consider the derivation for the two terms in the above equation. For the first term in Eqn. 15, using Jensen Inequality we have

$$\begin{aligned} &\sum_{l=0}^{L-2} \log \sum_{\mathbf{z}^{(l+1)}} \sum_{\mathbf{d}^{(l)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G})p_0(\mathbf{d}^{(l)}) \\ =& \sum_{l=0}^{L-2} \log \sum_{\mathbf{z}^{(l+1)}} \sum_{\mathbf{d}^{(l)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G})p_0(\mathbf{d}^{(l)}) \frac{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \\ \geq& \sum_{l=0}^{L-2} \sum_{\mathbf{d}^{(l)}} q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}) \cdot \left[\log \sum_{\mathbf{z}^{(l+1)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)},\mathbf{d}^{(l)},\mathcal{G}) \frac{p_0(\mathbf{d}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}\right]. \end{aligned} \tag{16}$$

For the second term in Eqn. 15, similarly we have

$$\begin{aligned} &\log \sum_{\mathbf{d}^{(L-1)}} p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)},\mathbf{d}^{(L-1)},\mathcal{G})p_0(\mathbf{d}^{(L-1)}) \\ \geq& \sum_{\mathbf{d}^{(L-1)}} q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)}) \cdot \left[\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)},\mathbf{d}^{(L-1)},\mathcal{G}) \frac{p_0(\mathbf{d}^{(L-1)})}{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}\right]. \end{aligned} \tag{17}$$

By combing the results of Eqn. 16 and Eqn. 17 with Eqn. 15, we have the variational lower bound of $\log p_\theta(\mathbf{y}|do(\mathbf{x}), \mathcal{G})$:

$$\begin{aligned}&\sum_{l=0}^{L-2} \mathbb{E}_{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \left[\log \sum_{\mathbf{z}^{(l+1)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G}) \frac{p_0(\mathbf{d}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}\right] \\ &+\mathbb{E}_{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} \left[\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)}, \mathbf{d}^{(L-1)}, \mathcal{G}) \frac{p_0(\mathbf{d}^{(L-1)})}{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}\right],\end{aligned} \tag{18}$$

where the marginalization over $\mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(L-1)}$ (which is omitted in Eqn. 8 for brevity) is through the delta predictive distribution $p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G})$.

Furthermore, in Eqn. 16 the equality holds if and only if $q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}) = p(\mathbf{d}^{(l)}|\mathbf{z}^{(l+1)}, \mathbf{z}^{(l)}, \mathcal{G}) \cdot \frac{p_0(\mathbf{d}^{(l)})}{p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}$ in which case the function inside the expectation stays as a constant w.r.t. $\mathbf{d}^{(l)}$. Similarly, for Eqn. 17, the equality holds if and only if $q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)}) = p(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L)}, \mathbf{z}^{(L-1)}, \mathcal{G}) \cdot \frac{p_0(\mathbf{d}^{(L-1)})}{p(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}$.

A.3. Derivation for Eqn. 9

The objective Eqn. 9 can be derived in a similar way as Eqn. 12 on the basis of Eqn. 18. In specific, we leverage the model-free prior $p_0(\mathbf{d}^{(l)})$ to replace $p(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})$ in the first line of Eqn. 12:

$$\begin{aligned}&\sum_{l=0}^{L-2} \mathbb{E}_{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})} \left[\log \sum_{\mathbf{z}^{(l+1)}} p_\theta(\mathbf{z}^{(l+1)}|\mathbf{z}^{(l)}, \mathbf{d}^{(l)}, \mathcal{G}) \frac{p_0(\mathbf{d}^{(l)})}{q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)})}\right] \\ &+\mathbb{E}_{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})} \left[\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(L-1)}, \mathbf{d}^{(L-1)}, \mathcal{G}) \frac{p_0(\mathbf{d}^{(L-1)})}{q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}\right] \\ &=\mathbb{E}_{q_\phi(\mathbf{d}^{(0)}|\mathbf{z}^{(0)}), \cdots, q_\phi(\mathbf{d}^{(L-1)}|\mathbf{z}^{(L-1)})}[\log p_\theta(\mathbf{z}^{(L)}|\mathbf{z}^{(0)}, \mathbf{d}^{(0)}, \cdots, \mathbf{d}^{(L-1)}, \mathcal{G})] \\ &-\sum_{l=0}^{L-1} \mathrm{KL}(q_\phi(\mathbf{d}^{(l)}|\mathbf{z}^{(l)}), p_0(\mathbf{d}^{(l)})).\end{aligned} \tag{19}$$

B. Dataset Introduction

`Twitch` (Rozemberczki & Sarkar, 2021) is a social network dataset which comprises multiple disconnected subgraphs. Each subgraph is a social network from a particular region and contains thousands of node instances that are interconnected. Each node is a use on Twitch and the node label is the gender of the user. Edges denote the friendship among users. Since in this dataset, nodes from different subgraphs can be seen as samples from distinct distributions (Table 4 shows different statistics of these subgraphs), we split the training and testing data according to subgraphs. Specifically, we use the nodes from subgraphs DE, PT and RU as training data, and the nodes from subgraphs ES, FR and EN as testing data.

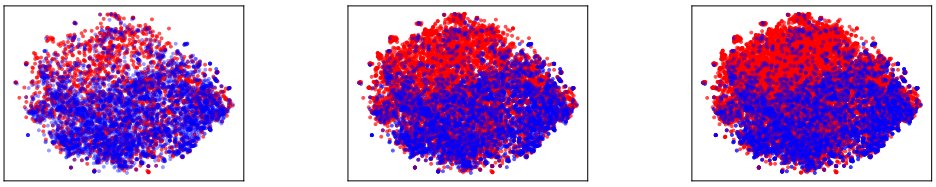
`Arxiv` is a citation network dataset provided by (Hu et al., 2020a). It records the citation relationship between papers and the goal is to predict the paper's subarea based on the word embeddings of paper keywords as the node features. The dataset also provides the time information about when the paper is published, and we use the publication year to split the training and testing sets. Since the research subareas and citation relationships of papers are sensitive to time, there natural exist distribution shifts between papers published in different time windows. In our experiments, we use papers published before 2014 for training, and papers published after 2014 for testing. We further divide the testing data into three subsets comprised of the papers published within 2014-2016, 2016-2018 and 2018-2020, respectively. As verification for the distribution shifts, Table 4 visualizes the different distributions of training and testing instances.

`CIFAR-10` and `STL-10` are two image datasets where each image is an instance and there are no observed structures interconnecting the instances. Following the recent work (Wu et al., 2023c), we use all the 13000 images, each of which belongs to one of ten classes, for `STL-10`, and choose 1500 images from each of 10 classes for `CIFAR-10`. The input features of image instances are pre-computed by using the self-supervised approach SimCLR (Chen et al., 2020) (that does not use labels for training) and training a ResNet-18 that extracts the feature map as the input features of each instance. We use the k-nearest-neighbor to construct the inter-instance relational structures for these datasets, and consider different k's and distance functions to introduce the distribution shifts. More details about how we choose the values of k and distance functions are presented in Sec. 5.1.

Table 4: Statistics for different subgraphs of Twitch.

Datasets	DE	PT	RU	ES	FR	EN
# Instances	9498	1912	4385	4648	6549	7126
Density	0.0033	0.0171	0.0038	0.0054	0.0052	0.0013
Max Node Degree	3475	455	575	809	1517	465
Mean Node Degree	16	16	8	12	17	4

DPPIN (Fu & He, 2022) is comprised of 12 individual dynamic network datasets at different scales, and each network records the protein-protein interactions. Each dynamic network is obtained by one protein identification method, which can be seen as one domain, and consists of 36 snapshots. Each protein has a sequence of continuous scalar features with 36 time stamps, which records the evolution of gene expression values within metabolic cycles of yeast cells. The interconnecting structures among protein instances are determined by co-expressed protein pairs at one time, and can change as the co-expression activities evolve with time. We consider the regression task for the gene expression value of each protein instance within one snapshot, and ignore the temporal dependence between different snapshots. In specific, the model input contains the proteins' gene expression value in the previous ten snapshot (as input features) and the interaction structures at the current snapshot, based on which the prediction target is the gene expression value of proteins at the current snapshot. As observed by (Fu & He, 2022), the protein instances from different datasets have distinct topological patterns (e.g., the distributions of cliques), so we split the training and testing data based on different datasets.



(a) 2014-2016 test instances (b) 2016-2018 test instances (c) 2018-2020 test instances

Figure 4: T-SNE visualization of input features for training instances (blue) and testing instances (red) on Arxiv.

C. Implementation Details

Our implementation is based on PyTorch 1.9.0 and PyTorch Geometric 2.0.3. All of our experiments are run on a Tesla V100 with 16 GB memory. We adopt Adam with weight decay for training. We set a fixed training budget with 1000 epochs for DPPIN and 500 epochs for other datasets. The testing performance achieved by the epoch where the model gives the best performance on validation data is reported.

The detailed architecture of GLIND is described as follows. The model architecture consists of the following modules in sequential order:

- A fully-connected layer with hidden size $D \times d$ (transforming D-dim input raw features into d-dim embeddings).
- L-layer message passing network with hidden size d (each layer contains K branches that have independent parameterization), based on the three instantiations in Sec. 4.
- A fully-connected layer with hidden size $d \times C$ (mapping d-dim embeddings to C classes).

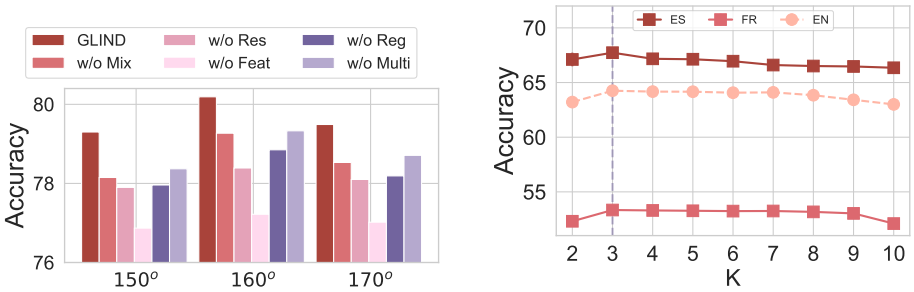
In each layer, we use ReLU activation, dropout and the residual link where the weight α is set as 0.5 across all datasets.

The probability estimation for each branch is modeled by a feed-forward layer with hidden size $d \times K$ in each layer as introduced in Sec. 4. The sampling process for the selected branch is approximated by Gumbel Softmax. For computing the prior distribution in the objective, we introduce a pseudo dataset of size T as described in Eqn. 10. We set $T = \text{int}(0.01 * N)$, where N denotes the number of instances in the dataset, for all the datasets.

For other hyper-parameters, we search them for each dataset with grid search on the validation set. The searching spaces for all the hyper-parameters are as follows.

- Number of message passing layers L: [2, 3, 4, 5].
- Hidden dimension d: [32, 64, 128].
- Dropout ratio: [0.0, 0.1, 0.2, 0.5].
- Learning rate: [0.001, 0.005, 0.01, 0.02].
- Weight decay: [0, 5e-5, 5e-4, 1e-3].
- Number of diffusivity hypothesis K: [3, 4, 5, 10].
- Gumbel Softmax temperature τ: [1, 2, 3, 5].

D. Additional Experiment Results



(a) Accuracy on CIFAR (b) Accuracy on Twitch

Figure 5: (a) Ablation studies for GLIND on CIFAR. (b) Performance of GLIND on three testing sets of Twitch with different numbers of diffusivity K's.