TRACI: A Data-centric Approach for Multi-Domain Generalization on Graphs

Yusheng Zhao[1], Changhu Wang[2], Xiao Luo[3,*], Junyu Luo[1], Wei Ju[1], Zhiping Xiao[4,*], Ming Zhang[1*]

[1]State Key Laboratory for Multimedia Information Processing,
School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China
[2]Department of Statistics, University of California, Los Angeles, CA, USA
[3]Department of Computer Science, University of California, Los Angeles, CA, USA
[4]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA
{yusheng.zhao, luojunyu}@stu.pku.edu.cn, wangch156@g.ucla.edu,
xiaoluo@cs.ucla.edu, {juwei, mzhang_cs}@pku.edu.cn, patxiao@uw.edu

Abstract

Graph neural networks (GNNs) have gained superior performance in graph-based prediction tasks with a variety of applications such as social analysis and drug discovery. Despite the remarkable progress, their performance often degrades on test graphs with distribution shifts. Existing domain adaptation methods rely on unlabeled test graphs during optimization, limiting their applicability to graphs in the wild. Towards this end, this paper studies the problem of multi-domain generalization on graphs, which utilizes multiple source graphs to learn a GNN with high performance on unseen target graphs. We propose a new approach named Topological Adversarial Learning with Prototypical Mixup (TRACI) to solve the problem. The fundamental principle behind our TRACI is to produce virtual adversarial and mixed graph samples from a data-centric view. In particular, TRACI enhances GNN generalization by employing a gradient-ascent strategy that considers both label prediction entropy and graph topology to craft challenging adversarial samples. Additionally, it generates domain-agnostic node representations by characterizing class-graph pair prototypes through latent distributions and applying multi-sample prototypical Mixup for distribution alignment across graphs. We further provide theoretical analysis showing that TRACI reduces the model's excess risk. Extensive experiments on various benchmark datasets demonstrate that TRACI outperforms state-of-the-art baselines, validating its effectiveness.

Introduction

Graph neural networks (GNNs) have demonstrated promising performance in various graph machine learning problems (Kipf and Welling 2017; Ju et al. 2024a). Learning latent representations of these graph structures is beneficial for various downstream tasks (Lu et al. 2021; Chen et al. 2022b; Huang et al. 2024). Graph neural networks usually learn node representations using the neighborhood aggregation mechanism where each node updates the representations using its neighborhood information iteratively (Xu et al. 2019). These informative node representations would be incorporated into various downstream applications including social network analysis (Zhang et al. 2022b), drug-drug interaction (Ma and Lei 2023), recommender systems (Liu et al. 2021), and traffic flow forecasting (Zhao et al. 2023; Ju et al. 2024b; Li et al. 2024).

Despite their immense prevalence, GNNs are data-hungry and usually require the accessibility of target nodes. In particular, early efforts are hugely made in transductive learning (Kipf and Welling 2017; Tang and Liu 2023), which infers unlabeled nodes from a graph with partially labeled nodes. Recently, graph domain adaptation (Guo et al. 2022; Wang et al. 2024b; Liu et al. 2024) has been put forward to transmit knowledge from a label-rich graph to a label-scarce graph and handle distribution shifts across different graphs at the same time. These methods usually develop more advanced GNNs for representation learning and incorporate domain adaptation techniques to reduce the domain shift. However, these methods require the availability of all the target nodes, while real-world applications are always with a range of unseen target graphs in the wild. Therefore, in this paper, we study a novel yet under-explored problem of multi-domain generalization on graphs, which learns a powerful GNN using a set of training graphs (domains) so that it can generalize effectively on unseen test graphs.

However, designing a new multi-domain generalization GNN framework remains challenging, which requires us to answer the following two questions. Firstly, *how to enhance the generalization capacity during optimization?* Since we cannot acquire any target graphs, the core of multi-domain generalization is to enhance the generalization capacity of GNNs. Although domain generalization has been studied in Euclidean data such as images and texts (Chen et al. 2022a; Hsu et al. 2023; Long et al. 2024), the problem on graphs would confront structured data, which requires us to take the structural semantics into consideration. Secondly, *how to reduce the distribution shift across multiple domains?* Domain shifts on graphs include both structural shifts and nodal shifts. Structural shifts refer to differences in link construction, which could bring in various densities of the graphs. In contrast, nodal shifts refer to distribution discrepancy in nodal attributes across graphs. Therefore, it is highly challenging to generate domain-invariant and discriminative node representations across various graphs.

To answer the following two questions, in this work, we propose a new approach named Topological Adversarial

*Corresponding authors.

Learning with Prototypical Mixup (TRACI). The core of our TRACI is to produce virtual adversarial and mixed graph samples from a data-centric view [1]. In particular, to boost the generalization capacity of GNNs, we add perturbations to both node attributes and latent features. A new objective is proposed for adversarial sample generation, which aims to not only maximize the entropy of label predictions and maximize the average distance of the neighboring nodes in graphs. In this way, we can generate challenging graph samples with high-frequency graph noise for robust optimization. In addition, to generate domain-invariant representations, we first characterize the prototype of each graph-class pair using a Gaussian distribution in the latent space and then propose a multi-sample Mixup strategy for mixed virtual samples. Different from previous methods which mix representations from two nodes (Wang et al. 2021; Park, Shim, and Yang 2022), our TRACI introduces prototypical learning to generate infinite augmented views by sampling from distributions. We also incorporate contrastive learning between different mixed views to enhance the alignment between graphs. We conduct extensive experiments on various benchmark datasets. Experimental results validate the superiority of the proposed TRACI over a range of baselines. The contribution of this work is highlighted as follows:

- **Problem Formulation.** We investigate a practical yet underexplored problem of multi-domain generalization on graphs, which aims to learn a powerful GNN using a set of training graphs and expects the learned model to generalize effectively on unseen target graphs.
- **New Methodology.** We propose a new approach named TRACI, which not only adopts projected gradient ascend with the consideration of both entropy and graph structure for hard adversarial samples, but also conducts multi-sample prototypical Mixup to enhance the graph alignment in the latent space.
- **Theoretical Insight.** We provide a theoretical analysis demonstrating that the proposed method reduces the excess risk of the model.
- **Empirical Validation.** Extensive experiments on various benchmark datasets validate the superiority of the proposed TRACI over a range of baselines.

Related Works

Domain Generalization aims to construct a model that is capable of generalizing to unobserved target domains using multiple source domains (Li et al. 2017, 2018a; Wang et al. 2022a, 2024a). One potential solution to domain generalization is to minimize the distribution discrepancy for domain-invariant representations in the embedding space. A range of approaches (Ghifary et al. 2016; Li et al. 2020; Wang et al. 2024a) usually introduce different metrics to measure the distribution discrepancy such as maximum mean discrepancy (Li et al. 2018b), and second-order correlation (Ghifary et al. 2016; Jin et al. 2020). Another line to generate domain-invariant representations is to utilize adversarial learning (Rahman et al. 2020; Dayal et al. 2024), which involves a domain discriminator with a gradient reserve layer. Recently, self-supervised learning has been incorporated into this problem, which involves various self-supervised tasks to enhance the generalization of representations (Carlucci et al. 2019; Maniyar et al. 2020). However, domain generalization is mostly studied in texts and images while the problem in non-Euclidean graph data remains underexplored. Therefore, in this paper, we explore the multi-domain generalization problem on graph data, and utilize rich structural information for better generalization capability on graphs in the face of domain shifts.

Graph Transfer Learning (Lee et al. 2017) has been a long-standing problem that aims to transfer a pre-trained model to a new problem. Early efforts have been made for graph domain adaptation (Wu et al. 2020; Guo et al. 2022; Wang et al. 2024b; Liu et al. 2024; Yin et al. 2022), which transfers knowledge from source graphs into target graphs. These methods usually incorporate domain alignment techniques such as adversarial learning into graph neural networks. Recently, out-of-distribution problem (Gui et al. 2022; Liu et al. 2023; Luo et al. 2023b,a) has received increasing attention, in which test data are not in the same distribution of training data. However, these methods often fail to utilize information across domains (Wu et al. 2022) or allow the use of testing input for training (Liu et al. 2023), which would be difficult in real-world applications and thus different from domain generalization. Therefore, in this paper, we propose a new approach named TRACI for graph domain generalization, which produces virtual adversarial and mixed graph samples from a data-centric view.

The Proposed TRACI

Problem Definition. This work studies the problem of multi-domain generalization on graphs, where M source graphs $\{G^1, \cdots, G^M\}$ are involved with $G^m = \{V^m, E^m\}$. V^m and $E^m \subset V^m \times V^m$ represent the node set and the edge set for the m-th graph, respectively. $\boldsymbol{X}^m \in \mathbb{R}^{N^m \times d_{in}}$ represents the node attribute matrix of G^m where d_{in} is the attribute dimension and N^m is the number of nodes. Let $\boldsymbol{Y}^m$ denote the label matrix in which $\boldsymbol{Y}^m_{ik} = 1$ if node i is associated with the k-th category and $\boldsymbol{Y}^m_{ik} = 0$ if not. These graphs share the same label space, i.e., $\{1, 2, \ldots, C\}$ but have different data distributions and topological structures. We aim to learn a graph machine learning model f using these source graphs to achieve superior performance on unknown target graphs.

Framework Overview. In this paper, we propose a new approach named TRACI to solve the multi-domain generalization on graphs problem. The core of our TRACI is to generate adversarial and mixed virtual samples from a data-centric perspective. In particular, our TRACI mainly has the following two modules: (1) *Topological Adversarial Learning for Challenging Samples*. To increase the generalization capacity of GNNs, we adopt projected gradient ascend with the consideration of both entropy and graph structure for gen-

[1] The data-centric view emphasizes an approach that focuses on transforming and enriching the data itself, particularly through synthetic data generation and manipulation, rather than only optimizing the model's architecture or parameters.

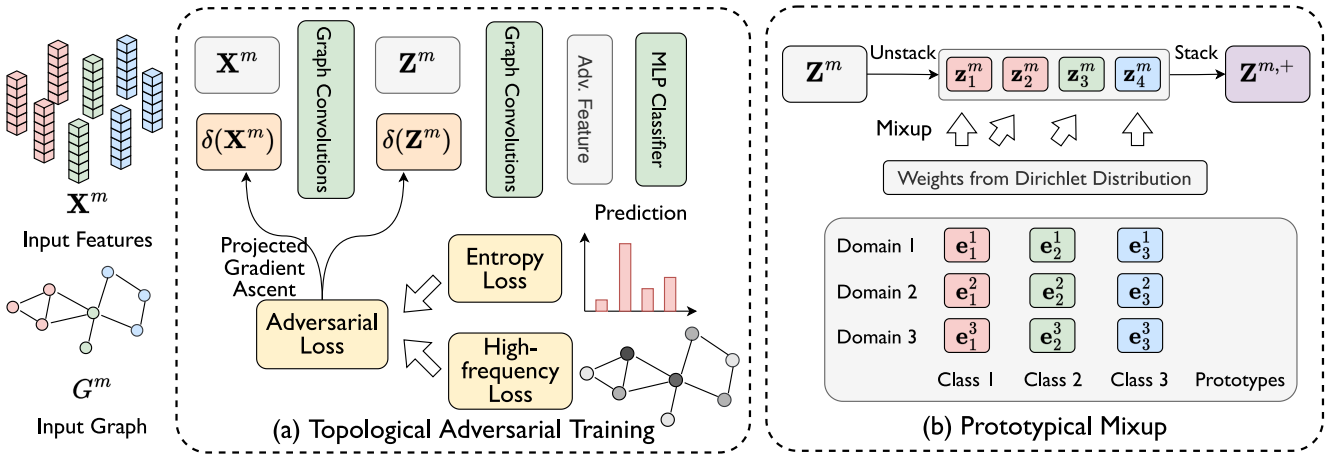


Figure 1: The framework of the proposed TRACI consisting of (a) topological adversarial learning, and (b) prototypical Mixup.

erating hard adversarial samples. (2) *Prototypical Mixup for Graph Alignment*. To decrease the discrepancy of multiple domains, we estimate the latent distributions of prototypes for class-graph pairs and conduct multi-sample prototypical Mixup to enhance the graph alignment. An overview of our proposed TRACI is shown in Figure 1.

Topological Adversarial Learning for Challenging Samples

Since GNNs would be evaluated on unseen target graphs, the major goal of domain generalization on graphs is to increase the generalization capacity of GNNs. To achieve it, we introduce a data-centric perspective (Yin et al. 2022; Yang, Zhang, and Yang 2021; Yang et al. 2023a), which generates adversarial virtual samples using structural perturbations to both node attributes and hidden layers for robust optimization. Here, we adopt projected gradient ascent (Madry et al. 2017) with the consideration of both the entropy of predicted distributions and the topological structure for challenging and feasible adversarial samples. These challenging graph samples would be fed into GNNs for further optimization.

In detail, given a source graph G^m with attribute matrix, we can add noise to $\boldsymbol{X}^m$:

$$\tilde{\boldsymbol{X}}^m = \boldsymbol{X}^m + \delta(\boldsymbol{X}^m), \tag{1}$$

where $\delta(\cdot)$ returns the perturbation matrix (*i.e.* the noise). Here, we turn to projected gradient ascent to ensure the perturbed attributes within the feasible region, which is still close to the original one. Previous works (Szegedy et al. 2013; Wang et al. 2022b) usually adopt the loss function for adversarial learning. Here, we move forward by considering the entropy of the prediction and graph structure. On the one hand, the entropy is maximized to generate confusing samples following the adversarial learning routine. On the other hand, we encourage our perturbation dissimilar to the graph structure for challenging samples. In formulation, the loss function for adversarial learning is written as:

$$\begin{aligned}\mathcal{L}_{adv}^a = \sum_{m=1}^{M}(\sum_{i\in V^m} -\tilde{\boldsymbol{Y}}_i^m \log \tilde{\boldsymbol{Y}}_i^m \\ + \omega \operatorname{tr}(\delta(\boldsymbol{X}^m)^T(\boldsymbol{D}^m - \boldsymbol{A}^m)\delta(\boldsymbol{X}^m))),\end{aligned} \tag{2}$$

where $\boldsymbol{D}^m$ denotes the degree matrix from the adjacency matrix $\boldsymbol{A}^m$, ω is the weight of the structure dissimilarity term (second term), and $\operatorname{tr}(\cdot)$ computes the trace of a matrix. Here, the first term calculates the entropy of each node, while the second term can be written as:

$$\mathcal{S} = \sum_{ij} \boldsymbol{A}_{ij}^m \left\| \delta(\boldsymbol{X}_i^m) - \delta(\boldsymbol{X}_j^m) \right\|^2, \tag{3}$$

where $\delta(\boldsymbol{X}_i^m)$ denotes the perturbation of each node (Weiss, Torralba, and Fergus 2008). $\mathcal{S}$ calculates the average distance of perturbation vectors between neighbors of graphs. A small $\mathcal{S}$ would indicate a low-frequency signal related to the graph structure. We expect the perturbation would be irrelevant to the graph structure for challenging adversarial samples. Low-frequent noise is more consistent with the neighborhood information (graph structure), which makes easy samples, while high-frequent noise makes challenging samples (Liu et al. 2022). Thus, by maximizing this adversarial objective, the more challenging high-frequent perturbation is added to the adversarial samples.

Then, we perturb the input within a small region to keep the semantics. In formulation, the node attribute is updated at the step t using projected gradient ascent as follows:

$$\tilde{\boldsymbol{X}}^{m,t+1} = \Pi_\epsilon \left(\tilde{\boldsymbol{X}}^{m,t} + \alpha \operatorname{sgn}\left(\nabla_{\boldsymbol{X}^m} \mathcal{L}_{adv}^a\right) \right), \tag{4}$$

where Π_ϵ projects the perturbed attributes back to the compact set $\{\tilde{\boldsymbol{X}}^m | ||\tilde{\boldsymbol{X}}^m - \boldsymbol{X}^m|| < \epsilon\}$ and ϵ is a preset small parameter to decide the range of perturbation. α denotes the learning rate. After T-step updating, we conduct adversarial attacks on the input for subsequent optimization, which can enhance the robustness of the GNN model. However, these adversarial samples do not involve the graph structure. To tackle this, we further add perturbation to the output of the

first GNN layer $\boldsymbol{Z}^m$, i.e., $\tilde{\boldsymbol{Z}}^m = \boldsymbol{Z}^m + \delta(\boldsymbol{Z}^m)$. As the GNNs utilize the graph structure by neighborhood aggregation, the hidden embeddings naturally contain structural information. Then, Eqn. 2 can be written as:

$$\begin{aligned}\mathcal{L}_{adv}^{f} = \sum_{m=1}^{M} (\sum_{i \in V^m} -\tilde{\boldsymbol{Y}}_i^m \log \tilde{\boldsymbol{Y}}_i^m \\ + \omega \delta(\boldsymbol{Z}^m)^T (\boldsymbol{D}^m - \boldsymbol{A}^m) \delta(\boldsymbol{Z}^m)).\end{aligned} \tag{5}$$

Similarly, we would rewrite Eq. 4 for adversarial learning in the latent space:

$$\tilde{\boldsymbol{Z}}^{m,t+1} = \Pi_{\epsilon} \left(\tilde{\boldsymbol{Z}}^{m,t} + \alpha \operatorname{sgn} \left(\nabla_{\boldsymbol{Z}^m} \mathcal{L}_{adv}^{f} \right) \right), \tag{6}$$

where ϵ decides the range of feature perturbation and α denotes the learning rate. We incorporate both types of perturbations into graphs for extensive adversarial virtual samples.

Remark. Traditional adversarial training (Wang et al. 2022b; Andriushchenko and Flammarion 2020) usually utilizes the same loss objective during parameter optimization and adversarial sample generation. In contrast, our TRACI utilizes a different loss objective for adversarial sample generation, which considers both the entropy of label prediction and graph structures. The introduction of entropy can tackle potential label shifts, while the high-frequency noise is added for the generation of challenging samples while incorporating topological information.

Prototypical Mixup for Graph Alignment

Furthermore, a powerful and generalized GNN should be invariant to environmental changes, focusing on target semantics. To achieve this, a data-centric insight is to generate virtual samples using multiple domains to amplify target semantics (Yang et al. 2023b; Han et al. 2022; Zhang 2017). Towards this end, we introduce a prototypical Mixup to enhance the graph alignment. Our prototypical Mixup first estimates the latent distributions in the hidden space for all prototypes of class-graph pairs, and then selects prototypes for each node. Then, a multi-sample Mixup strategy is used to blend latent features with virtual samples from latent distributions with cross-domain interaction.

In particular, we characterize the prototype of each class-graph pair as a Gaussian distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in the hidden space, incorporating node attributes into topological structures through GNNs. Then, each hidden feature corresponding to this prototype can be viewed as a sample from the latent distribution. Through them, we can estimate the mean $\boldsymbol{\mu}_c^m$ and covariance matrix $\boldsymbol{\Sigma}_c^m$ for the c-th prototype of G^m. Formally, they can be written as:

$$\boldsymbol{\mu}_c^m = \frac{\sum_{i=1}^{N^m} \mathbf{1}_{y_i=c} \boldsymbol{z}_i^m}{\sum_{i=1}^{N^m} \mathbf{1}_{y_i^m=c}}, \tag{7}$$

$$\boldsymbol{\Sigma}_c^m = \frac{\sum_{i=1}^{N^m} \mathbf{1}_{y_i^m=c} (\boldsymbol{z}_i^m - \boldsymbol{\mu}_c^m) (\boldsymbol{z}_i^m - \boldsymbol{\mu}_c^m)^T}{\sum_{i=1}^{N^m} \mathbf{1}_{y_i^m=c}}, \tag{8}$$

where $\mathbf{1}_{\cdot}$ is an indicator for all samples with the same class. To reduce the computational complexity, we would utilize a diagonal covariance matrix for each latent distribution.

Then, given a node i, we would produce virtual samples from distributions with the same semantics. Then, a multiple-sample Mixup (Zhang et al. 2022a; Abhishek, Brown, and Hamarneh 2024; Jin et al. 2024) strategy is leveraged to generate augmented latent features. Considering that we have M prototypical distributions for each class, we introduce a Dirichlet distribution, which considers every source graph equally as follows:

$$(\lambda_1, \cdots, \lambda_m) \sim Dir(\frac{1}{M}, \cdots, \frac{1}{M}). \tag{9}$$

We can also change the parameters in the Dirichlet distribution when we have some prior about target graphs. Moreover, γ comes from a Beta distribution randomly to decide the ratio for original features:

$$\gamma \sim Beta(\beta_1, \beta_2), \tag{10}$$

in which β_1 and β_2 are two coefficients. In practice, we set β_2 to 1. Finally, the augmented features of node i with label c from G^m can be written as:

$$\boldsymbol{z}_i^{m,+} = \gamma \boldsymbol{z}_i^m + (1-\gamma)(\sum_{m'=1}^{M} \lambda^{m'} \boldsymbol{e}_c^{m'})), \tag{11}$$

where $\boldsymbol{e}_c^{m'}$ is sampled from the c-th prototype in $G^{m'}$ to indicate virtual samples. We stack these virtual nodes into a matrix, denoted as $\boldsymbol{Z}^{m,+}$.

Our prototypical Mixup has the following benefits as follows: Firstly, through mixing with virtual samples, we can extend the Empirical Risk Minimization (ERM) principle to a more advanced Vicinal Risk Minimization (VRM) (Chapelle et al. 2000; Zhang, Hsieh, and Tao 2018; Zhang 2017) principle, which has shown superior generalization capacity by enlarging the datasets. Secondly, we incorporate prototypes from different source graphs to generate infinite augmented views, which can mitigate the distribution discrepancy and thus produce unbiased label predictions. Thirdly, compared with Mixup using features of real nodes (Han et al. 2022), we can save computational costs by storing and sampling the distributions with a global view of prototypical semantics.

Summarization

In a nutshell, we generate two groups of virtual samples from different perspectives, i.e., adversarial samples and mixed samples. These samples would be combined and fed into our graph neural network to generate predictions, and the cross-entropy objective is:

$$\mathcal{L}_{mce} = \sum_{m=1}^{M} CE(\tilde{\boldsymbol{Y}}^m, \boldsymbol{Y}^m), \tag{12}$$

where $CE(\cdot, \cdot)$ is the cross-entropy and $\tilde{\boldsymbol{Y}}^m$ is label distributions from virtual samples, i.e., $\tilde{\boldsymbol{X}}^{m,T}$, $\tilde{\boldsymbol{Z}}^{m,T}$ and $\boldsymbol{Z}^{m,+}$.

Moreover, to further reduce the domain discrepancy in the latent space, we introduce supervised contrastive learning (Chen et al. 2020; Khosla et al. 2020) with prototypical Mixup, which enforces different views of augmented nodes

to be consistent compared with different nodes. In particular, we regularize the difference between augmented nodes and the original nodes using contrastive learning, formulated as:

$$\mathcal{L}_{cl} = -\sum_{m=1}^{M}\sum_{v\in G^m} \log \frac{exp(\boldsymbol{h}_v^{m,+} \star \boldsymbol{h}_v^m)}{\sum_{v'\in G^m} exp(\boldsymbol{h}_v^{m,+} \star \boldsymbol{h}_{v'}^m)} \quad (13)$$

where $\boldsymbol{h}_v^{m,+}$ and $\boldsymbol{h}_v^m$ are deep features after L GNN layers. $\star$ calculates the cosine similarity of two vectors. Our contrastive learning can further align node representations with mixed information from different graphs to reduce the domain shift, by encouraging query and augmented views with potential domain shifts to be close compared with the other nodes.

The final objective of the proposed TRACI can be summarized as follows:

$$\mathcal{L} = \mathcal{L}_{mce} + \lambda \mathcal{L}_{cl}, \quad (14)$$

where λ is a parameter to balance two loss objectives.

For the optimization procedure, we first warmup the model using the source graphs. At each epoch, we first generate adversarial (using projected gradient ascent with Eqn. 2 and 5) and virtual (using Eqn. 11) samples. Then we optimize the model's parameters using these generated data with Eqn 14.

Theoretical Analysis

This section aims to provide a theoretical justification for the integration of virtual samples into graph neural networks (GNNs), a step necessitated by the observed distribution shifts across different graph structures (Bartlett, Montanari, and Rakhlin 2021). We begin with a brief overview of key notations. Remember, $\boldsymbol{X}^m$ is the observed node attribute matrix, and $\boldsymbol{Y}^m$ the label matrix, with the distribution of observed samples $(\boldsymbol{X}_i^m, \boldsymbol{Y}_i^m)$ for $m = 1, \ldots, M$ and $i = 1, \ldots, N_m$ denoted as $P^m(\boldsymbol{x}, \boldsymbol{y})$. Here, the indices $\boldsymbol{X}_i^m$ and $\boldsymbol{Y}_i^m$ correspond to the i-th row in their respective matrices. A significant challenge in GNNs is addressed by the distribution $P^m(\boldsymbol{x}, \boldsymbol{y})$, which highlights how data distribution changes across multiple graphs. To manage this challenge, we introduce $P_{\text{new}}(\boldsymbol{x}, \boldsymbol{y})$, representing the distribution for a new graph's samples $(\boldsymbol{X}_i^*, \boldsymbol{Y}_i^*)$, where $i = 1, \ldots, N_{\text{new}}$. For simplicity, and without loss of generality, we narrow our focus to the case where $m = 1$, thus referring to $P^1(\boldsymbol{x}, \boldsymbol{y})$ as simply $P(\boldsymbol{x}, \boldsymbol{y})$. It is critical to note that $P_{\text{new}}(\boldsymbol{x}, \boldsymbol{y})$ differs from $P(\boldsymbol{x}, \boldsymbol{y})$, emphasizing the crucial role of distribution shifts in our analysis. To bridge this gap, we posit the existence of a virtual sample distribution, $P_{\text{virt}}(\boldsymbol{x}, \boldsymbol{y})$, underscoring the essential role these samples play in enhancing the adaptability and efficacy of GNNs in dynamic environments.

We consider all observed, new, and virtual samples as independently and identically distributed (i.i.d.) from their respective distributions: $P(\boldsymbol{x}, \boldsymbol{y})$ for observed, $P_{\text{new}}(\boldsymbol{x}, \boldsymbol{y})$ for new, and $P_{\text{virt}}(\boldsymbol{x}, \boldsymbol{y})$ for virtual samples, with virtual samples indexed as $(\boldsymbol{X}_i^\dagger, \boldsymbol{Y}_i^\dagger)$ for $i = 1, \ldots, N_{\text{virt}}$. We assess the predictive performance of a function f, projecting the input space $\mathcal{X}$ to the output space $\mathcal{Y}$, through a loss $\ell(f(\boldsymbol{X}), \boldsymbol{Y})$, allowing for either 0/1 classification loss or cross-entropy loss. The risk function is defined formally by:

$$L(f) = \mathbb{E}_P[\ell(f(\boldsymbol{X}), \boldsymbol{Y})], \quad L_{\text{new}}(f) = \mathbb{E}_{P_{\text{new}}}[\ell(f(\boldsymbol{X}), \boldsymbol{Y})],$$

and

$$L_{\text{virt}}(f) = \mathbb{E}_{P_{\text{virt}}}[\ell(f(\boldsymbol{X}), \boldsymbol{Y})],$$

leading to the functions:

$$f^* = \operatorname*{argmin}_{f\in\mathcal{F}} L(f), \quad f^*_{\text{virt}} = \operatorname*{argmin}_{f\in\mathcal{F}} L_{\text{virt}}(f),$$

where $\mathcal{F}$ denotes the function space modeled by GNNs. Empirical risks, based on these samples, are calculated as:

$$\begin{aligned} \widehat{L}(f) &:= \frac{1}{N}\sum_{i=1}^{N} \ell(f(\boldsymbol{X}_i), \boldsymbol{Y}_i), \\ \widehat{L}_{\text{virt}}(f) &:= \frac{1}{N_{\text{virt}}}\sum_{i=1}^{N_{\text{virt}}} \ell(f(\boldsymbol{X}_i^\dagger), \boldsymbol{Y}_i^\dagger), \end{aligned} \quad (15)$$

leading to estimators:

$$\widehat{f} = \operatorname*{argmin}_{f\in\mathcal{F}} \widehat{L}(f), \quad \widehat{f}_{\text{virt}} = \operatorname*{argmin}_{f\in\mathcal{F}} \widehat{L}_{\text{virt}}(f).$$

Then, we define the excess risk of $\widehat{f}$ and $\widehat{f}_{\text{virt}}$ as follows:

$$\begin{aligned} R(\widehat{f}) &= L_{\text{new}}(\widehat{f}) - \inf_{f\in\mathcal{F}} L_{\text{new}}(f), \\ R(\widehat{f}_{\text{virt}}) &= L_{\text{new}}(\widehat{f}_{\text{virt}}) - \inf_{f\in\mathcal{F}} L_{\text{new}}(f). \end{aligned} \quad (16)$$

Theorem 1 *Considering a binary classification problem, in which $\mathcal{Y} = \{\pm 1\}$, we define the ± 1 loss, $\ell_{\pm 1}(\widehat{y}, y) = -y\widehat{y}$. For $\mathcal{F} \subseteq [-1, 1]^{\mathcal{X}}$, we have*

$$\begin{aligned} \mathbb{E}\left\{R(\widehat{f}_{virt}) - R(\widehat{f})\right\} \leq & \mathbb{E}\left\{\widehat{L}_{new}(\widehat{f}) - \widehat{L}_{new}(\widehat{f}_{virt})\right\} \\ & - 4\sqrt{\frac{2\log\left(2\Pi_{\mathcal{F}}(n)\right)}{n}} \end{aligned} \quad (17)$$

where $\Pi_{\mathcal{F}}(n) = \max\{|\{(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)) : f \in \mathcal{F}\}| : \boldsymbol{x}_{1\ldots n}, \ldots, \boldsymbol{x}_n \in \mathcal{X}\}$ is the VC-dimension of $\mathcal{F}$ on $\mathcal{X}$.

Theorem 1 establishes that when $\sqrt{\frac{2\log(2\Pi_{\mathcal{F}}(n))}{n}} \to 0$, indicating the VC-dimension of $\mathcal{F}$ is suitably constrained, the expected difference in excess risk between $R(\widehat{f})$ and $R(\widehat{f}_{\text{virt}})$ can be effectively bounded. Specifically, if $P_{\text{virt}}(\boldsymbol{x}, \boldsymbol{y})$ more closely approximates $P_{\text{new}}(\boldsymbol{x}, \boldsymbol{y})$ than $P(\boldsymbol{x}, \boldsymbol{y})$ does, the difference $\widehat{L}_{\text{new}}(\widehat{f}) - \widehat{L}_{\text{new}}$ will exceed zero, suggesting that incorporating virtual samples enhances generalizability. Importantly, when $P_{\text{virt}}(\boldsymbol{x}, \boldsymbol{y})$ exactly matches $P_{\text{new}}(\boldsymbol{x}, \boldsymbol{y})$, $\widehat{L}_{\text{new}}(\widehat{f}) - \widehat{L}_{\text{new}}$ is positive with high probability, thereby ensuring that the expected difference, $\widehat{L}_{\text{new}}(\widehat{f}) - \widehat{L}_{\text{new}}$, remains above zero, indicating the benefits of virtual sample integration on model performance.

Methods	AC → D		AD → C		CD → A	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
GCN (Kipf and Welling 2017)	65.62±0.30	61.46±0.27	63.87±1.18	59.06±1.05	61.99±0.54	59.24±2.00
GIN (Xu et al. 2019)	59.46±0.70	53.09±1.26	38.85±5.05	32.29±3.85	13.30±4.19	10.37±3.90
GAT (Veličković et al. 2018)	53.34±0.95	47.16±0.84	50.02±0.79	45.99±0.39	52.15±0.78	48.37±1.52
SGC (Wu et al. 2019)	63.54±1.06	55.24±1.37	50.04±5.21	39.73±4.95	38.89±4.53	29.93±4.34
EGC (Tailor et al. 2022)	64.08±0.56	59.24±0.51	55.46±1.65	50.43±3.50	48.88±3.19	43.83±4.58
ADA (Volpi et al. 2018)	66.06±1.11	61.06±0.42	61.26±1.73	56.63±1.21	60.68±0.85	57.17±1.85
MAT (Wang et al. 2022b)	65.81±0.61	60.35±0.74	58.90±0.72	53.60±0.65	58.49±0.59	50.95±0.49
FLOOD (Liu et al. 2023)	67.26±0.50	63.34±0.67	64.46±1.86	59.07±1.88	60.16±0.60	55.92±0.40
MARIO (Zhu et al. 2024)	66.94±0.18	61.84±0.26	65.81±1.50	60.75±1.05	63.76±0.39	58.76±2.67
TRACI (Ours)	**72.95±0.77**	**69.25±1.22**	**74.20±0.68**	**69.08±1.07**	**67.01±1.04**	**65.03±0.35**

Table 1: The prediction accuracy on the Citation benchmark.

Methods	P1	P2	P3	P4	Average
GCN	49.24	44.62	34.18	47.97	44.00
GIN	42.92	50.79	41.23	44.88	44.96
GAT	26.69	51.03	37.97	33.01	37.18
SGC	42.37	46.19	32.19	37.56	39.58
EGC	49.93	40.75	31.83	41.14	40.91
ADA	49.11	44.14	34.18	48.29	43.93
MAT	48.97	44.14	34.18	48.13	43.86
FLOOD	50.48	52.60	34.18	37.25	44.39
MARIO	49.70	52.92	38.82	45.04	46.62
TRACI	**51.58**	**58.65**	**49.73**	**48.78**	**52.18**

Table 2: The prediction accuracy on the Proteins benchmark.

Experiments

Experimental Settings

Datasets. Two benchmarks are used in the experiments: Citations and Proteins. The Citations benchmark consists of three real-world networks from (Tang et al. 2008), *i.e.* ACMv9, Citationv1 and DBLPv7, collected by different sources during different periods. The Proteins benchmark consists of four protein datasets from the PDB database (Berman et al. 2000) with their single-chain data as graphs. The secondary structure types of every residue are our prediction target, computed using DSSP (Joosten et al. 2010).

Baselines Methods. Our method is compared with a number of baseline methods, including standard graph neural network methods, *i.e.* GCN (Kipf and Welling 2017), GIN (Xu et al. 2019), GAT (Veličković et al. 2017), SGC (Wu et al. 2019) and EGC (Tailor et al. 2022), domain generalization methods for Euclidean data, *i.e.* ADA (Volpi et al. 2018) and MAT (Wang et al. 2022b), and graph structured data, *i.e.* FLOOD (Liu et al. 2023) and MARIO (Zhu et al. 2024).

Evaluation and Implementation Details. We employ the leave-one-domain-out protocol (Dubey et al. 2021). Micro and Macro F1 are used for evaluation for the Citations benchmark, while accuracy is reported for the Proteins benchmark. For hyperparameters, ϵ is set to 0.6, ω is 0.02, λ in Eq.14 is 0.1, and L is 2. For details, see https://github.com/YushengZhao/TRACI-AAAI25/.

The Performance of TRACI

The results for TRACI compared to baseline methods are presented in Table 1 and Table 2. According to the results, we have the following observations: **Firstly**, TRACI achieves a consistent lead in both Citation and Proteins benchmarks, compared to the baselines, which shows the overall effectiveness of the method. **Secondly**, Among the baselines for domain generalization, ADA and MAT are proposed for Euclidean data (e.g. images or texts). By design, they do not consider the structure information in the data, which makes them weak in the face of distribution shift of the graph structure. **Thirdly**, for the baselines designed for graph structured data, FLOOD utilizes simple augmentations like feature masking and edge dropping, which may not generate challenging samples. For MARIO, although it generates adversarial samples for training, it adopts a naive adversarial loss and only perturbs the graph attributes. In contrast, we design novel adversarial loss objectives (Eqn. 2 and 5) that consider both the entropy of label predictions and the graph structure. Moreover, the adversarial perturbations are added to both the graph attributes and the hidden features. This explains our superior performance.

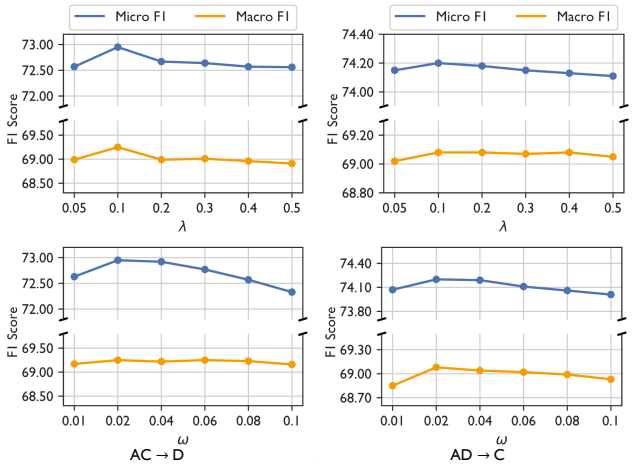


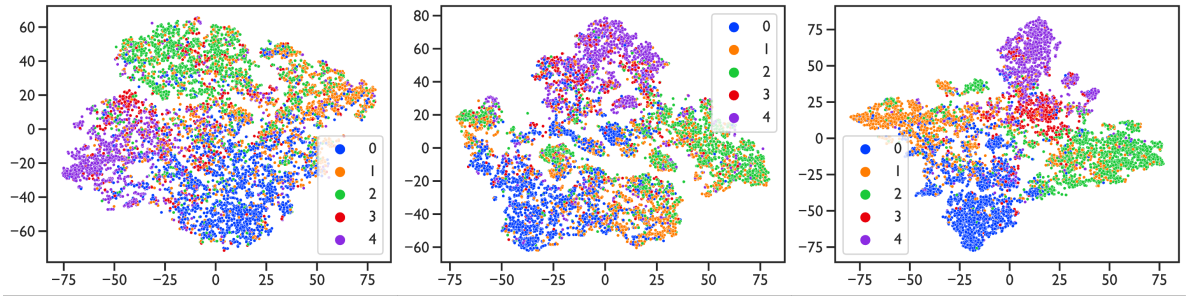
Figure 2: The parameter sensitivity experiments of TRACI on the Citations benchmark.

Sensitivity Analysis

In this subsection, the model's sensitivity to hyperparameters is explored. Concretely, we focus on two hyperparameters: (1) the weight of contrastive learning loss (*i.e.* λ in Eq. 14), and (2) the weight of high-frequent perturbation in the adversarial objective (*i.e.* ω in Eq. 2). The experimen-

Exp. No.	PM	TAL			AC → D		AD → C		CD → A	
		F	A	S	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
1					65.62±0.30	61.46±0.27	63.87±1.18	59.06±1.05	61.99±0.54	59.24±2.00
2	✓				67.57±0.01	64.23±0.08	65.99±0.05	61.41±0.32	62.11±0.42	59.91±1.45
3	✓	✓			69.10±0.58	65.68±0.61	67.67±0.20	63.18±0.34	63.27±0.34	61.31±0.32
4	✓	✓	✓		72.04±0.81	68.98±0.40	72.76±1.06	68.57±0.54	66.25±0.66	63.72±1.44
5	✓	✓	✓	✓	72.95±0.77	69.25±1.22	74.10±0.68	69.08±1.07	67.01±1.04	65.03±0.35

Table 3: Ablation studies on the Citations benchmark.



(a) GCN (b) FLOOD (c) TRACI

Figure 3: Visualization of learned representations on the Citations benchmark in AD→C setting. Three methods are compared, including GCN (a), FLOOD (b), and the proposed TRACI (c).

tal results are shown in Figure 2. As can be seen from the results, the proposed TRACI is generally not sensitive to hyperparameters. The evaluation accuracy changes moderately when we perturb the hyperparameters. Therefore, we set the weight of contrastive learning loss λ to 0.1 and the weight of the high-frequent perturbation term in the adversarial objective (*e.g.* ω) to 0.02.

Ablation Studies

In this subsection, ablated studies are performed to demonstrate the efficacy of each component, including (1) the prototypical Mixup, denoted as "PM", (2) the adversarial perturbation of the hidden features (*i.e.* $\delta(\boldsymbol{Z}^m)$), denoted as "TAL(F)", (3) the adversarial perturbation of the attributes (*i.e.* $\delta(\boldsymbol{X}^m)$), denoted as "TAL(A)", and (4) the high-frequent adversarial perturbation (*i.e.* $\mathcal{S}$), denoted as "TAL(S)". The last three components constitute topological adversarial learning (TAL). The results are shown in Table 3. As can be seen form the results, prototypical Mixup (PM) contributes to the model's improvement. Adding the PM improves the model's performance across settings. Since the prototypical Mixup is able to align graph representations from multiple source domains via generating virtual samples, it is reasonable that it helps with TRACI's performance. Moreover, topological adversarial learning (TAL) contributes significantly to the model's performance. The three components in topological adversarial learning, *i.e.* F, A and S, all contribute to the model's performance (Line 2 to Line 5). This shows that adversarial training considering both entropy and graph structure is beneficial for the generalization capability of the model.

Visualization

In this section, we visualize the learned representations of nodes in the graphs using t-SNE (Rauber, Falcão, and Telea 2016), and the results are shown in Figure 3. The experiments are performed on the Citations benchmark (AD→C), where the three models (GCN, FLOOD and TRACI) are compared. For convenience, we only visualize the nodes that belong to a single category. As can be seen from the results in Figure 3, the proposed TRACI is able to yield more condensed features. For example, in GCN, Class 0 are scattered and mixed with Class 4 and 1. In comparison, our model learns more condensed features for Class 0, as the blue dots tend to form clusters. As for FLOOD model in Figure 3 (b), although some nodes are clustered locally (*e.g.* Class 0 and 2), they are mixed together and hard to separate. By comparison, our model generates more separable node representations. In a nutshell, TRACI is able to generate more condensed feature representations for the target domain, which shows that the proposed techniques help the model to learn robust representations.

Conclusion

This paper investigates the problem of multi-domain generalization on graphs and then proposes a novel method named TRACI for this problem which generates mixed and virtual adversarial graph examples from a data-centric view. We also provide a theoretical analysis showing that our method decreases the excess risk of the model on the target domain. Moreover, we compared the proposed method with numerous competing baselines on several benchmarks to show our superiority. In future work, we would extend the proposed TRACI to a broader range of real-world tasks including social network analysis and anomaly detection.

Acknowledgments

This paper is partially supported by the National Key Research and Development Program of China with Grant No. 2023YFC3341203 as well as the National Natural Science Foundation of China with Grant Numbers 62276002 and 62306014.

References

Abhishek, K.; Brown, C. J.; and Hamarneh, G. 2024. Multi-sample ζ-mixup: richer, more realistic synthetic samples from ap-series interpolant. *Journal of big Data*, 11(1): 43.

Andriushchenko, M.; and Flammarion, N. 2020. Understanding and improving fast adversarial training. *NeurIPS*.

Bartlett, P. L.; Montanari, A.; and Rakhlin, A. 2021. Deep learning: a statistical viewpoint. *Acta Numerica*, 30: 87 – 201.

Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; and Bourne, P. E. 2000. The protein data bank. *Nucleic Acids Research*, 28(1): 235–242.

Carlucci, F. M.; D'Innocente, A.; Bucci, S.; Caputo, B.; and Tommasi, T. 2019. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2229–2238.

Chapelle, O.; Weston, J.; Bottou, L.; and Vapnik, V. 2000. Vicinal risk minimization. *NeurIPS*, 13.

Chen, C.; Li, J.; Han, X.; Liu, X.; and Yu, Y. 2022a. Compound Domain Generalization via Meta-Knowledge Encoding. In *CVPR*, 7119–7129.

Chen, F.; Wang, J.; Wei, Y.; Zheng, H.-T.; and Shao, J. 2022b. Breaking isolation: Multimodal graph fusion for multimedia recommendation by edge-wise modulation. In *ACM MM*, 385–394.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *ICML*.

Dayal, A.; KB, V.; Cenkeramaddi, L. R.; Mohan, C.; Kumar, A.; and N Balasubramanian, V. 2024. MADG: margin-based adversarial learning for domain generalization. *NeurIPS*, 36.

Dubey, A.; Ramanathan, V.; Pentland, A.; and Mahajan, D. 2021. Adaptive methods for real-world domain generalization. In *CVPR*, 14340–14349.

Ghifary, M.; Balduzzi, D.; Kleijn, W. B.; and Zhang, M. 2016. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *TPAMI*.

Gui, S.; Li, X.; Wang, L.; and Ji, S. 2022. Good: A graph out-of-distribution benchmark. *NeurIPS*, 35: 2059–2073.

Guo, G.; Wang, C.; Yan, B.; Lou, Y.; Feng, H.; Zhu, J.; Chen, J.; He, F.; and Yu, P. 2022. Learning Adaptive Node Embeddings across Graphs. *TKDE*.

Han, X.; Jiang, Z.; Liu, N.; and Hu, X. 2022. G-mixup: Graph data augmentation for graph classification. In *ICML*.

Hsu, K.-C.; Ren, A. Z.; Nguyen, D. P.; Majumdar, A.; and Fisac, J. F. 2023. Sim-to-Lab-to-Real: Safe reinforcement learning with shielding and generalization guarantees. *Artificial Intelligence*, 314: 103811.

Huang, K.; Jin, Y.; Candes, E.; and Leskovec, J. 2024. Uncertainty quantification over graph with conformalized graph neural networks. *NeurIPS*, 36.

Jin, X.; Lan, C.; Zeng, W.; and Chen, Z. 2020. Feature alignment and restoration for domain generalization and adaptation. *arXiv preprint arXiv:2006.12009*.

Jin, X.; Zhu, H.; Li, S.; Wang, Z.; Liu, Z.; Yu, C.; Qin, H.; and Li, S. Z. 2024. A Survey on Mixup Augmentations and Beyond. *arXiv preprint arXiv:2409.05202*.

Joosten, R. P.; Te Beek, T. A.; Krieger, E.; Hekkelman, M. L.; Hooft, R. W.; Schneider, R.; Sander, C.; and Vriend, G. 2010. A series of PDB related databases for everyday needs. *Nucleic acids research*, 39(suppl_1): D411–D419.

Ju, W.; Fang, Z.; Gu, Y.; Liu, Z.; Long, Q.; Qiao, Z.; Qin, Y.; Shen, J.; Sun, F.; Xiao, Z.; et al. 2024a. A comprehensive survey on deep graph representation learning. *Neural Networks*, 106207.

Ju, W.; Zhao, Y.; Qin, Y.; Yi, S.; Yuan, J.; Xiao, Z.; Luo, X.; Yan, X.; and Zhang, M. 2024b. Cool: a conjoint perspective on spatio-temporal graph neural network for traffic forecasting. *Information Fusion*, 107: 102341.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *NeurIPS*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Lee, J.; Kim, H.; Lee, J.; and Yoon, S. 2017. Transfer learning for deep learning on graph-structured data. In *AAAI*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2018a. Learning to generalize: Meta-learning for domain generalization. In *AAAI*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *ICCV*.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018b. Domain generalization with adversarial feature learning. In *CVPR*.

Li, H.; Wang, Y.; Wan, R.; Wang, S.; Li, T.-Q.; and Kot, A. 2020. Domain generalization for medical imaging classification with linear-dependency regularization. In *NeurIPS*.

Li, H.; Zhao, Y.; Mao, Z.; Qin, Y.; Xiao, Z.; Feng, J.; Gu, Y.; Ju, W.; Luo, X.; and Zhao, M. 2024. A survey on graph neural networks in intelligent transportation systems. *arXiv e-prints*, arXiv–2401.

Liu, M.; Fang, Z.; Zhang, Z.; Gu, M.; Zhou, S.; Wang, X.; and Bu, J. 2024. Rethinking propagation for unsupervised graph domain adaptation. In *AAAI*.

Liu, N.; Wang, X.; Bo, D.; Shi, C.; and Pei, J. 2022. Revisiting graph contrastive learning from the perspective of graph spectrum. *NeurIPS*.

Liu, Y.; Ao, X.; Feng, F.; Ma, Y.; Li, K.; Chua, T.-S.; and He, Q. 2023. FLOOD: A Flexible Invariant Learning Framework for Out-of-Distribution Generalization on Graphs. In *KDD*.

Liu, Y.; Yang, S.; Lei, C.; Wang, G.; Tang, H.; Zhang, J.; Sun, A.; and Miao, C. 2021. Pre-training graph transformer with multimodal side information for recommendation. In *ACM MM*, 2853–2861.

Long, S.; Zhou, Q.; Ying, C.; Ma, L.; and Luo, Y. 2024. Rethinking domain generalization: Discriminability and generalizability. *TCSVT*.

Lu, X.; Zhu, L.; Liu, L.; Nie, L.; and Zhang, H. 2021. Graph convolutional multi-modal hashing for flexible multimedia retrieval. In *ACM MM*, 1414–1422.

Luo, X.; Zhao, Y.; Mao, Z.; Qin, Y.; Ju, W.; Zhang, M.; and Sun, Y. 2023a. Rignn: A rationale perspective for semi-supervised open-world graph classification. *TMLR*.

Luo, X.; Zhao, Y.; Qin, Y.; Ju, W.; and Zhang, M. 2023b. Towards semi-supervised universal graph classification. *TKDE*, 36(1): 416–428.

Ma, M.; and Lei, X. 2023. A dual graph neural network for drug–drug interactions prediction based on molecular structure and interactions. *PLOS Computational Biology*, 19(1).

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Maniyar, U.; Deshmukh, A. A.; Dogan, U.; Balasubramanian, V. N.; et al. 2020. Zero shot domain generalization. *arXiv preprint arXiv:2008.07443*.

Park, J.; Shim, H.; and Yang, E. 2022. Graph transplant: Node saliency-guided graph mixup with local structure preservation. In *AAAI*, volume 36, 7966–7974.

Rahman, M. M.; Fookes, C.; Baktashmotlagh, M.; and Sridharan, S. 2020. Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition*, 100: 107124.

Rauber, P. E.; Falcão, A. X.; and Telea, A. C. 2016. Visualizing Time-Dependent Data Using Dynamic t-SNE. In *EuroVis*, 73–77.

Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; et al. 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 118(15): e2016239118.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tailor, S. A.; Opolka, F.; Lio, P.; and Lane, N. D. 2022. Do We Need Anisotropic Graph Neural Networks? In *ICLR*.

Tang, H.; and Liu, Y. 2023. Towards understanding generalization of graph neural networks. In *ICML*.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: extraction and mining of academic social networks. In *KDD*, 990–998.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Volpi, R.; Namkoong, H.; Sener, O.; Duchi, J. C.; Murino, V.; and Savarese, S. 2018. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; and Yu, P. 2022a. Generalizing to unseen domains: A survey on domain generalization. *TKDE*.

Wang, M.; Liu, Y.; Yuan, J.; Wang, S.; Wang, Z.; and Wang, W. 2024a. Inter-class and inter-domain semantic augmentation for domain generalization. *TIP*.

Wang, Q.; Wang, Y.; Zhu, H.; and Wang, Y. 2022b. Improving Out-of-Distribution Generalization by Adversarial Training with Structured Priors. *arXiv preprint arXiv:2210.06807*.

Wang, Y.; Wang, W.; Liang, Y.; Cai, Y.; and Hooi, B. 2021. Mixup for node and graph classification. In *WWW*.

Wang, Y.; Zhu, R.; Ji, P.; and Li, S. 2024b. Open-Set Graph Domain Adaptation via Separate Domain Alignment. In *AAAI*, volume 38, 9142–9150.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. *NeurIPS*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *ICML*.

Wu, M.; Pan, S.; Zhou, C.; Chang, X.; and Zhu, X. 2020. Unsupervised domain adaptive graph convolutional networks. In *WWW*, 1457–1467.

Wu, Q.; Zhang, H.; Yan, J.; and Wipf, D. 2022. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *ICLR*.

Xu, Q.; Zhang, R.; Zhang, Y.; Wang, Y.; and Tian, Q. 2021. A fourier-based framework for domain generalization. In *CVPR*, 14383–14392.

Yang, C.; Bo, D.; Liu, J.; Peng, Y.; Chen, B.; Dai, H.; Sun, A.; Yu, Y.; Xiao, Y.; Zhang, Q.; et al. 2023a. Data-centric graph learning: A survey. *arXiv preprint arXiv:2310.04987*.

Yang, L.; Zhang, L.; and Yang, W. 2021. Graph Adversarial Self-Supervised Learning. In *NeurIPS*.

Yang, X.; Wang, H.; Sun, J.; Zhang, S.; Chen, C.; Hua, X.-S.; and Luo, X. 2023b. Prototypical Mixing and Retrieval-based Refinement for Label Noise-resistant Image Retrieval. In *ICCV*, 11239–11249.

Yin, N.; Shen, L.; Li, B.; Wang, M.; Luo, X.; Chen, C.; Luo, Z.; and Hua, X.-S. 2022. Deal: An unsupervised domain adaptive framework for graph-level classification. In *ACMMM*.

Zhang, C.; Hsieh, M.-H.; and Tao, D. 2018. Generalization bounds for vicinal risk minimization principle. *arXiv preprint arXiv:1811.04351*.

Zhang, H. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, S.; Liu, M.; Yan, J.; Zhang, H.; Huang, L.; Yang, X.; and Lu, P. 2022a. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. In *KDD*.

Zhang, Y.; Gao, H.; Pei, J.; and Huang, H. 2022b. Robust Self-Supervised Structural Graph Neural Network for Social Network Prediction. In *WWW*, 1352–1361.

Zhao, Y.; Luo, X.; Ju, W.; Chen, C.; Hua, X.-S.; and Zhang, M. 2023. Dynamic hypergraph structure learning for traffic flow forecasting. In *ICDE*, 2303–2316. IEEE.

Zhu, Y.; Shi, H.; Zhang, Z.; and Tang, S. 2024. Mario: Model agnostic recipe for improving ood generalization of graph contrastive learning. In *WWW*, 300–311.