SPOT: Spectral Optimal Transport for Graph Domain Generalization

YUSHENG ZHAO, State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China
XIAO LUO, Department of Statistics, University of Wisconsin–Madison, Madison, Wisconsin, USA
JUNYU LUO and WEI JU, State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China
ZHONGHUI GU, Peking-Tsinghua Center for Life Sciences, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China
ZHIPING XIAO, Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, Washington, USA
XIAN-SHENG HUA, Terminus Group, Beijing, China
MING ZHANG, State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China

Graph neural networks (GNNs) have essentially taken over as the *de facto* model for learning graph-structured data. However, the majority of existing methods perform transductive learning in a known graph, which is unable to tackle abundant in-the-wild unseen graphs with potential domain shifts. Even worse, these graphs, accompanied by domain shifts on structural topology and node attributes, bring in vulnerable data bias and thus a huge drop in performance. To tackle this, we propose a novel GNN method named spectral optimal transport (SPOT) for effective domain generalization on graphs. Our method is motivated by the fact that the high-frequency graph spectrum is more likely to indicate domain differences. In particular, we formulate the structural augmentation as an optimal transport problem to retain low-frequency key knowledge and solve the problem using Sinkhorn-Knopp algorithm. In addition, we incorporate an adaptive perturbation strategy to deep features, where the direction of the additive noise is determined by the homophily degrees to maintain semantic properties. Accordingly, we meticulously construct a collection of real-world benchmark

Ming Zhang and Yusheng Zhao are supported by grants from the National Key Research and Development Program of China with Grant No. 2023YFC3341203 and the National Natural Science Foundation of China (NSFC Grant Number 62276002).
Authors' Contact Information: Yusheng Zhao, State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China; e-mail: yusheng.zhao@stu.pku.edu.cn; Xiao Luo (corresponding author), Department of Statistics, University of Wisconsin–Madison, Madison, Wisconsin, USA; e-mail: xiao.luo@wisc.edu; Junyu Luo, State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China; e-mail: luojunyu@stu.pku.edu.cn; Wei Ju, State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China; e-mail: juwei@pku.edu.cn; Zhonghui Gu, Peking-Tsinghua Center for Life Sciences, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China; e-mail: guzhh20@stu.pku.edu.cn; Zhiping Xiao (corresponding author), Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, Washington, USA; e-mail: patxiao@uw.edu; Xian-Sheng Hua, Terminus Group, Beijing, China; e-mail: huaxiansheng@gmail.com; Ming Zhang (corresponding author), State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University, Beijing, China; e-mail: mzhang_cs@pku.edu.cn.

ACM 1556-472X/2025/11-ART6
https://doi.org/10.1145/3772720

datasets to assess the domain generalization capability of our model on graphs, and extensive experiments confirm the effectiveness of our proposed SPOT.

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Learning latent representations**;

Additional Key Words and Phrases: Domain Generalization, Node Classification, Graph Neural Networks, Optimal Transport

Associate Editor: Suhang Wang

ACM Reference format:
Yusheng Zhao, Xiao Luo, Junyu Luo, Wei Ju, Zhonghui Gu, Zhiping Xiao, Xian-Sheng Hua, and Ming Zhang. 2025. SPOT: Spectral Optimal Transport for Graph Domain Generalization. *ACM Trans. Knowl. Discov. Data.* 20, 1, Article 6 (November 2025), 22 pages.
https://doi.org/10.1145/3772720

1 Introduction

The real world is always filled with abundant graph-structured data [90], which could manifest as social networks [41, 99], biological networks [55, 96], and transportation networks [43, 111, 112]. **Graph neural networks (GNNs)** have provided an effective tool for graph data mining [5, 41, 94, 115]. In particular, they often use a message passing mechanism that aggregates information from neighboring nodes to update the central nodes, bringing in superior node classification performance.

Despite their tremendous popularity, GNNs are primarily studied in common semi-supervised [11, 23, 41, 88, 98] and domain adaptive settings [16, 28, 56, 89]. The former concentrates on transductive learning on a single graph, while the latter transfers information from label-rich graph data to label-scarce graph data. They both demand that the test nodes be in a known graph, which constrains their applications in real-world scenarios with a variety of unknown test graphs. Two essential components of graphs, i.e., nodes and edges, indicate two different forms of domain variance in test graphs, i.e., structural shifts and nodal shifts. The first denotes shifts in adjacency matrices, which could originate from a variety of sources and standards to create edge connectivities. The second refers to distribution shifts of node attributes, which may come from the changes in the background or environment [53, 114]. To fill up this gap, this study investigates the problem of domain generalization on graphs, which calls for learning GNNs from a heterogeneous collection of training graphs to generalize well on unseen test graphs.

Although domain generalization on Euclidean data has been well studied in previous works [8, 44, 105, 117], formalizing a domain-invariant framework on graphs is still a difficult task because of two difficulties. First, model performance could suffer from structural shifts between graphs. It has been proven that GNNs are vulnerable to edge variance during message passing [15]. However, in practical applications, edges are always sensitive to temporal and source variation [28]. For instance, social networks allow for the addition or deletion of user relationships over time. Even worse, several sources (such as DBLP and ACM) could generate multiple relational graphs for the same collection of objectives [89]. Therefore, it is highly anticipated to train GNN models invariant to these structural variances. Second, the majority of current efforts on domain generalization are focused on **independent and identically distributed (i.i.d.)** data [8, 20, 21, 39, 45, 51, 68, 69, 79, 80, 87, 95, 103, 108]. Hence, they cannot be utilized for capturing distribution shifts in relational nodes on graphs. Specifically, it is crucial to take the relational structures into consideration when dipping into attribute distribution in case of semantics loss. It is important to note that there has been a preliminary development of **out-of-distribution (OOD)** generalization on graphs [47, 48, 104, 119]. In OOD generalization, in-distribution (source) data and OOD (target) data are usually on

the same graph [27, 30], whereas in domain generalization, the source domains and target domains are different graphs, making the task more challenging.

In this article, we provide a novel perspective for domain generalization on graphs. Our motivation originates from a basic observation that the high-frequency signals are more related to the distribution shifts rather than key semantics in signal processing fields [53, 59, 82]. We deduce that the high-frequency component in the graph spectrum is more likely to indicate domain difference, while the low-frequency component in the graph spectrum would retain the key semantic information. To validate this, we try various popular graph augmentation strategies in recent works [31, 100, 120] and validate that retaining low-frequency components in the graph spectrum can obtain better performance for out-of-domain generalization.

Based on the above insight, we propose a novel GNN method named **spectrum optimal transport (SPOT)** for effective domain generalization on graphs. Optimal transport is initially proposed to measure the distribution divergence, which has been extended to search for global solutions in various fields [7, 14, 17]. Here, we use optimal transport to perform spectrum-aware structural augmentation, which retains low-frequency semantics knowledge in the graph spectrum. In particular, we encourage maximizing global alignment between the edge variance matrix and a high-pass prior matrix inside the transportation polytope to generate balanced and appropriate perturbations for each node. The optimization problem is then solved by Sinkhorn-Knopp algorithm [14], resulting in a semantic-preserving fictitious structure with potential domain shifts. In addition to structure augmentation using optimal transport, we also model the distribution variance of structural data via partitioning all the nodes according to the homophily degree of nodes and then measure the statistics of deep features, which would imply the perturbation direction with structure semantics embedded. We also build benchmark datasets, including social datasets and biological datasets, to evaluate the domain generalization capacity on graphs. We perform extensive experiments, and the results validate the effectiveness of our SPOT compared with competing baselines. Our main contributions are summarized as follows:

— *Problem and Perspective*: We study the problem of domain generalization on graphs that aims to learn a GNN model with high generalization capability to unseen target graphs using various source graphs, and build the connections between graph spectrum with the model's domain generalization ability.
— *Novel Methodologies*: We propose a novel method named SPOT which not only formulates an optimal transport problem for the key structural augmentation but also models node distribution variance with the consideration of node homophily.
— *Empirical Validation*: We build both social and biological benchmarks for the evaluation of domain generalization on graphs, and extensive experiments validate the effectiveness of our SPOT compared with various state-of-the-art baselines.

2 Preliminaries

Problem Definition. In our problem, we are given M source graphs $\{G^1, \cdots, G^M\}$, and each source graph is expressed as $G^m = \{V^m, E^m\}$ where V^m denotes the node set with size n^m and $E^m \subset V^m \times V^m$ denotes the edge set. $\boldsymbol{X}^m \in \mathbb{R}^{n^m \times F}$ is the node feature matrix where F is the attribute dimension. Y^m denotes the label matrix where $y^m_{ik} = 1$ if node i belongs to the kth category and $y^m_{ik} = 0$ otherwise. We train a model using source graphs and evaluate our model on an unknown target graph $G^t = \{V^t, E^t\}$ with potential domain shifts.

Graph Spectrum. Given the graph $G = (V, E)$, adjacent matrix $\boldsymbol{A}$ and degree matrix $\boldsymbol{D}$, $\hat{\boldsymbol{L}} = \boldsymbol{I}_n - \hat{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{D}^{-\frac{1}{2}}$ represents the symmetric normalized graph Laplacian [18, 57], which can be decomposed into $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where $\boldsymbol{U} = \left[\boldsymbol{u}_1^\top, \ldots, \boldsymbol{u}_n^\top\right] \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\Lambda} = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$ are

composed of eigenvectors and eigenvalues, respectively. Without loss of generality, we assume $0 \leq \lambda_1 \leq \cdots \leq \lambda_N < 2$ [41] and then the graph Laplacian can be viewed as a signal filter. In particular, the graph spectrum [63] denotes these amplitudes of frequency components. For example, the graph spectrum of the graph Laplacian is naturally $g(\lambda_i) = \lambda_i$, and message passing with the normalized adjacent matrix corresponds to the graph spectrum $g(\lambda_i) = 1 - \lambda_i$.

GNNs. Every example is linked by abundant edges in graphs, which is different from i.i.d. image and text data. As an effective tool to encode graph-structured data, GNNs [22, 41, 75, 77, 94, 106, 107] are typically formulated as $\mathbf{H}^{(l+1)} = f\left(\mathbf{A}, \mathbf{H}^{(l)}; \mathbf{W}^{(l)}\right)$ where $\mathbf{A}$, $\mathbf{H}^{(l)}$ and $\mathbf{W}^{(l)}$ denote the adjacent matrix, the latent embedding matrix at the lth layer and the parameter weights at the lth layer, respectively. These methods typically adopt the paradigm of message passing, where each node attains knowledge from its neighbors along the edges to update the node representation. The representative GCN [41] is formulated as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}\mathbf{H}^{(l)}\mathbf{W}^{(k)}\right), \tag{1}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacent matrix with self-loop, $\widetilde{\mathbf{D}}$ is its degree matrix for normalization and $\sigma(\cdot)$ is an activation function. In comparison to convolution neural networks, structural relation is sufficiently utilized through the feature propagation operation along the edges to boost the performance in semi-supervised settings.

3 Problem Insights

Domain variances would bring both structural shifts and nodal shifts, and we can find clues in computer vision [61] to tackle the latter. Therefore, the primary challenge of our problem is to model the structural shifts. Here, we look into the relationships between the graph spectrum and generalization ability. In detail, we choose four classic graph augmentation strategies, i.e., PageRank [120], **Personalized PageRank (PPR)** [25], Edge Perturbation [100], and Heat Diffusion [31] and measure the change of the amplitude under augmentations for various frequencies by matrix perturbation theory. In formulation, the eigenvalue change, i.e., $\Delta\lambda_i = \lambda_i' - \lambda_i$ under augmentations can be deduced using matrix perturbation theory [70]. On this basis, we plot the graph spectra of the augmented adjacent matrices in Figure 1.

According to the findings, we can observe that PPR and Heat Diffusion tend to give more modification in the high-frequency regions, while Edge Perturbation and PageRank do so in the low-frequency regions. In addition, we explore the generalization ability under these augmentation strategies by training a GNN model in a source graph and testing it in a different target graph with potential domain shifts. From the compared results in Figure 1, we observe that there is a relation between the perturbation in the graph spectrum and the generalization capability of the model adopting the augmentation. Specifically, we can observe that perturbing the high-frequency regions of the graph spectrum during model training is beneficial to the model's generalization capability in the task of domain generalization. This is demonstrated by the higher performance of PPR and Heat Diffusion augmentations. This observation is also consistent with the fact that in computer vision [59], the low-frequency phase is more closely connected to semantic information while the high-frequency phase tends to be redundant noise. Based on this observation, we design the spectrum-aware structural augmentation that augments the high-frequency regions in the graph spectrum and uses optimal transport to obtain the augmentations.

Additionally, we argue that a data-centric solution is better than the model-centric alternative of enforcing low-pass filters. First, stacking low-pass filters has the risk of over-smoothing [38]. Second, graphs contain both low-frequency and high-frequency signals. The proposed spectrum-aware structural augmentation allows the model to train on graphs with different high-frequency

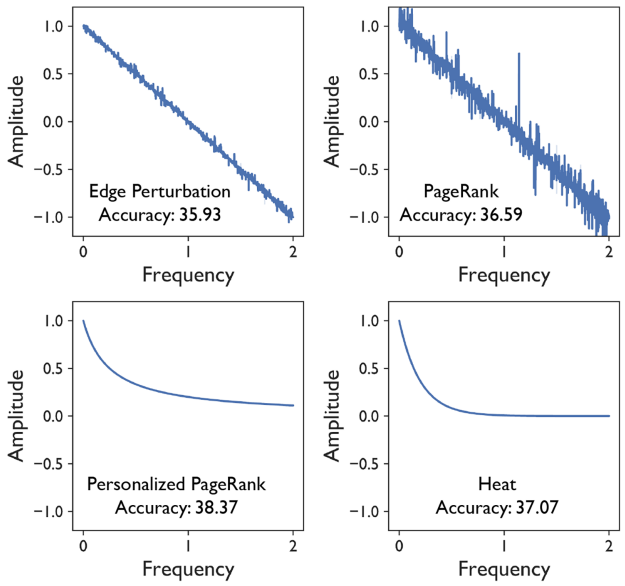

Fig. 1. The graph spectra and generalized performance of four graph augmentation strategies.

spectra, improving the robustness against domain shifts. By comparison, enforcing low-pass filters directly discards the high-frequency signals, leading to information loss. Third, stacked graph convolution layers like GCN [41] can be viewed as low-pass filters [62], and we show in Section 5.2 that our method outperforms these baselines.

4 Methodology

4.1 Overview

We propose a novel method named SPOT for effective domain generalization on graphs. Potential domain shifts on graphs consist of structural variances on connectivity and nodal variances on attributes. Therefore, the primary task of our SPOT is to explore the graph spectrum to simulate the structural domain variance with an augmentation operator O^S. In particular, we solve an optimal transport problem using Sinkhorn-Knopp algorithm to generate the fictitious structure with semantics incorporated. In addition, we take the homophily degree into account to derive a feature augmentation operator O^F to maintain relational semantic properties. With two augmentation operators, we decompose the GNN network into $f_\psi^2 \circ f_\theta^1$ where f_θ^1 denotes the shallow layer to generate node features and f_ψ^2 outputs the final prediction. The loss function is written as:

$$\mathcal{L} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n^m}\sum_{i\in G^m}\ell\left(f^2(O^F(f^1(\boldsymbol{x}_i)), O^S(\boldsymbol{A}^s)), y_i\right), \tag{2}$$

where $\ell(\cdot,\cdot)$ calculates the cross-entropy loss. An overview of our SPOT is illustrated in Figure 2. Then, we elaborate on our two key augmentation operators in our method.

4.2 Spectrum-Aware Structural Augmentation

The major challenge of domain generalization is to model structural domain variance, which has a huge difference from traditional domain generalization methods on i.i.d. data [84]. According to our discovery in Section 3, it is preferable to modify high-frequency regions rather than low-frequency

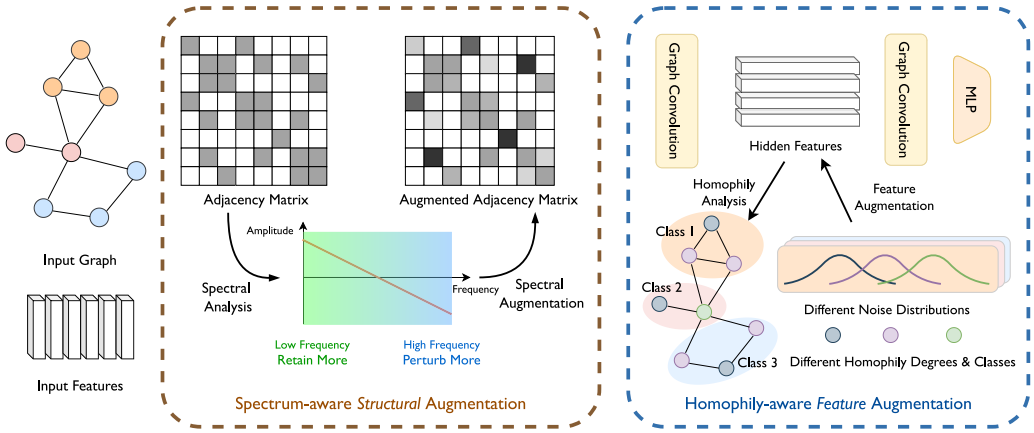

Fig. 2. The schematic of the proposed framework SPOT. On the one hand, we look into the graph spectrum and formulate an optimal transport problem to generate an augmented structure with more high-frequency signals filtered. On the other hand, we separate nodes based on their homophily degrees and generate augmented node features with structural semantics embedded.

regions for domain generalization. Toward this end, we formulate an optimal transport problem for the structural augmentation, which would allow low-frequency semantics knowledge in the graph spectrum [57, 64] to be retained from a global perspective.

In particular, we first decompose the variation of the adjacent matrix into two positive matrices, i.e., $\dot{\boldsymbol{A}}_+ - \dot{\boldsymbol{A}}_-$. To generate meaningful structures, we introduce a transportation polytope (i.e., a polytope's configuration matrix) for $\dot{\boldsymbol{A}}_+$ and $\dot{\boldsymbol{A}}_-$. Taking $\dot{\boldsymbol{A}}_+$ as an example with the omitted subscript, the transport polytope [2, 6] is formulated as:

$$\mathcal{U} = \left\{ \dot{\boldsymbol{A}} \in \mathbb{R}^{n \times n} \mid \dot{\boldsymbol{A}} \mathbf{1}_n = \boldsymbol{e}, \dot{\boldsymbol{A}}^\top \mathbf{1}_n = \boldsymbol{e} \right\}, \tag{3}$$

where $\boldsymbol{e}$ is the normalized degree vectors. $\mathcal{U}$ can be illustrated as a set of all plausible meaningful perturbations for $\boldsymbol{A}$, which enforces that the perturbed edge numbers for each node are balanced by proportionating to its degree. Then, to keep the low-frequency regions, we maximize the similarity between our target matrix $\dot{\boldsymbol{A}}$ and a high-pass filter matrix $\boldsymbol{P}$. In formulation, the optimal transport problem is written as:

$$\dot{\boldsymbol{A}}^* = \arg\max_{\dot{\boldsymbol{A}} \in \mathcal{U}} \alpha \sigma \left(Tr(\dot{\boldsymbol{A}}^T \boldsymbol{P} / \|\boldsymbol{P}\|_2) \right) + \epsilon H \left(\dot{\boldsymbol{A}} \right), \tag{4}$$

where $\sigma(\cdot)$ denotes an activation function, $Tr(\cdot)$ denotes the traces of the matrix, $||\boldsymbol{P}||_2$ is to normalize the prior matrix $\boldsymbol{P}$. $H(\boldsymbol{P}) = \sum_{i,j} \boldsymbol{P}_{i,j}(1 - \log(\boldsymbol{P}_{i,j}))$ measures the matrix entropy, which serves as a regularization for diverse edges. Note that the normalized graph Laplacian naturally filters more low-frequency signals, we utilize $\boldsymbol{P} = \hat{\boldsymbol{L}}$ for simplicity. However, the solution of Equation (4) is difficult to deduce, and thus, we convert the constraint of the transport polytope into soft penalties using the Lagrange multiplier method. The objective $\mathcal{T}$ can be written as:

$$\arg\max_{\dot{\boldsymbol{A}} \in \mathcal{R}^+} \mathcal{T} = \arg\max_{\dot{\boldsymbol{A}} \in \mathcal{R}^+} \alpha \sigma \left(Tr(\dot{\boldsymbol{A}}^T \boldsymbol{P} / \|\boldsymbol{P}\|_2) \right) + \epsilon H(\dot{\boldsymbol{A}}) + \left\langle \boldsymbol{f}, \dot{\boldsymbol{A}} \mathbf{1}_n - \boldsymbol{e} \right\rangle + \left\langle \boldsymbol{g}, \dot{\boldsymbol{A}}^\top \mathbf{1}_n - \boldsymbol{e} \right\rangle, \tag{5}$$

where $\boldsymbol{f}, \boldsymbol{g} \in \mathbb{R}^{n \times 1}$ denotes the Lagrange multipliers to regularize $\dot{\boldsymbol{A}}$ for being a polytope. $\langle \cdot, \cdot \rangle$ measures the product of two vectors. To maximize the objective, we calculate the partial of $\mathcal{T}$ with

Algorithm 1: Sinkhorn-Knopp Algorithm

Input: The prior matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$; The normalized degree matrix $\boldsymbol{e}$;
Output: The target matrix $\dot{\boldsymbol{A}}$;

1: Initialize $\dot{\boldsymbol{A}} \in (0,1)^{n\times n}$;
2: **while** not convergence **do**
3: $\quad \dot{\boldsymbol{A}} \leftarrow exp(\alpha/||\boldsymbol{P}||_2^2 \sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}))(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}})))\boldsymbol{P})$.
4: $\quad$ Row renomalization of $\dot{\boldsymbol{A}}$ to ensure $\dot{\boldsymbol{A}}\mathbf{1}_n = \boldsymbol{e}$.
5: $\quad$ Column renormlization of $\dot{\boldsymbol{A}}$ to ensure $\dot{\boldsymbol{A}}^T\mathbf{1}_n = \boldsymbol{e}$.
6: **end while**

respect to $\dot{\boldsymbol{A}}_{ij}$ as:

$$\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{A}}_{ij}} = \frac{\alpha}{||\boldsymbol{P}||_2^2}\sigma(Tr(\dot{\boldsymbol{A}}^T\boldsymbol{P}))(1-\sigma(Tr(\dot{\boldsymbol{A}}^T\boldsymbol{P}_{ij})))\boldsymbol{P}_{ij} - \epsilon \log \dot{\boldsymbol{A}}_{ij} + \boldsymbol{f}_i + \boldsymbol{g}_j. \tag{6}$$

On the basis of [57], the existence of the solution to Equation (5) can be promised using the following proposition with proper multipliers.

Proposition 1. *When* $\frac{\alpha \boldsymbol{P}_{ij}}{4||\boldsymbol{P}||_2^2} + \boldsymbol{f}_i + \boldsymbol{g}_j < 0$, $\mathcal{T}$ *can reach the maximal value for* $\dot{\boldsymbol{A}}_{ij} \in (0,1)$.

Proof. Since

$$\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{A}}_{ij}} = \frac{\alpha}{||\boldsymbol{P}||_2^2}\sigma(Tr(\dot{\boldsymbol{A}}^T\boldsymbol{P}))(1-\sigma(Tr(\dot{\boldsymbol{A}}^T\boldsymbol{P}_{ij})))\boldsymbol{P}_{ij} - \epsilon \log \dot{\boldsymbol{A}}_{ij} + \boldsymbol{f}_i + \boldsymbol{g}_j, \tag{7}$$

we have $\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{A}}_{ij}}|_{\dot{\boldsymbol{A}}=0} \to +\infty$ and $\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{A}}_{ij}}|_{\dot{\boldsymbol{A}}=1} \leq \frac{\alpha \boldsymbol{P}_{ij}}{4||\boldsymbol{P}||_2^2} + \boldsymbol{f}_i + \boldsymbol{g}_j < 0$. Note that $\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{A}}_{ij}}$ is continuous with respect to $\dot{\boldsymbol{A}}_{ij}$. According to the intermediate value theorem, $\exists \dot{\boldsymbol{A}}_{ij} \in (0,1)$, s.t. $\frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{A}}_{ij}} = 0$. Hence, $\mathcal{T}$ can reach the maximal value. □

However, $\partial \mathcal{T}/\partial\left(\dot{\boldsymbol{A}}\right) = 0$ is hard to solve. Note that $\dot{\boldsymbol{A}}$ would be updated smoothly in our experiments, we turn to an iterative algorithm where the previous value of $\dot{\boldsymbol{A}}$ is utilized to obtain the current value. By incorporating these current values, we have:

$$\alpha/||\boldsymbol{P}||_2^2 \sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}'))(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}')))\boldsymbol{P}_{ij} = \epsilon \log \dot{\boldsymbol{A}}_{ij} + \boldsymbol{f}_i + \boldsymbol{g}_j, \tag{8}$$

where $\dot{\boldsymbol{A}}'$ denotes the value at the previous epoch. Hence, the updating rule for $\dot{\boldsymbol{A}}$ is written as:

$$\dot{\boldsymbol{A}}_{ij} \leftarrow \operatorname{diag}\left(\exp\left(\frac{\boldsymbol{f}_i}{\epsilon}\right)\right)\exp\left(\frac{\alpha}{||\boldsymbol{P}||_2^2}\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}))\left(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}))\right)\boldsymbol{P}_{ij}\right)\operatorname{diag}\left(\exp\left(\frac{\boldsymbol{g}_j}{\epsilon}\right)\right). \tag{9}$$

Recall the restriction of the transport polytope, i.e., $\mathbf{1}_n = \boldsymbol{e}$, $\dot{\boldsymbol{A}}^\top \mathbf{1}_n = \boldsymbol{e}$. We solve the optimal transport problem using the Sinkhorn-Knopp algorithm [14], which conducts iterative row- and column-normalization on $\exp(\frac{\alpha}{||\boldsymbol{P}||_2^2}\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}))(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}})))\boldsymbol{P}_{ij})$ instead of calculating $\boldsymbol{f}$ and $\boldsymbol{g}$. The Sinkhorn-Knopp algorithm is summarized in Algorithm 1.

Based on the algorithm and [57], we can show the approximate difference between $\dot{\boldsymbol{A}}$ and $\dot{\boldsymbol{A}}'$ is bounded.

Proposition 2. *We have* $|\zeta \dot{\boldsymbol{A}}_{ij} - \dot{\boldsymbol{A}}'_{ij}| < \rho + \delta$ *where* $\delta = \frac{2\boldsymbol{P}_{ij} + \sum_{i'\neq i, j'\neq j} \boldsymbol{P}^2_{i'j'}\dot{\boldsymbol{A}}'_{i'j'}}{\boldsymbol{P}^2_{ij}} + \max_{\dot{\boldsymbol{A}}'}(1 - \sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}')))$ *and* $\rho = \frac{\zeta}{1-\gamma}\{d(\boldsymbol{r}, \boldsymbol{e}) + d(\boldsymbol{c}, \boldsymbol{e})\}$. $\boldsymbol{r}$ *and* $\boldsymbol{c}$ *are the initial row and column vectors for normalization in Algorithm 1.* γ *is the contraction ratio as in [65].*

Proof. First, we recall the convergence of Sinkhorn-Knopp algorithm, which gives the following inequation:

$$\left\|\log \dot{\boldsymbol{A}}^{l} - \log \dot{\boldsymbol{A}}\right\|_{\infty} \leq \frac{1}{1-\gamma}\left\{d\left(\boldsymbol{r}^{l}, \boldsymbol{e}\right) + d\left(\boldsymbol{c}^{l}, \boldsymbol{e}\right)\right\}, \tag{10}$$

where $\dot{\boldsymbol{A}}^{l}$ and $\dot{\boldsymbol{A}}$ refer to the results at lth iteration and the final value and γ is a fixed coefficient. $\boldsymbol{r}^{l}$ and $\boldsymbol{c}^{l}$ refer to the row and column vectors for normalization at lth iteration. The proof of Equation (10) can be directly achieved from [65]. Let $l = 0$ and we have:

$$\left|\frac{\alpha}{\epsilon||\boldsymbol{P}||_2^2}\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}'))(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}')))\boldsymbol{P}_{ij} - \log \dot{\boldsymbol{A}}_{ij}\right| \leq \frac{1}{1-\gamma}\left\{d\left(\boldsymbol{r}, \boldsymbol{e}\right) + d\left(\boldsymbol{c}, \boldsymbol{e}\right)\right\}, \tag{11}$$

$$\left|\frac{\alpha}{\epsilon||\boldsymbol{P}||_2^2}\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}'))(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}')))\boldsymbol{P}_{ij} - \dot{\boldsymbol{A}}_{ij}\right| - 1 \leq \frac{1}{1-\gamma}\left\{d\left(\boldsymbol{r}, \boldsymbol{e}\right) + d\left(\boldsymbol{c}, \boldsymbol{e}\right)\right\}. \tag{12}$$

Let $\Pi = \max_{\dot{\boldsymbol{A}}'}(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}')))$, we have

$$\left|\frac{\Pi\alpha}{4\epsilon||\boldsymbol{P}||_2^2}(2 + Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}'))\boldsymbol{P}_{ij} - \dot{\boldsymbol{A}}_{ij}\right| - 1 \leq \frac{1}{1-\gamma}\left\{d\left(\boldsymbol{r}, \boldsymbol{e}\right) + d\left(\boldsymbol{c}, \boldsymbol{e}\right)\right\}, \tag{13}$$

$$\left|\frac{\Pi\alpha}{4\epsilon||\boldsymbol{P}||_2^2}\left(2\boldsymbol{P}_{ij} + \sum_{i'\neq i, j'\neq j}\boldsymbol{P}_{i'j'}^2\dot{\boldsymbol{A}}'_{i'j'} + \boldsymbol{P}_{ij}^2\dot{\boldsymbol{A}}'_{ij}\right) - \dot{\boldsymbol{A}}_{ij}\right| - 1 \leq \frac{1}{1-\gamma}\left\{d\left(\boldsymbol{r}, \boldsymbol{e}\right) + d\left(\boldsymbol{c}, \boldsymbol{e}\right)\right\}. \tag{14}$$

Let $\zeta = \frac{4\epsilon||\boldsymbol{P}||_2^2}{\Pi\alpha\boldsymbol{P}_{ij}^2}$, and we can obtain:

$$|\zeta\dot{\boldsymbol{A}}_{ij} - \dot{\boldsymbol{A}}'_{ij}| < \frac{\zeta}{1-\gamma}\left\{d\left(\boldsymbol{r}, \boldsymbol{e}\right) + d\left(\boldsymbol{c}, \boldsymbol{e}\right)\right\} + \gamma + \frac{2\boldsymbol{P}_{ij} + \sum_{i'\neq i, j'\neq j}\boldsymbol{P}_{i'j'}^2\dot{\boldsymbol{A}}'_{i'j'}}{\boldsymbol{P}_{ij}^2}. \tag{15}$$

□

Similarly, we derive the optimal $\dot{\boldsymbol{A}}_{-}^{*}$ by:

$$(\dot{\boldsymbol{A}}_{-})_{ij} \leftarrow \operatorname{diag}\left(\exp\left(\frac{\boldsymbol{f}_i}{\epsilon}\right)\right)\exp\left(-\frac{\alpha}{||\boldsymbol{P}||_2^2}\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}_{-}))(1-\sigma(Tr(\boldsymbol{P}, \dot{\boldsymbol{A}}_{-})))\boldsymbol{P}_{ij}\right)\operatorname{diag}\left(\exp\left(\frac{\boldsymbol{g}_j}{\epsilon}\right)\right). \tag{16}$$

To maintain the sparsity of the derived adjacent matrix, we restrict the perturbation to the neighborhood around each node. The augmented adjacent matrix is written as:

$$\hat{\boldsymbol{A}} = \mathcal{O}^{s}(\boldsymbol{A}) = \boldsymbol{A} + \mathbf{1}_{\boldsymbol{A}>0} \odot (\dot{\boldsymbol{A}}^{*} - \dot{\boldsymbol{A}}_{-}^{*}), \tag{17}$$

where $\odot$ is the Hadamard product of two matrices. In addition, to maintain the diversity for augmented graphs, we vary $\alpha \in (0, \alpha_{max})$ randomly in Equation (5).

4.3 Homophily-Aware Feature Augmentation

Feature perturbation has been a frequently used tool to boost the model generalization ability since deep features are often fully linearized [53, 79, 118]. However, this technique usually deals with i.i.d data rather than relational data on graphs. The relational semantics would change if all nodes are treated independently. To address this, we quantify node homophily degrees and modify node embeddings in the feature space in data-dependent directions.

In detail, we first measure the homophily degree, which is crucial for comprehending the structural information in the graph. Here, the homophily degree of node i is defined as:

$$s_i = \frac{|\{j|y_i = y_j, j \in N(i)\}|}{|\{j|j \in N(i)\}|}, \tag{18}$$

where $|\cdot|$ calculates the number of elements in the set. A high s_i implies more semantically similar neighboring nodes around node i. Then, we divide all nodes into K bins, $[0, t_1], \cdots, [t_{K-1}, 1]$ according to these homophily degrees where K is a predefined parameter. The nodes in the same bin share similar homophily degrees, which could exhibit similar properties under distribution shifts. To maintain the structural information, we utilize a Gaussian mixture model to measure the distribution of deep features in each bin. In particular, a normal distribution $\mathcal{N}(\boldsymbol{\mu}_c^k, \Sigma_c^k)$ is used to characterize the distribution for each class c in the kth bin. Then, we augment deep features with semantics statistics preserved. In formulation, the augmented features are written as:

$$\hat{\boldsymbol{z}}_i = O^F(\boldsymbol{z}_i) = \boldsymbol{z}_i + \eta \boldsymbol{\beta}_i, \tag{19}$$

where $\boldsymbol{\beta}_i$ is sampled from $\mathcal{N}(\mathbf{0}, \Sigma_c^k)$ and η is a coefficient to determines the strength of perturbation. Here Σ_c^k is a covariance matrix estimated from deep features from the corresponding bin. Compared with i.i.d. feature augmentations, our strategy would follow a skew distribution to keep the semantics and crucial structural relations in the graphs. In particular, deep features with lower homophily degrees tend to show large variances in their distribution and our homophily-aware feature augmentation keeps their structural property. The homophily-aware feature augmentation and spectrum-aware structural augmentation are performed when training on the source domains. During testing, the target graphs are not augmented.

4.4 Compared with Existing Works

Finally, we discuss the differences and relations between SPOT and existing works.

Relation to Graph Contrastive Learning. Among various graph contrastive learning methods [46, 54, 72, 92, 100], SpCo [57] and our method both look into graph spectrum but in different scenarios. Our method utilizes two types of augmentations to produce proper fictitious data while graph contrastive learning attempts to maximize the mutual information between two augmented views. In particular, SpCo maximizes the amplitudes of high-frequency signals to generate different views. In contrast, we are the first to study the impact of graph spectrum on out-of-domain generalization and develop an optimal transport problem for the structural augmentation.

Relation to Adversarial Augmentation. Several works in computer vision follow the worst-case problem [79, 86, 118] to utilize adversarial learning for data augmentation, which generates challenging samples to learn a model robust to domain variance. For example, M-ADA [66] utilizes a meta-learning scheme while L2D [86] turns to a style-aware variational module. These methods and our method all construct desirable augmentations for out-of-domain generalization but focus on different data types. Specifically, these methods almost address i.i.d. image data while our method focuses on relational graph data and presents spectrum-aware structural augmentation and homophily-aware feature augmentation for our problem.

4.5 Time Complexity Analysis

In this subsection, we provide a time complexity analysis of the proposed SPOT. For simplicity, we consider one source domain (one graph) with $|V|$ nodes and $|E|$ edges in a single epoch. For spectrum-aware structural augmentation, the Sinkhorn-Knopp algorithm (Algorithm 1) has the overall complexity of $O(T_{sk}|V|^2)$, where T_{sk} is the number of iterations in the Sinkhorn-Knopp algorithm. In practice, this algorithm converges within several iterations [6], and T_{sk} can be viewed

as a constant, leading to the overall complexity of $O(|V|^2)$. Furthermore, modern research in optimal transport [52] has shown that the Sinkhorn-Knopp algorithm can be further optimized to have complexity of $\tilde{O}(|V|)$. For homophily-aware feature augmentation, the computation of homophily degree is $O(|E|)$ and the augmentation is $O(|V|)$. The overall time complexity is $O(|V|^2)$. For sparse graphs, the algorithm can be further optimized to have complexity of $\tilde{O}(|V|)$.

5 Experiments

In this section, experiments are conducted to demonstrate the effectiveness of the proposed SPOT. Furthermore, a comprehensive analysis is provided on the performance of the model and the role of spectrum-aware structural augmentation and homophily-aware feature augmentation.

5.1 Experimental Setup

Datasets. In the experiments, we construct two benchmarks with real-world data: the Citation benchmark and the Proteins benchmark.

The Citation benchmark contains three real-world citation networks [74] including ACMv9, Citationv1, and DBLPv7. The citation networks contain papers published in different time ranges and collected by different sources. More specifically, ACMv9 network is constructed by ACM and contains papers published after 2010. Citationv1 network is constructed by Microsoft Academic Graph and contains papers published before 2008. DBLPv7 network is constructed by DBLP and contains papers published between 2004 and 2008. Each node in the network corresponds to a paper, and the edges represent citations. Note that we consider the network as an undirected graph. As for the node attributes, we construct bag-of-words vectors using the keywords from the paper titles. Each node (paper) belongs to *some* of the five categories ("Networking," "Information Security," "Computer Vision," "Artificial Intelligence," and "Databases"), and thus, we use F1 scores to measure the classification performance.

The Proteins benchmark contains four proteins which consist of single chain. The sequence identity between these four proteins is less than 30%. We fetched their experimentally resolved structures from PDB database (PDB id: 1S5J, 6BBM, 6N8P and 3UJZ) [3], and the secondary structure types of each residue were computed by DSSP [34] as the target for prediction. Graph input for each protein was constructed, and every residue is represented by a node whose attribute comprises ESM-1b protein language embedding [67] and the residue type. Then, an edge between two nodes was added when the distance between the C_α atoms is below 10 Å.

The Open Graph Benchmark (OGB)-Products benchmark [13] is a product co-purchasing network collected from Amazon. We use the version from the OGB [32]. The nodes in the graph depict items, while the edges imply that the two items are purchased together. We adopt this benchmark in the OOD generalization setting where we split the graph into two parts according to the degrees of the nodes and use one part as in-distribution data for training, the other as OOD for testing.

Evaluation Setting and Metrics. For both benchmarks, we adopt the leave-one-domain-out evaluation protocol in alignment with previous works [20, 95]. For each task, we select one domain as the test domain and train the model on all other domains. For the Citation benchmark, we use A, C, and D to denote ACMv9, Citationv1, and DBLPv7. Thus, we can construct three tasks: AC $\rightarrow$ D, AD $\rightarrow$ C, and CD $\rightarrow$ A. Similarly, for the Proteins dataset, we leave one network out and construct four tasks, namely P1, P2, P3, and P4. As for the evaluation metrics, we use micro f1 and macro F1 for citation networks and accuracy for protein networks.

Baseline Methods. We compare our model with a wealth of baselines, including GCN [41], GIN [94], GAT [78], SGC [88], EGC [73], ADA [79], MAT [84], and FLOOD [58]. Their details are listed as follows:

— *GCN* [41] which uses normalized degrees as weights to aggregate neighboring node features and achieves promising generalization capability.
— *GIN* [94] which is proposed to enhance the representation power of GNNs.
— *GAT* [78] which uses the attention mechanism to compute the weights for aggregating adjacent node features.
— *SGC* [88] which simplifies traditional GNNs by removing the non-linearity between the convolution operators.
— *EGC* [73] which proposes an isotropic GNN and achieves promising results with lower memory consumption.
— *ADA* [79] which considers a worst-case formulation and adopts adversarial examples to enhance the model's generalization ability.
— *MAT* [84] which utilizes universal adversarial training to generate robust large-scale perturbations that better help with the model's generalization capability.
— *FLOOD* [58] which focuses on improving the generalization of models to OOD data by leveraging the principle of invariance. By excluding the test-time-training objective, FLOOD aligns more closely with the needs of multi-domain generalization settings, where the goal is to ensure consistent performance across various domains without the necessity for domain-specific tuning.
— *MARIO* [119] which adopts the information bottleneck principle and the invariant principle. MARIO improves graph contrastive learning in the face of OOD samples, which leads to better results.

The baselines cover GNNs initially developed for single domains (GCN, GIN, GAT, SGC, and EGC), domain generalization methods for i.i.d. data like images (ADA and MAT), and domain generalization methods for OOD data (FLOOD).

Implementation Details. We implement the proposed SPOT with PyTorch and train the model with an NVIDIA RTX GPU. In Equations (4) and (5), we set α to 6.0 and ϵ to 0.05. In Equation (19), we set η to 0.6, and Gaussian noise is also added to the feature. The number of bins in homophily-aware feature augmentation (i.e., K) is set to 3. For optimization, Adam optimizer [40] is used and the model is trained for 200 epochs with the learning rate of 0.003. For graph convolution, we use GCN [41] as default. The proposed model contains two graph convolution modules. In the middle of the two graph convolutions, we use a ReLU as non-linearity and we also use a dropout layer (with the dropout probability of 0.1) to avoid overfitting. After ReLU and dropout, the features are augmented using homophily-aware feature augmentation. During inference, neither spectrum-aware structural augmentation nor homophily-aware feature augmentation is used in order for a stable and accurate classification result.

5.2 The Performance of SPOT

The performance of SPOT in comparison with baseline methods is listed in Table 1 (the Citation benchmark) and Table 2 (the Proteins benchmark and the OGB-Products benchmark). According to the results, we have two observations described as follows.

First, the proposed SPOT achieves the best performance compared to various baseline methods in both Citation and Proteins benchmarks. The significant improvement can be attributed to two aspects: (1) The model leverages spectrum-aware structural augmentation to generate semantic-preserving structures. This augmentation retains the key structures that generalize across domains, while augmenting the high-frequent information that is more domain-specific. (2) The model adopts homophily-aware feature augmentation that perturbs node attributes according to their homophily

Table 1. Performance Comparison on the Citation Benchmark

Methods	AC → D		AD → C		CD → A	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
GCN [41]	66.45 ± 0.66	61.49 ± 0.60	62.96 ± 0.16	59.26 ± 0.67	63.04 ± 0.61	61.04 ± 1.51
GIN [94]	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 [78]	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 [88]	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 [73]	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 [79]	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 [84]	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 [58]	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 [119]	66.94 ± 0.18	61.84 ± 0.26	65.81 ± 1.50	60.75 ± 1.05	63.76 ± 0.39	58.76 ± 2.67
SPOT (ours)	**70.76 ± 0.28**	**66.20 ± 0.42**	**69.68 ± 0.13**	**65.57 ± 0.78**	**67.52 ± 0.30**	**66.38 ± 0.26**

The best results are marked in bold.

Table 2. Comparison of Prediction Accuracy on Both Proteins and OGB-Products Benchmarks

Methods	Proteins					OGB-Products
	P1	P2	P3	P4	Average	
GCN	49.24	44.62	34.18	47.97	44.00	35.80
GIN	42.92	50.79	**41.23**	44.88	44.96	24.28
GAT	26.69	51.03	37.97	33.01	37.18	31.92
SGC	42.37	46.19	32.19	37.56	39.58	35.96
EGC	49.93	40.75	31.83	41.14	40.91	9.09
ADA	49.11	44.14	34.18	48.29	43.93	35.97
MAT	48.97	44.14	34.18	48.13	43.86	36.04
FLOOD	50.48	52.60	34.18	37.25	44.39	36.22
MARIO	49.70	52.92	38.82	45.04	46.62	37.06
SPOT	**53.92**	**52.60**	38.70	**48.62**	**48.46**	**38.23**

The best results are marked in bold.

degree. This helps the model overcome distribution variances in node attributes and provides better augmentation for domain generalization on graphs.

Moreover, we observe that domain generalization methods proposed for i.i.d. data (e.g., images) fail to achieve satisfactory accuracy in graphical data where the independence of data is not guaranteed. Most previous methods requiring the independence assumption (e.g., ADA and MAT) show little performance gain compared to their GNN baseline (i.e., GCN). This demonstrates the existence and importance of structural shifts in graph domain generalization, which drives us to design structural augmentations that preserve domain-invariant structures. On the other hand, methods like ADA and MAT also make the assumption that instances are identically distributed, which is often violated in graphical data. This could be a possible reason for their inferior performance, and in comparison, we utilize the homophily information in the network for better perturbation.

Additionally, the proposed SPOT generalizes well under distribution shifts, as is shown by the results on the OGB-Products dataset in Table 2. This shows that the proposed method can also be applied to the common OOD generalization setting, where the source data and target data are usually on the same graph.

Table 3. Ablation Studies on the Citation Benchmark

Methods	AC → D		AD → C		CD → A	
	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1
w/o SAS Aug.	66.32	62.21	64.08	60.24	64.88	62.5
Edge Purterbation	66.33	62.58	64.07	60.35	65.04	63.01
w/o HAF Aug.	69.30	64.46	67.03	62.84	65.42	62.01
w/o Homophily	70.06	65.69	68.53	64.20	66.24	63.15
SPOT	70.76	66.20	69.68	65.57	67.52	66.38

5.3 Ablation Studies

In this subsection, ablation studies are conducted to demonstrate the effectiveness of the proposed spectrum-aware structural augmentation and homophily-aware feature augmentation. To this end, we evaluate the performance of four variants of SPOT:

—The model without spectrum-aware structural augmentation ("w/o SAS Aug.").
—The SPOT model but replacing the structural augmentation with edge perturbation ("Edge perturbation").
—The model without the entire homophily-aware feature augmentation module ("w/o HAF Aug.").
—The model without homophily awareness (without dividing nodes into K bins) in homophily-aware feature augmentation while preserving the feature augmentation ("w/o Homophily").

The experimental results are shown in Table 3. According to the results, we have several observations. First, both spectrum-aware structural augmentation and homophily-aware feature augmentation contribute to the model's performance. Removing each model (Lines 1 and 3 in Table 3) causes both micro F1 and macro F1 scores to drop. Relatively speaking, structural augmentation plays a more significant role than attribute augmentation, with a 4.44% decline in macro F1 compared to a 1.46% decline (AC → D). This suggests that the potential structural shift is more essential in graph domain generalization.

To further investigate the effect of spectrum-aware structural augmentation, we replace it with another random augmentation: edge perturbation. The replacing augmentation does not differentiate the magnitude of perturbation across the graph spectrum. As we can see from the result (Line 2 in Table 3), it provides little performance gain with respect to both micro F1 and macro F1. This shows that augmenting the graph structure uniformly does little help with the model's generalization capability and that by employing spectrum-aware structural augmentation, the proposed SPOT better preserves domain-invariant information while perturbing the domain-specific information.

To better understand the effect of Homophily-aware attribute augmentation, we remove the homophily awareness in the model. The result (Line 4 in Table 3) shows that without homophily awareness, the model's performance drops, which validates the effect of homophily-aware perturbations.

5.4 Hyperparameter Analysis

In this subsection, we explore the effect of hyperparameters on the Citation benchmark. Concretely, we focus on two key hyperparameters: (1) α in Equation (5), which controls the weight of the first term in the Lagrangian objective, and (2) η in Equation (19), which controls the magnitude of homophily-aware perturbation. The results are shown in Figure 3.

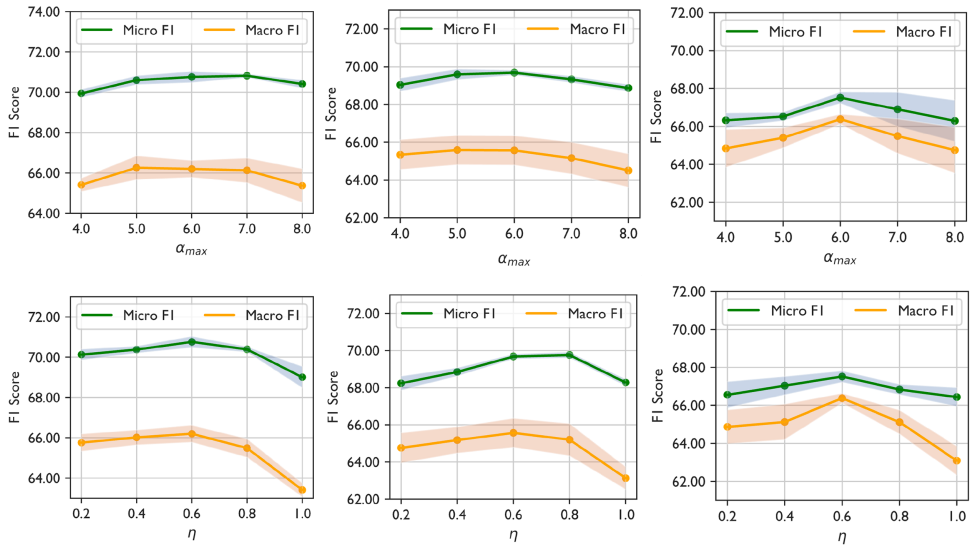

Fig. 3. The model's sensitivity to hyperparameters. The upper row shows the change in terms of micro F1 and macro F1 with different αs (in Equations (4) and (5)) on AC → D, AD → C, CD → A, respectively. The lower row presents the performance change with different ηs (in Equation (19)) on AC → D, AD → C, CD → A, respectively.

Generally, the proposed SPOT is not sensitive to these hyperparameters. For hyperparameter α, we observe that the model achieves the best performance when α is set to 6.0 or 7.0, which provides the appropriate weight for the first term (the closeness between the high-pass filter matrix and the perturbation). When α is too small, the entropy regularizer (second term in Equation (5)) gets relatively more weight and the perturbation gets more uniformly distributed, degrading the spectral property of the augmentation. On the other hand, a large α could diminish the regularization effect of the entropy, which results in sub-optimal augmentations.

As for the hyperparameter η, the best performance is achieved when it is set to 0.6, which provides the appropriate balance between homophily-aware perturbation and the original graph signal. When η is low, the perturbation is small which has worse performance compared to higher ηs. On the other side, when η grows too large, it has the potential to overwhelm the original graph signal, which hurts the performance.

5.5 Visualization of the Training Process

We also provide the visualization results of the training process. Concretely, we compare our model with GCN and draw the curves of the cross-entropy loss. More specifically, we conduct experiments on all three sub-tasks (i.e., AC → D, AD → C, and CD → A) of the Citation dataset. The results are shown in Figure 4, where the blue curves show the cross-entropy loss of the source domain and the orange curves denote the cross-entropy loss of the target domain. As can be seen from the figure, vanilla GCN easily overfits the source domain as the training proceeds after 50 epochs. The loss continues to drop in the source domain but grows in the target domain, which is a signal of overfitting the source data and limits the model's generalization capability. By contrast, the proposed SPOT avoids overfitting via efficient augmentations (i.e., spectrum-aware structural augmentation and homophily-aware feature augmentation), which helps with the model's generalization ability to the target domain.

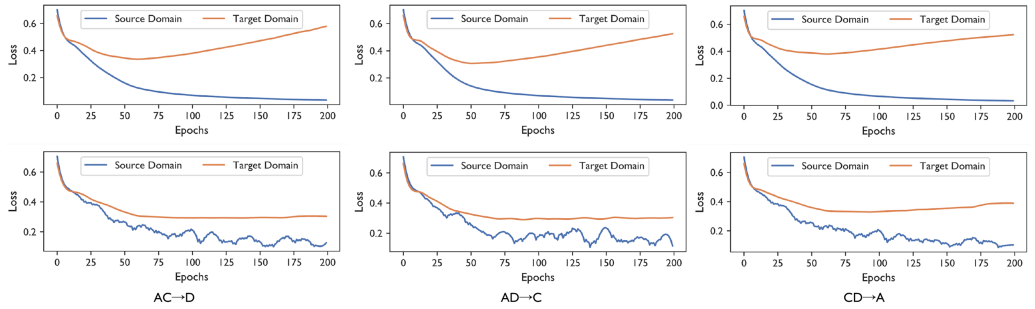

Fig. 4. The cross-entropy loss on the source domain and target domain. The upper row shows the result of GCN, while the bottom row presents the result of SPOT. The proposed model better avoids overfitting to the source domains (blue curves) and generalizes well to the target domain (orange curves).

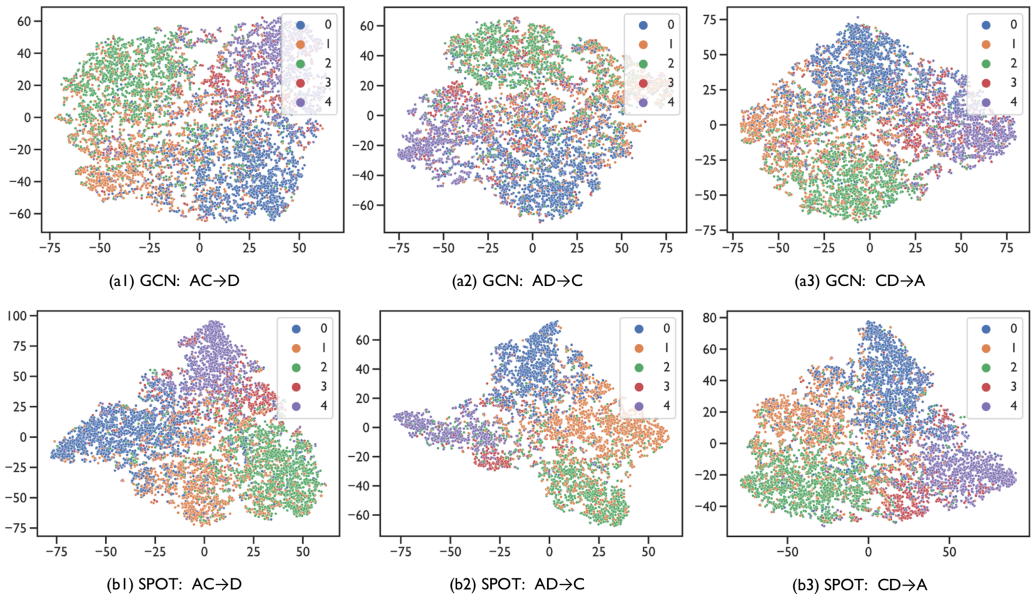

Fig. 5. Visualization of learned representations of each node in the Citation dataset using t-SNE. *Top*: the representations learned by GCN. *Bottom*: the representations learned by SPOT.

5.6 Visualization of Learned Representations

In this section, we provide visualization results of learned representations of each node in the graphs using t-SNE [76]. The results of GCN and the proposed SPOT are shown in Figure 5. The experiments are performed on all the sub-tasks of the Citation dataset. Specifically, AC → D is shown in the top row, sub-figure (a1) and (b1); AD → C is shown in the middle row, sub-figure (a2) and (b2); CD → A is shown in the bottom row, sub-figure (a3) and (b3). As can be seen from the results, the proposed SPOT learns a more condensed and clustered representation of the nodes in the graphs across all the sub-tasks. This suggests that the proposed method is able to transfer the learned knowledge from the source domains to the target domain. Moreover, the results also show that the proposed method tends to learn more separate representations in the feature space. For instance, in sub-figure (b1) and (b2), Classes 0 and 4 are more separate than the corresponding baseline in (a1) and (a2).

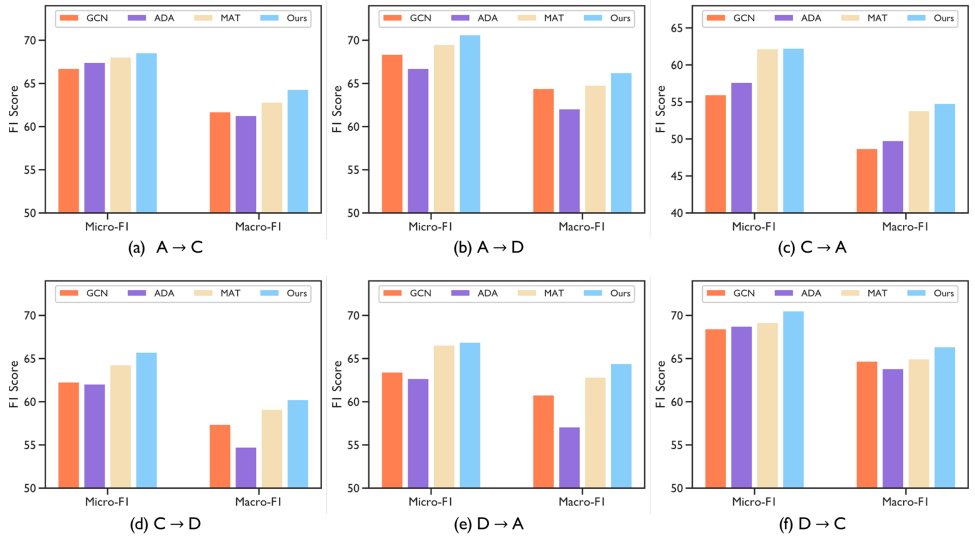

Fig. 6. The model's performance on the single-source domain generalization setting in comparison with the baselines. The experiments are performed on the Citation dataset (A: ACMv9, C: Citationv1, D: DBLPv7, labeled as source → target). The models are trained on a single domain and then tested on another domain.

We attribute this to the augmentations, both structural (spectrum-aware structural augmentation) and nodal (homophily-aware feature augmentation) that generate augmented views around the input sample in the feature space and thus help the model to learn separate representations that benefit the downstream classification task.

5.7 Performance for Single-Source Domain Generalization

We also verify the proposed SPOT's generalizability to the single-source domain generalization setting, where only a single domain is used for training the model. The experiments are performed on the Citation dataset, leading to six sub-tasks in total (i.e., A → C, A → D, C → A, C → D, D → A, and D → C). The results are shown in Figure 6. As presented in the figures, our method outperforms all the baseline methods in all sub-tasks, which shows that our method generalizes well on the single-source domain generalization setting. The proposed spectrum-aware structural augmentation and homophily-aware feature augmentation are able to learn generalizable graph features from a single domain. Moreover, when comparing with the performance with multi-domain generalization setting, we notice that some methods' performance on the multi-domain setting is worse than single-domain setting. For example, in A → C setting, GCN achieves over 65 Micro-F1 with only one source domain, whereas in the AD → C setting (in Table 1), GCN only has less than 63 Micro-F1 score. This suggests that the baseline models fail to fully utilize all information from different source domains. Compared to these methods, the proposed SPOT achieves better performance on average with an increasing number of source domains. This demonstrates that the proposed model can better utilize information from multiple sources and learn transferable representations across domains.

5.8 Efficiency Analysis

We also provide an efficiency analysis of the proposed SPOT in comparison with several baselines. Specifically, we report the training time and the GPU memory in Table 4. The two metrics are

Table 4. The Training Time and Memory Cost of the Proposed SPOT and Baselines

Methods	GCN	ADA	FLOOD	SPOT
Training time (s)	19.1	29.2	29.65	34.0
GPU memory (GB)	2.3	2.3	4.2	2.4

measured in the Citation dataset (AD → C setting). As can be seen from the results, the proposed method can be trained very efficiently, with comparable training time and GPU memory to other domain generalization or OOD generalization methods.

6 Related Works

6.1 Graph Machine Learning

With the advancement of various techniques in deep learning, GNNs have shown promising empirical performance in analyzing graph-structured data in recent years [35, 37, 50]. A range of GNN methods have been developed for various machine learning problems, such as node classification [41, 94], link prediction [4], and graph classification [12]. Most GNNs can be formulated into a message passing paradigm, with each node collecting propagated information from its adjacent neighbors and engaging in a recursive node representation update [94]. Despite their enormous success, existing methods usually neglect the potential domain shifts in real-world unknown graphs [36]. To tackle this issue, we investigate the problem of graph domain generalization, which attempts to learn a GNN model with superior generalization ability using several source graphs.

6.2 Domain Generalization

The target of domain generalization is to construct a model that can well generalize to unobserved target domains using merely source data [81, 83]. Different from domain adaptation [16, 28, 56, 60, 89] or inductive learning [29], domain generalization aims to make the model more robust against distribution shifts during training, without accessing the test input. Existing literature has offered various solutions to this problem. By reducing differences across distinct domains, early approaches often presuppose the presence of several source domains and attempt to learn domain-invariant representations. These techniques typically either reduce the explicit distribution difference [49, 85] or accomplish implicit domain alignment via adversarial learning [1, 33, 110]. Recently, data augmentation has been proven to be a powerful tool for domain generalization, which constructs fictitious and challenging target data to solve data scarcity [79, 113, 118].

To improve the generalization ability of GNNs, efforts have been made in graph domain generalization [9, 102] and graph OOD generalization [47, 48, 104, 119]. Comprehensive surveys [48, 104] have provided taxonomies of these approaches, including data-centric methods, model-centric methods, and special training strategies. While some of these methods also adopt a data-centric perspective [9, 119], this work tackles this problem from the less-explored graph spectrum perspective with solid theoretical foundations and proposes spectrum-aware structural augmentation in addition to homophily-aware feature augmentation.

6.3 Graph Modeling with Spectrum Learning

The spectra of graphs [42, 97] provide a new perspective for modeling and learning on graphs. Recently, efforts have been made in combining the graph spectra with GNNs [41], and methods have been proposed to enhance graph modeling with graph spectra in various tasks, including

contrastive learning [57], adversarial learning [10], heterophilic graph learning [19, 24, 93, 116], domain adaptation [64, 91, 101], OOD detection [26], and generalization [71, 109]. This article adopts the perspective of graph spectra for structural augmentation and reformulates the problem into optimal transport, which is solved using Sinkhorn-Knopp iterations.

7 Conclusion

This study investigates an understudied yet useful problem of domain generalization on graphs and proposes a novel solution dubbed SPOT. On the basis of our preliminary experiments, our SPOT formulates the structural augmentation as an optimum transport problem to maintain low-frequency important information. Furthermore, we implement an adaptive perturbation strategy on deep features with structural semantics retained, where the direction of additive noise is governed by homophily degrees. We build numerous benchmark datasets to evaluate the domain generalization capabilities of graphs, and comprehensive experiments validate the efficacy of our proposed SPOT.

Acknowledgment

The authors are grateful to the anonymous reviewers for their critical reading of this article and for giving important suggestions to improve this article.

References

[1] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H. Falk, and Ioannis Mitliagkas. 2019. Generalizing to unseen domains via distribution matching. arXiv:1911.00804. Retrieved from https://arxiv.org/abs/1911.00804

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *ICLR*.

[3] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. The protein data bank. *Nucleic Acids Research* 28, 1 (2000), 235–242.

[4] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. 2021. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5103–5113.

[5] Wenming Cao, Canta Zheng, Zhiyue Yan, and Weixin Xie. 2022. Geometric deep learning: Progress, applications and challenges. *Science China Information Sciences* 65, 2 (2022), 126101.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 9912–9924.

[7] Wanxing Chang, Ye Shi, Hoang Duong Tuan, and Jingya Wang. 2022. Unified optimal transport framework for universal domain adaptation. In *NeurIPS*.

[8] Chaoqi Chen, Jiongcheng Li, Xiaoguang Han, Xiaoqing Liu, and Yizhou Yu. 2022. Compound domain generalization via Meta-Knowledge encoding. In *CVPR*, 7119–7129.

[9] Guanzi Chen, Jiying Zhang, and Yang Li. 2025. Graph augmentation for cross graph domain generalization. arXiv:2502.18188. Retrieved from https://arxiv.org/abs/2502.18188

[10] Haoran Chen, Xianchen Zhou, Jiwei Zhang, and Hongxia Wang. 2025. L2M-GCN: A new framework for learning robust GCN against structural attacks. *Neurocomputing* 636 (2025), 129962.

[11] Ying Chen, Siwei Qiang, Mingming Ha, Xiaolei Liu, Shaoshuai Li, Jiabi Tong, Lingfeng Yuan, Xiaobo Guo, and Zhenfeng Zhu. 2023. Semi-supervised heterogeneous graph learning with multi-level data augmentation. *ACM Transactions on Knowledge Discovery from Data* 18, 2 (2023), 1–27.

[12] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, M. A. Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022. Learning causally invariant representations for out-of-distribution generalization on graphs. In *NeurIPS*.

[13] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gyn: An efficient algorithm for training deep and large graph convolutional networks. In *KDD*, 257–266.

[14] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*.

[15] Kenyan Dai, Wei Jin, Hui Liu, and Suhang Wang. 2022. Towards robust graph neural networks for noisy graphs with sparse labels. In *WSDM*, 181–191.

[16] Quandy Dai, Xiao-Ming Wu, Jeeren Xiao, Xiao Shen, and Dan Wang. 2022. Graph transfer learning via adversarial domain adaptation with graph convolution. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4908–4922.

[17] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamur, Devis Tuia, and Nicolas Courty. 2018. Depot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, 447–463.
[18] Siemon C. de Lange, Marcel A. de Reus, and Martijn P. van den Heuvel. 2014. The laplacian spectrum of neural networks. *Frontiers in Computational Neuroscience* 7 (2014), 189.
[19] Rui Duan, Mingjian Guang, Junli Wang, Chungang Yan, Hongda Qi, Wenkang Su, Can Tian, and Haoran Yang. 2024. Unifying homophily and heterophily for spectral graph neural networks via triple filter ensembles. In *NIPS*, Vol. 37, 93540–93567.
[20] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. 2021. Adaptive methods for real-world domain generalization. In *CVPR*, 14340–14349.
[21] Xinjie Fan, Qifei Wang, Junjie Ke, Feng Yang, Boqing Gong, and Mingyuan Zhou. 2021. Adversarially adaptive normalization for single domain generalization. In *CVPR*, 8208–8217.
[22] Zheng Fang, Lingjun Xu, Guojie Song, Qingqing Long, and Yingxue Zhang. 2022. Polarized graph neural networks. In *WWW*, 1404–1413.
[23] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. 2020. Graph random neural networks for semi-supervised learning on graphs. In *NeurIPS*, 22092–22103.
[24] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *WWW*, 1528–1538.
[25] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized PageRank. In *ICLR*.
[26] Jiawei Gu, Ziyue Qiao, and Zechao Li. 2025. SpectralGap: Graph-level out-of-distribution detection via laplacian eigenvalue gaps. arXiv:2505.15177. Retrieved from https://arxiv.org/abs/2505.15177
[27] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. 2022. Good: A graph out-of-distribution benchmark. In *NIPS*, Vol. 35, 2059–2073.
[28] Gaoyang Guo, Chaokun Wang, Bencheng Yan, Yunkai Lou, Hao Feng, Junchao Zhu, Jun Chen, Fei He, and Philip Yu. 2022. Learning adaptive node embeddings across graphs. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2022), 6028–6042.
[29] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.
[30] Xiaoxue Han, Huzefa Rangwala, and Yue Ning. 2024. DeCaf: A causal decoupling framework for OOD generalization on node classification. arXiv:2410.20295. Retrieved from https://arxiv.org/abs/2410.20295
[31] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*.
[32] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *NIPS*, Vol. 33, 22118–22133.
[33] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. 2020. Single-side domain generalization for face anti-spoofing. In *CVPR*, 8484–8493.
[34] Robbie P. Joosten, Tim A. H. Te Beek, Elmar Krieger, Maarten L. Hekkelman, Rob W. W. Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. 2010. A series of PDB related databases for everyday needs. *Nucleic Acids Research* 39, suppl_1 (2010), D411–D419.
[35] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. 2024. A comprehensive survey on deep graph representation learning. *Neural Networks* 173 (2024), 106207.
[36] Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. 2024. A survey of graph neural networks in real world: Imbalance, noise, privacy and OOD challenges. arXiv:2403.04468. Retrieved from https://arxiv.org/abs/2403.04468
[37] Wei Ju, Yusheng Zhao, Yifang Qin, Siyu Yi, Jingyang Yuan, Zhiping Xiao, Xiao Luo, Xiting Yan, and Ming Zhang. 2024. Cool: A conjoint perspective on spatio-temporal graph neural network for traffic forecasting. *Information Fusion* 107 (2024), 102341.
[38] Nicolas Keriven. 2022. Not too little, not too much: A theoretical analysis of graph (over) smoothing. In *NIPS*, Vol. 35, 2268–2281.
[39] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. 2021. Selfreg: Self-supervised contrastive regularization for domain generalization. In *ICCV*, 9619–9628.
[40] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980
[41] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
[42] Sandeep Kumar, Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar. 2020. A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research* 21, 22 (2020), 1–60.

[43] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *ICML*, 11906–11917.
[44] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao. 2022. Invariant information bottleneck for domain generalization. In *AAAI*, 7399–7407.
[45] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. 2021. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 8300–8311.
[46] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. 2021. Disentangled contrastive learning on graphs. In *NIPS*, Vol. 34, 21872–21884.
[47] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2022), 7328–7340.
[48] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2022. Out-of-distribution generalization on graphs: A survey. arXiv:2202.07987. Retrieved from https://arxiv.org/abs/2202.07987
[49] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex Kot. 2020. Domain generalization for medical imaging classification with linear-dependency regularization. In *NeurIPS*, 3118–3129.
[50] Hourun Li, Yusheng Zhao, Zhengyang Mao, Yifang Qin, Zhiping Xiao, Jiaqi Feng, Yiyang Gu, Wei Ju, Xiao Luo, and Ming Zhang. 2024. Graph neural networks in intelligent transportation systems: Advances, applications and trends. arXiv:2401.0713. Retrieved from https://arxiv.org/abs/2401.0713
[51] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. 2021. Progressive domain expansion network for single domain generalization. In *CVPR*, 224–233.
[52] Mengyu Li, Jun Yu, Tao Li, and Cheng Meng. 2023. Importance sparsification for sinkhorn algorithm. *Journal of Machine Learning Research* 24, 247 (2023), 1–44.
[53] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M. Hospedales. 2021. A simple feature augmentation for domain generalization. In *ICCV*, 8886–8895.
[54] Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. 2022. Let invariant rationale discovery inspire graph contrastive learning. In *ICML*, 13052–13065.
[55] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. 2021. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *KDD*, 975–985.
[56] Chengwu Liao, Chao Chen, Wanyi Zhang, Suiming Guo, and Chao Liu. 2024. AGENDA: Predicting trip purposes with a new graph embedding network and active domain adaptation. *ACM Transactions on Knowledge Discovery from Data* 18, 8 (2024), 1–25.
[57] Nian Liu, Xiao Wang, Deyu Bo, Chuan Shi, and Jian Pei. 2022. Revisiting graph contrastive learning from the perspective of graph spectrum. In *NeurIPS*.
[58] Yang Liu, Xiang Ao, Fuli Feng, Yunshan Ma, Kuan Li, Tat-Seng Chua, and Qing He. 2023. FLOOD: A flexible invariant learning framework for out-of-distribution generalization on graphs. In *KDD*, 1548–1558.
[59] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. 2021. Generalizing face forgery detection with high-frequency features. In *CVPR*, 16317–16326.
[60] Xin Ma, Yifan Wang, Siyu Yi, Wei Ju, Bei Wu, Ziyue Qiao, Chenwei Tang, and Jiancheng Lv. 2025. PALA: Class-imbalanced graph domain adaptation via prototype-anchored learning and alignment. In *IJCAI*, 3198–3207.
[61] Lucas Mansilla, Rodrigo Echeveste, Diego H. Milone, and Enzo Ferrante. 2021. Domain generalization via gradient surgery. In *ICCV*, 6630–6638.
[62] Hoang Nt and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. arXiv:1905.09550. Retrieved from https://arxiv.org/abs/1905.09550
[63] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. 2018. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE* 106, 5 (2018), 808–828.
[64] Jinhui Pang, Zixuan Wang, Jiliang Tang, Mingyan Xiao, and Nan Yin. 2023. Sa-gda: Spectral augmentation for graph domain adaptation. In *MM*, 309–318.
[65] Gabriel Peyré and Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11, 5–6 (2019), 355–607.
[66] Fengchun Qiao, Long Zhao, and Xi Peng. 2020. Learning to learn single domain generalization. In *CVPR*, 12556–12565.
[67] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, et al. 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* 118, 15 (2021), e2016239118.
[68] Alexander Robey, George J. Pappas, and Hamed Hassani. 2021. Model-based domain generalization. In *NIPS*, Vol. 34, 20210–20229.
[69] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. 2021. Open domain generalization with domain-augmented meta-learning. In *CVPR*, 9624–9633.

[70] Gilbert W. Stewart and Ji-guang Sun. 1990. Matrix perturbation theory.
[71] Henan Sun, Xunkai Li, Lei Zhu, Junyi Han, Guang Zeng, Ronghua Li, and Guoren Wang. 2025. Rethinking graph out-of-distribution generalization: A learnable random walk perspective. arXiv:2505.05785. Retrieved from https://arxiv.org/abs/2505.05785
[72] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. 2021. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*, 15920–15933.
[73] Shyam A. Tailor, Felix Opolka, Pietro Lio, and Nicholas Donald Lane. 2022. Do we need anisotropic graph neural networks? In *ICLR*.
[74] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: Extraction and mining of academic social networks. In *KDD*, 990–998.
[75] Lei Tian and Huaming Wu. 2022. MI-GCN: Node mutual information-based graph convolutional network. In *WWW*, 996–1003.
[76] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008).
[77] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. arXiv:1710.10903. Retrieved from https://arxiv.org/abs/1710.10903
[78] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
[79] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C. Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*.
[80] Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. 2021. On calibration and out-of-domain generalization. In *NIPS*, Vol. 34, 2215–2227.
[81] Haoqing Wang, Huiyu Mai, Yuhang Gong, and Zhi-Hong Deng. 2023. Towards well-generalizing meta-learning via adversarial task augmentation. *Artificial Intelligence* 317 (2023), 103875.
[82] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. 2020. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, 8684–8694.
[83] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8052–8072.
[84] Qixun Wang, Yifei Wang, Hong Zhu, and Yisen Wang. 2022. Improving out-of-distribution generalization by adversarial training with structured priors. arXiv:2210.06807. Retrieved from https://arxiv.org/abs/2210.06807
[85] Ziqi Wang, Marco Loog, and Jan van Gemert. 2021. Respecting domain relations: Hypothesis invariance for domain generalization. In *ICPR*, 9756–9763.
[86] Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. 2021. Learning to diversify for single domain generalization. In *ICCV*, 834–843.
[87] Zhuo Wang, Zezheng Wang, Zitong Yu, Weihong Deng, Jiahong Li, Tingting Gao, and Zhongyuan Wang. 2022. Domain generalization via shuffled style assembly for face anti-spoofing. In *CVPR*, 4123–4133.
[88] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *ICML*.
[89] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. 2020. Unsupervised domain adaptive graph convolutional networks. In *WWW*, 1457–1467.
[90] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.
[91] Zhiqing Xiao, Haobo Wang, Ying Jin, Lei Feng, Gang Chen, Fei Huang, and Junbo Zhao. 2023. SPA: A graph spectral alignment perspective for domain adaptation. In *NIPS*, Vol. 36, 37252–37272.
[92] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. 2021. Infogcl: Information-aware graph contrastive learning. In *NeurIPS*, 30414–30425.
[93] Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. 2024. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *AAAI*, Vol. 38, 9214–9222.
[94] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *ICLR*.
[95] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021. A fourier-based framework for domain generalization. In *CVPR*, 14383–14392.
[96] Junwei Yang, Hanwen Xu, Srbuhi Mirzoyan, Tong Chen, Zixuan Liu, Zequn Liu, Wei Ju, Luchen Liu, Zhiping Xiao, Ming Zhang, et al. 2024. Poisoning medical knowledge using large language models. *Nature Machine Intelligence* 6, 10 (2024), 1156–1168.

[97] Mingqi Yang, Yanming Shen, Rui Li, Heng Qi, Qiang Zhang, and Baocai Yin. 2022. A new perspective on the effects of spectrum in graph neural networks. In *ICML*. PMLR, 25261–25279.
[98] Xihong Yang, Yiqi Wang, Yue Liu, Yi Wen, Lingyuan Meng, Sihang Zhou, Xinwang Liu, and En Zhu. 2024. Mixed graph contrastive network for semi-supervised node classification. *ACM Transactions on Knowledge Discovery from Data* 18, 7 (2024), 1–19.
[99] Kaixuan Yao, Jiye Liang, Jianqing Liang, Ming Li, and Feilong Cao. 2022. Multi-view graph convolutional networks with attention mechanism. *Artificial Intelligence* 307 (2022), 103708.
[100] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *NeurIPS*, 5812–5823.
[101] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2023. Graph domain adaptation via theory-grounded spectral regularization. In *ICLR*.
[102] Junchi Yu, Jian Liang, and Ran He. 2022. Finding diverse and predictable subgraphs for graph domain generalization. arXiv:2206.09345. Retrieved from https://arxiv.org/abs/2206.09345
[103] Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P. Xing. 2022. Towards principled disentanglement for domain generalization. In *CVPR*, 8024–8034.
[104] Kexin Zhang, Shuhan Liu, Song Wang, Weili Shi, Chen Chen, Pan Li, Sheng Li, Jundong Li, and Kaize Ding. 2024. A survey of deep graph learning under distribution shifts: From graph out-of-distribution generalization to adaptation. arXiv:2410.19265. Retrieved from https://arxiv.org/abs/2410.19265
[105] Ruipeng Zhang, Qinwei Xu, Jiangchao Yao, Ya Zhang, Qi Tian, and Yanfeng Wang. 2023. Federated domain generalization with generalization adjustment. In *CVPR*, 3954–3963.
[106] Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Xiaokang Yang, and Pinyan Lu. 2022. M-mix: Generating hard negatives via multi-sample mixing for contrastive learning. In *KDD*, 2461–2470.
[107] Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. 2021. Adversarial separation network for cross-network node classification. In *CIKM*, 2618–2626.
[108] Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. 2022. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *CVPR*, 8035–8045.
[109] Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. 2023. Spectral invariant learning for dynamic graphs under distribution shifts. In *NIPS*, Vol. 36, 6619–6633.
[110] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. 2020. Domain generalization via entropy regularization. In *NeurIPS*, 16096–16107.
[111] Yusheng Zhao, Xiao Luo, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. 2023. Dynamic hypergraph structure learning for traffic flow forecasting. In *ICDE*. IEEE, 2303–2316.
[112] Yusheng Zhao, Xiao Luo, Haomin Wen, Zhiping Xiao, Wei Ju, and Ming Zhang. 2025. Embracing large language models in traffic flow forecasting. In *ACL*.
[113] Yusheng Zhao, Changhu Wang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. 2025. TRACI: A data-centric approach for Multi-Domain generalization on graphs. In *AAAI*, Vol. 39, 13401–13409.
[114] Yusheng Zhao, Qixin Zhang, Xiao Luo, Junyu Luo, Wei Ju, Zhiping Xiao, and Ming Zhang. 2025. Test-time adaptation on graphs via adaptive subgraph-based selection and regularized prototypes. In *ICML*.
[115] Yusheng Zhao, Qixin Zhang, Xiao Luo, Weizhi Zhang, Zhiping Xiao, Wei Ju, Philip S. Yu, and Ming Zhang. 2025. Dynamic Bundling with Large Language Models for Zero-Shot Inference on Text-Attributed Graphs. arXiv:2505.17599. Retrieved from https://arxiv.org/abs/2505.17599
[116] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. 2022. Graph neural networks for graphs with heterophily: A survey. arXiv:2202.07082. Retrieved from https://arxiv.org/abs/2202.07082
[117] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2023. Mixstyle neural networks for domain generalization and adaptation. *International Journal of Computer Vision* 132, 3 (2023), 1–15.
[118] Ronghang Zhu and Sheng Li. 2022. CrossMatch: Cross-Classifier consistency regularization for open-set single domain generalization. In *IZCLR*.
[119] Yun Zhu, Haizhou Shi, Zhenshuo Zhang, and Siliang Tang. 2024. Mario: Model agnostic recipe for improving OOD generalization of graph contrastive learning. In *WWW*, 300–311.
[120] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*, 2069–2080.

Received 9 December 2024; revised 15 July 2025; accepted 29 September 2025