GNN DOMAIN ADAPTATION USING OPTIMAL TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

While Graph Convolutional Networks (GCNs) have recently grown in popularity due to their excellent performance on graph data, their performance under domain shift has not been studied extensively. In this work, we first explore the ability of GCNs to generalize to out-of-distribution data using contextual stochastic block models (CSBMs) on the node classification task. Our results in this area provide the first generalization criteria for GCNs on feature distribution and structure changes. Next we examine a popular Unsupervised Domain Adaptation (UDA) covariate shift assumption and demonstrate that it rarely holds for graph data. Motivated by these results, we propose addressing bias in graph models using domain adaptation with optimal transport - GDOT, which features a transportation plan that minimizes the cost of the joint feature and estimated label distribution $\mathbf{P}(X, \hat{Y})$ between source and target domains. Additionally, we demonstrate that such transportation cost metric serves as a good proxy for estimating transferability between source and target graphs, and is better as a transferability metric than other common metrics like maximum mean discrepancy (MMD). In our controlled CSBM experiments, GDOT demonstrates robustness towards distributional shift, resulting in 90% ROC AUC (vs. the second-best algorithm achieving $< 80\%$ on feature shift). Comprehensive experiments on both semi-supervised and supervised real-world node classification problems show that our method is the only one that performs consistently better than baseline GNNs in the cross-domain adaptation setting.

1 INTRODUCTION

While Graph Neural Networks (GNNs) (Kipf & Welling, 2017; Velickovic et al., 2017; Hamilton et al., 2017; Chami et al., 2022) had great success in node classification on *i.i.d.* data, they are susceptible to deterioration of performance under data distribution shift, where the data used for training and inference come from different distributions (Koh et al., 2021). Consequently, the behavior of GNNs under distribution shift has become a focus of several recent works, which show that a distribution shift in both graph structure and node features can result in deterioration of GNN performance (Zhu et al., 2021; Ma et al., 2021a; Wu et al., 2020). Moreover, Baranwal et al. (2021) investigated the connection between the out-of-distribution generalization and graph structure change. They used a graph convolution on a simplified data-generation model called a contextual stochastic block model (CSBM) (Deshpande et al., 2018). In this paper, we extend their findings to both graph structural and feature changes in test data. Our results show when the generalization of a single-layer graph convolution network would be worse than that of a linear classifier.

At the same time, the behavior of classical (i.e. not GNNs) machine learning models under domain shift has been studied extensively. Common approaches for Unsupervised Domain Adaptation (UDA), where unlabeled target/inference data is available, include learning Domain Invariant Representation Learning (DIRL) (Ben-David et al., 2010; Ganin et al., 2016) and Invariant Risk Minimization (IRM) (Arjovsky et al., 2019). DIRL attempts to align features of source and target data, and IRM assumes that source and target differ in spurious correlations but share the same causal mechanism. However, the effectiveness of such UDA algorithms on GNNs remains unclear. Additionally, these approaches require either covariate shift or multiple training environments which are not justified on structured data. Recently, a new line of work pioneered by Courty et al. (2017) started to emerge. The idea is to assume that there is a non-linear transformation from joint feature space in the source domain onto a target domain. This transformation is not learned directly and instead an optimal transport algorithm optimizes for the best function. This algorithm does not rely on covariate shift assumption as long as a transportation plan exists.

We explore a UDA setting on CSBM and find that the covariate shift condition ($\mathbf{P}_s(Y|X) = \mathbf{P}_t(Y|X)$) rarely holds. Our conclusions confirm those from a recent study by Zhao et al. (2019), which states that variants of domain invariant representation learning cannot adapt to target graphs with innate deficiency (see also Section 3.2). Therefore, we come up with an efficient UDA for graph data using optimal transport, resulting in improvements over the DIRL methods.

We prove that such a transportation plan exists between the source and target CSBM graphs and implement an efficient subgraph sampling (*e.g.* GraphSAINT (Zeng et al., 2019)) to enable mini-batch computation of optimal transport. Further, we demonstrate that the transportation cost C is a metric that can be used to reason about the transferability between source and target graph data, and is much more indicative than metrics such as MMD (Long et al., 2015) and CMD (Zellinger et al., 2017) that are used successfully for transferability estimation (Ibrahim et al., 2021). Under both the supervised and semi-supervised distribution shift, our method outperforms existing domain adaptation algorithms designed for Neural Networks and Graph Neural Networks by an average of 1.5% accuracy over the second best approach on 7 datasets. As far as we know, our approach is the first domain adaptation algorithm in graph learning that does not assume targeted distribution shift, *e.g.* covariate shift.

Our contribution could be summarized as:

- Based on CSBM, we provide insights on the limits of OOD generalization of GNNs for both feature and structural changes. We find that the popular covariate shift assumption rarely holds, thus explaining the commonly observed poor performance of DIRL methods.
- We come up with an efficient mini-batch algorithm GDOT for optimal transport-based domain adaptation, and prove that the optimal mapping exists on CSBM. In addition, we demonstrate that such transportation cost is a superior transferability estimation metric for graph data.
- On synthetic and real datasets, GDOT is able to successfully mitigate the domain shift.

2 Related Work

Unsupervised Domain Adaptation is concerned with situations when training and testing data are drawn from two different distributions, and the goal of UDA algorithms is to transfer knowledge from the source onto target data, obtaining good generalization on target distribution. In the theoretical foundational work of domain adaptation, Ben-David et al. (2010) presented an upper bound of target risk using domain discrepancy measure called $\mathcal{H}$-divergence that represents the distance between source and target distributions. Since then, various domain adaptation algorithms that minimize some definitions of such distance in the latent space were proposed. To achieve domain invariant representation learning, an additional adversarial head is introduced in Ganin et al. (2016), with the goal of distinguishing source and target samples in the latent space. Conditional DANN work - CDAN (Long et al., 2018) - incorporates classifier predictions into the adversarial head, either via linear or multilinear conditioning, further improving UDA performance. There are also methods that seek to optimize some predefined discrepancy measure, which usually exhibits more stable performance and have fewer hyperparameters to tune. For example, Maximum Mean Discrepancy (MMD) (Gretton et al., 2012; Long et al., 2015) measures the difference of distribution means in Hilbert kernel space. More recently, central moment discrepancy metric CMD (Zellinger et al., 2017) attempts to match higher-order means in non-kernel space instead. CMD is shown to be less sensitive (than MMD) to the weight with which such regularization is added to the loss. However, a lot of DIRL algorithms assume a covariate shift: the feature distributions of source and target data are different, but the conditional distribution $P(Y|X)$ is the same between the source and target data. A recent study shows a provable bad generalization of these methods when conditional probability $\mathbf{P}(Y|X)$ varies across domains (Zhao et al., 2019). Optimal transportation OT problem in the literature has been used to compute distances between two distributions, e.g. earth mover distance. Courty et al. (2017) applied OT to the UDA setting, learning an implicit mapping between source and target samples on joint feature and label distribution. Instead of solving an optimal transportation plan, the label matching (Le et al., 2021) domain adaptation algorithm aims to mitigate the label shift when DIRL fails via optimizing the corresponding Wasserstein metric through discriminator training.

Out-of-Distribution Graph Data. Recently, Gui et al. (2022) studied the out-of-distribution data challenges in graph learning, which arise from various domain shifts such as graph size, molecular scaffolds *etc.*. To address the OOD generalization on graph data, early graph domain adaptation (Zhang et al., 2019) algorithms adopt DIRL approaches. UDAGCN (Wu et al., 2020), on top of using DIRL, further enforces global and local consistency and extracts cross-domain node embeddings. Zhu et al. (2021) proposed to improve the OOD generalization on semi-supervised node classification by instance weighting and minimizing CMD metric between source and target node representations. Zhang et al. (2021) tried to capture environment-invariant node properties and explicitly balance the multiple environments to generalize well under distribution shift. Built upon the DIRL work, Wu et al. (2022) tried to address the OOD generalization problem by minimizing the mean and variance of risks from multiple training environments that are generated by the environment generators. In our work, we forego DIRL and explore OT based solutions to graph distribution data shift.

3 Graph Convolution Risk on Domain Adaptation

Below we consider a node classification setup and describe CSBM (Deshpande et al., 2018) used in our analysis.

Definition 3.1 (Contexual Stochastic Block Model (CSBM))**.** *CSBM graph is defined as a tuple (A, X, Y), where A is node adjacency matrix, X describes nodes features and Y defines the nodes labels. Node labels y_i are random variables drawn from Bernoulli distribution (Ber(0.5)), and entries of adjacency matrix $a_{ij} \sim$ Ber(p) if i,j nodes belong to the same class and $a_{ij} \sim$ Ber(q) otherwise. Features are drawn according to $X_i = y_i\mu + \frac{Z_i}{\sqrt{d}}$, $y_i \in \{-1, 1\}$, $\mu \in \mathbb{R}^d$ is the feature mean and $Z_i \in \mathbb{R}^d$ is a gaussian random variable.*

The task of node classification takes nodes features X and structure of the graph A to predict labels Y. We denote the output of graph 1-layer convolution network as $Y = \mathbf{f}(H)$, where $H = D^{-1}(A+I)X$[1] and linear classification with only node features as $Y = \mathbf{f}(X)$. The classifier $\mathbf{f}$ is defined as follows,

$$\mathbf{f}(x) = \begin{cases} 1, & \text{if } \mathbf{w}^T x + b > 0 \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

Below we introduce the challenges of the generalization, study the problem of domain adaptation and analyze the effectiveness of existing methods. Then Sec. 4 describes our graph domain adaptation algorithm using optimal transport.

Definition 3.2 (Unsupervised Domain Adaptation (UDA))**.** *Given source labeled data $\{(x_i^s, y_i^s)\}_{i=1}^n$ containing n samples drawn i.i.d. from the source domain $\mathcal{D}_\mathcal{S}$, and a set of unlabeled target data $\{x_j^t\}_{j=1}^m$ sampled i.i.d. from the target domain $\mathcal{D}_\mathcal{T}$, UDA aims to find a predictive function $\mathbf{f}$ that generalizes well on target domain, using the available labeled source and unlabeled target data.*

3.1 OOD generalization of Graph Convolution Networks

We define the expected binary classification error ϵ on target data $\mathcal{D}^\mathcal{T}$ as,

$$\epsilon = \mathbb{E}_{x,y\sim\mathcal{D}^\mathcal{T}} \mathbb{I}\left[y \cdot (\mathbf{w}^{*T} x + b^*) < 0\right],\ y \in \{-1, 1\} \tag{2}$$

Then ϵ_h and ϵ_x denote the errors on graph convolution network and linear network, respectively.

Theorem 3.1. *Suppose we have training graph $\mathcal{G}_1 \sim$ CSBM$(u, -u, p, q)$ and testing graph $\mathcal{G}_2 \sim$ CSBM$(u', -u, p', q')$, D_{ii} is the degree of node i and $\Phi(|\cdot|)$ is the CDF function of multivariate gaussian distribution defined by distance. Then generalization error of an optimal classifier is given as the following form for a linear layer,*

$$\epsilon_x = 1 - \Phi(|\mu' \cdot \mu|),$$

and for a graph convolutional layer,

$$\epsilon_h = 1 - \Phi(|\sqrt{D'_{ii}} \frac{p'\mu' - q'\mu}{p' + q'} \cdot \mu|).$$

[1] The original graph convolutional network uses $H = D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}}X$. We use a slightly different version here for analysis.

Proof. See Appendix A.1.

Corollary 3.1.1 (Generalization performance under structure shift)**.** *When homophily ratio p/q changes, the generalization of graph convolution is worse than that of a linear layer when number of nodes is fewer than* $2\frac{p+q}{(p-q)^2}$.

Corollary 3.1.2 (Generalization performance under feature shift)**.** *Let* $\mu'_{\parallel}$ *be the horizontal component of* μ' *and* δ *as the relative distance to the origin mean,* $\mu'_{\parallel} = (1-\delta)\mu$*. Then the generalization of graph convolution layer is worse than that of linear layer when,*

$$\delta > 1 - \frac{q}{p - \sqrt{\frac{2(p+q)}{n}}}.$$

While we present here the results for one-layer GNNs and linear perceptron, our results can be extended to multi-layer graph convolutions with activations, which gain popularity recently (Baranwal et al., 2022). We leave this for future work.

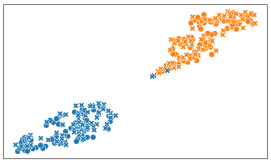
3.2 Challenges of Learning Domain Invariant Representations on Graph Data

We begin with introducing theoretical foundations that led to the creation of various Domain Invariant Representation Learning (DIRL) methods (Long et al., 2015; Ganin et al., 2016; Long et al., 2018). DIRL's key idea is to learn an encoder $\mathbf{g} : X \rightarrow Z \in \mathbb{R}^n$ that minimizes the distribution discrepancy of two domains in the latent space.

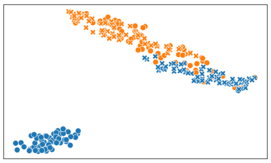
Theorem 3.2 (DIRL generalization bounds)**.** *(Ben-David et al., 2010)* $\mathbf{g} : X \rightarrow \{0, 1\}$ *is a hypothesis function from space* $\mathcal{H}$*,* $\hat{\varepsilon}_{\mathcal{S}}(\mathbf{g})$ *is the empirical risk of g under source domain* $\mathcal{D}_{\mathcal{S}}$ *and* $\varepsilon_{\mathcal{T}}(\mathbf{g})$ *is the true risk of* $\mathbf{g}$ *on target domain. If VC-dimension of* $\mathcal{H}$ *is d and* $\hat{\mathcal{D}}$ *is the empirical distribution each containing* n *samples, then with the probability at least* $1-\delta$*,* $\forall \mathbf{g} \in \mathcal{H}$*:*

$$\varepsilon_{\mathcal{T}}(\mathbf{g}) \leq \hat{\varepsilon}_{\mathcal{S}}(\mathbf{g}) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{\mathcal{S}}, \hat{\mathcal{D}}_{\mathcal{T}}) + \lambda^* + \mathcal{O}\left(\sqrt{\frac{d \log n + \log(1/\delta)}{n}}\right), \tag{3}$$

$\lambda^* = \varepsilon_{\mathcal{T}}(h^*) + \varepsilon_{\mathcal{S}}(h^*)$ is the joint optimal risk that can be achieved on both domains by optimal h^*. A small $\mathcal{H}\Delta\mathcal{H}$-divergence (second term in the bound, which essentially represents some distance between source and target domains)[2] leads to invariant or indistinguishable representation between source and target. DANN (Ganin et al., 2016) is one of the popular DIRL methods that seek to minimize this term by introducing an additional (adversarial) head $\mathbf{f} : Z \rightarrow \{0, 1\}$ that attempts to predict whether a sample came from the source or target domain.



$\mathbf{P}_s(Y|X) \approx \mathbf{P}_t(Y|X)$



$\mathbf{P}_s(Y|X) \neq \mathbf{P}_t(Y|X)$

This encourages the model to learn an embedding $\mathbf{g}$ that does not contain domain-specific features and that has similar distributions on the source and target data. As mentioned before, DIRL methods can mitigate the distributional shift (and assuming the combined error of an ideal joint hypothesis λ^* is small) while achieving small $\hat{\varepsilon}_{\mathcal{S}}(h)$ when covariate shift assumption $\mathbf{P}_s(Y|X) = \mathbf{P}_t(Y|X)$ holds.

In graph structure data however, even subtle structural or feature changes can cause large condition shift so $\mathbf{P}_s(Y|X) \neq \mathbf{P}_t(Y|X)$. In the plot on the right, we project the node TSNE embeddings of source and target CSBM graphs. Two different colors indicate class label, **O** dots are source data and **X** are target samples. When the distribution shift is small and covariate shift assumption holds, DANN can separate different classes well for both source and target domains (top). However, when there is the conditional shift, the classification accuracy on target is low because it only minimizes discrepancy between representations, and classes end up intermixed.

[2] It is symmetric difference of $\mathcal{H}$-divergence by xor operation, *i.e.* $\mathcal{H}\Delta\mathcal{H} = \{h(x) \oplus h'(x) | h, h' \in \mathcal{H}\}$

4 Graph UDA with Optimal Transport

In the previous section, we demonstrated the limitations of DIRL on graph data when covariate shift does not hold. In Transfer Learning literature (Courty et al., 2017) proposed recently an alternative to DIRL: they aim to find a nonlinear transformation between joint (features, labels) space between the source and target domains. To find this transformation, the optimal transport (OT) problem is formulated.

Definition 4.1 (Optimal Transport (Monge, 1781))**.** *Given source and target joint probability distributions $\mathbf{P}_s(X,Y)$, $\mathbf{P}_t(X,Y)$ respectively, and marginal source and target distributions over feature space X: $\mu_s = P_s(X)$ and $\mu_t = P_t(X)$, the Monge problem aims to find a mapping $\mathcal{T}^* : \Omega \rightarrow \Omega$ that transports μ_s to μ_t as,*

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \int_{\Omega} \mathbf{d}(x, \mathcal{T}(x)) d\mu_s(x),$$
$$s.t.\ \mathcal{T}(\mu_s)(B) = \mu_t(\mathcal{T}^{-1}(B)),\ \forall \textit{Borel set} B \in \Omega$$

In the relaxed transportation problem proposed by Kantorovich (1942), one looks for a discrete transport plan (a joint probability distribution) $\gamma \in \mathbf{P}(\Omega \times \Omega)$ such that,

$$\gamma^* = \arg\min_{\gamma \in \Pi(\mu_s, \mu_t)} \sum_{i,j} \mathbf{d}(u_i, v_j)\gamma(i,j)$$

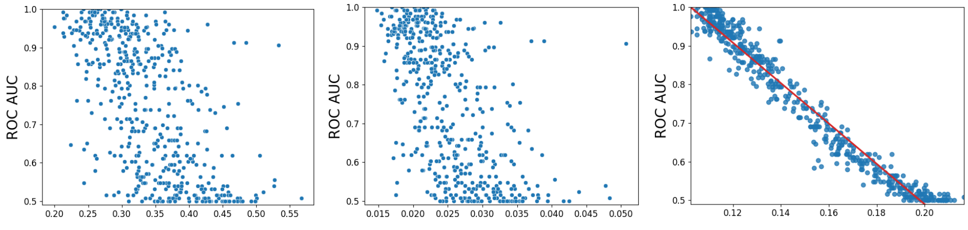
where u_i and v_i are pairs (x_i, y_i) from source $\mathbf{P}_s$ and target $\mathbf{P}_t$ distributions.

Lemma 4.1. *Let $C = \sqrt{\sum_{ij} \mathbf{d}(u_i, v_j)\gamma^*(i,j)}$ and $\mathbf{d}$ be the 2-Wasserstein distance $\mathbf{W}_2^2$, then C is a metric on GCN aggregated representation h between source and target CSBM graphs.*

We leave the proof in Appendix A.2. In the proof, we transform joint distribution on (h_i, y_i) into a Gaussian Mixture Model and find the closed-form solution of 2-Wasserstein distance $\mathbf{d}(u_i, v_j)$ between pair (i,j). Furthermore, there exists an optimal transportation plan γ between two graphs created by CSBM due to the fact that an OT problem has a unique solution when $\mathbf{d}$ is a metric (Villani, 2021).

Remark. When used as a metric, the transportation cost C quantifies the discrepancy between source and target CSBM graphs. $C = 0$ if and only if joint distributions on source and target are the same. Alternative non-optimal transport metrics that are often used for such purpose in DIRL settings are discrepancy measures like MMD (Long et al., 2015) and CMD (Zellinger et al., 2017).

To demonstrate the value of transportation cost as a discrepancy metric we train a 2-layer graph convolution networks using only source samples and compute their OT discrepancy measure on target data representations. The results are presented in Figure 3, which demonstrates a clear correlations of transportation cost and testing performance on CSBM graphs.



(a) CMD **(b)** MMD **(c)** OT Cost C

Figure 3: Comparison of two discrepancy measures (CMD, MMD) and transportation cost C (ours) in same GCN model, x-axis the metric value and y-axis is the test ROC AUC. In syn-csbm-δ, we have 500 source and target pairs of CSBM graphs where target data has a feature distribution shift. Each point in the plot corresponds to the measure computed between a pair of source and target.

Now we formally define the learning problem of unsupervised graph domain adaptation with optimal transport and present our algorithm GDOT. Given source labeled data $\{(x_i^s, y_i^s)\}_{i=1}^M$ in $\mathcal{G}^s$ and unlabeled target data $\{x_j^t\}$ in $\mathcal{G}^t$, we optimize the following loss function,

$$\mathcal{L}_{\text{GDOT}} = \frac{1}{M}\sum_i \mathcal{L}_{\mathcal{S}}(y_i^s, \hat{y}_i^s) + \sum_{ij} \gamma_{ij}(\alpha \| x_i^s - x_j^t \|^2 + \beta \mathcal{L}_{\mathcal{T}}(y_i^s, \hat{y}_j^t))). \tag{4}$$

where $\hat{y}_i$ and $\hat{y}_j$ are predictions on the source and target data produced by a graph neural network $\mathbf{g}$ respectively. Both $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{L}_{\mathcal{T}}$ are cross-entropy loss[3]. The loss consists of classification loss on $\mathcal{G}^s$ and optimal transport cost between source and target graphs; γ_{ij} is the transportation plan between node i in source graph $\mathcal{G}^s$ and j in target $\mathcal{G}^t$, $\sum_{ij}\gamma_{ij} = 1$. The term in the parenthesis is the realization of distance function $\mathbf{d}((x_i^s, y_i^s), (x_j^t, \hat{y}_j^t))$ between source distribution $\mathbf{P}_s(X, Y)$ and estimated target distribution $\mathbf{P}_t^{\mathbf{g}}(X, \mathbf{g}(X, A))$.

Our optimization procedure is similar to mini-batch training of DeepJDOT (Damodaran et al., 2018), apart from the fact that we need to perform neighborhood sampling on graph to obtain source and target subgraph samples for the input of graph neural network $\mathbf{g}$. We adopt a sub-graph based sampling method - GraphSAINT (Zeng et al., 2019) to obtain batch of nodes from source and target $\mathcal{G}_b^s \sim \mathcal{G}^s, \mathcal{G}_b^t \sim \mathcal{G}^t$, respectively. We describe the training process of GDOT in Algorithm 1. Finally, we present the domain adaptation bound using optimal transportation with 1-Wasserstein distance in Appendix A.4. Compared with DIRL UDA algorithms that optimize the bound from Theorem 3.2, GDOT optimizes the Wasserstein distance bound without covariate shift assumption. The transportation cost $C \approx \mathbf{W}_1(\hat{\mathbf{P}}_s, \hat{\mathbf{P}}_t^{\mathbf{g}})$ corresponds to the domain discrepancy measures in DIRL.

Model Analysis. The theoretical analysis of our work is based on graph convolutional networks on CSBM. In practice, the choice of GNN architectures depends on specific problems. There are two hyperparameters α, β for which we chose the values so that $\alpha \sum_{ij}\gamma_{ij}\|x_i - x_j\|^2$ and $\beta \sum_{ij}\gamma_{ij}\mathcal{L}_{\mathcal{T}}$ are on the same or smaller scale than the source loss $\mathcal{L}_{\mathcal{S}}$. In each step, let M be the size of mini-batch and N be the size of features $x_i \in \mathbb{R}^N$ and K classes, the additional computation cost of our method in each epoch is due to computing the transportation cost matrix $C \in \mathbb{R}^{M \times M}$ and solving the optimal transportation γ^*. The cost matrix takes $\mathcal{O}(M^2(N + K))$ time and a traditional network of simple algorithms takes $\mathcal{O}(M^3 \log(M))$. Therefore, the total time complexity of GDOT is $\mathcal{O}(M^3 \log(M) + M^2(N + K))$. To improve the efficiency, one can speed up the computation of optimal via Sinkhorn (Cuturi, 2013), obtaining complexity of $\mathcal{O}(M^2)$.

Algorithm 1: Pseudo code for GDOT optimization

```
Input: Training graph $\{\mathcal{G}^s, X^s, Y\}$; testing graph $\{\mathcal{G}^t, X^t\}$; Graph Sampler SAMPLE;
Output: Graph Neural Network g with trained weights;
for each batch of $(\mathcal{G}_b^s, x_b^s, y_b^s)$ and $(\mathcal{G}_b^t, x_b^t)$ from SAMPLE do
    fix g solve the optimal transportation plan $\gamma$ as of second term Equation 4
    fix $\gamma$ and update the weights of g via backward propagation
end
```

5 Synthetic Experiments

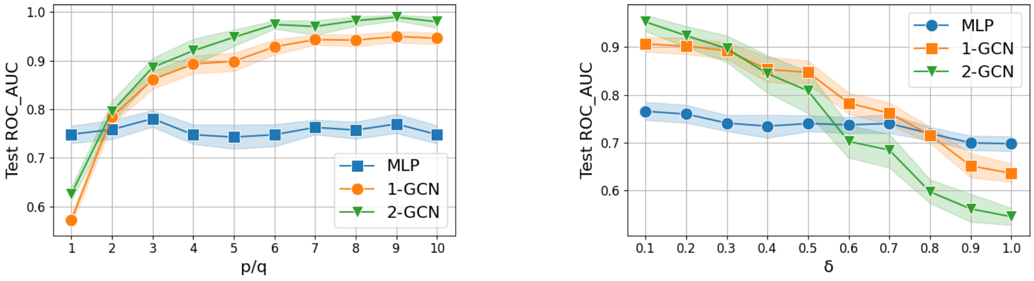
In this section, we seek to confirm our theoretical insights about the generalization ability and transferability of graph models. We do this using two different families of synthetic graphs: (1) CSBM graphs `syn-csbm-pq` and `syn-csbm-`δ with structure or feature shift, respectively (2) synthetic graphs constructed from real datasets `syn-cora` and `syn-products` proposed by Zhu et al. (2020). Each sample in CSBM graph is composed of a training and testing graph, where the testing graph exhibits either kind of distribution shift in various degrees. More information about the data can be found in Table 3 in the Appendix, numerical results for all of the figures are also available in the Appendix. In this section, we compare our method GDOT with standard DIRL algorithms including CMD (Zellinger et al., 2017) and CDAN (Long et al., 2018) using graph convolution network (Kipf & Welling, 2017). We chose these two algorithms as a point of comparison since CMD has been shown to outperform a number of other metric-based DIRL algorithms like MMD (Zellinger et al., 2017). Additionally, CDAN, which uses an adversarial head to impose domain irrelevance and additionally conditions on discriminative information (logits) has been shown to outperform a number of transfer learning techniques (Long et al., 2018).

5.1 OOD generalization of GCN and MLP

In our first experiment, we illustrate how distribution shift on structure and features affects neural networks (2-layer MLP), single and double layers of graph convolution networks on a test graph. We

[3] Note that cross-entropy loss is not a valid distance metric, in practice, when solving for γ we use the euclidean distance between one-hot source label vector and output of softmax$(\cdot)$.

aim to validate the results from theorem 3.1 and show that the generalization of graph convolution networks become worse when the testing graph has more inter-class edges or different feature means. Similar to Baranwal et al. (2021), we set the distance of means of two classes as $2/\sqrt{d}$ and $p/q = 5$ in training graphs, where MLP cannot classify two classes as accurately as graph convolution. In testing graphs, we keep the density of the graph unchanged (*i.e.* average degree) and vary the homophily ratio p/q or feature mean deviation δ as of corollary3.1.1 and corollary3.1.2. As shown in Figure 4, GCNs are affected by both structure and feature shift more than MLP (that cannot separate the training data accurately when shift is large). Calculating the thresholds from corollary 3.1.1, we get $5376 > n = 128$ (p/q=1) and $91.5 < n$ (p/q=2) which aligns well with result in Figure 4a. In Figure 4b, we estimate that when $\delta > 0.67$ GCN would have worse generalization than linear model (corollary 3.1.2), which is confirmed empirically by ROC_AUC plot. In addition, we can see deeper GCNs are more susceptible to feature shift. Therefore improving OOD generalization is crucial to real-world applications since deeper GNNs are often needed to obtain good performance. Overall, our experiments are consistent with our findings in theorem 3.1.



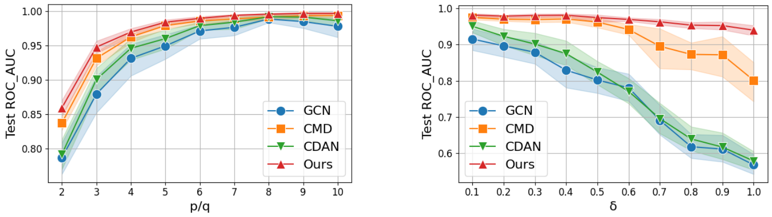
(a) structure shift on syn-csbm-pq

(b) feature shift on syn-csbm-δ

Figure 4: Out-of-distribution generalization of GCNs and a perceptron. We train on a CSBM graph and testing on the other. Each test graph deviates from train graph by sampling p/q from $\{1..10\}$ in syn-csbm-pq or $\delta \in \{0.1..1.0\}$ in syn-csbm-δ.

5.2 Domain adaptation on CSBM

In this experiment, we compare two DIRL algorithms - CDAN and CMD with GDOT on two synthetic CSBM datasets. We tune the hyperparameters of all three algorithms on validation data obtained from the training graph. As depicted in Figure 5a and 5b, GDOT yields better results over both baselines under feature and structure distribution shift. In Appendix A.3, we discuss the theoretical limitations of DIRL algorithms w.r.t. conditional shift. We observe that CDAN, which specifically minimizes the second term in the bound, enjoys a minor improvement over vanilla GCNs. The method CMD does not optimize the $\mathcal{H}\Delta\mathcal{H}$-divergence directly and is seemingly less susceptible to conditional shift.



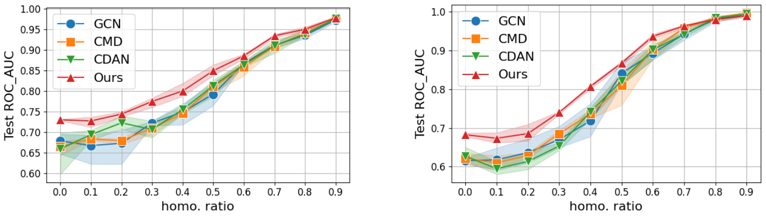
(a) domain adaptation on structure shift

(b) domain adaptation on feature shift

Figure 5: Domain adaptation on syn-csbm-pq and syn-csbm-δ. We use the same setting of OOD generalization experiment except specific domain adaptation loss is applied during training.

5.3 Domain adaptation on existing synthetic graphs

In our last suite of synthetic experiments, we examine the effectiveness of optimal transport domain adaptation on non-CSBM graphs. In the literature of low homophily GNN study (Zhu et al., 2020), syn-cora and syn-products are constructed from existing benchmarks via preferential attachment (Barabási & Albert, 1999) in order to control the homophily ratio. In our theorem A.4, our method should work without the assumption of node features. We aim to validate this on synthetic graphs generated from real graphs like Cora and ogbn-products. We train GCN on the same "easy" graph with 1.0 homophily ratio on both datasets and test on target graphs with various homophily ratios ranging from 0.0 to 0.9. On the multi-class classification task, we compare our method with the same domain adaptation baselines. The results in Figure 6 demonstrate that similar to the previous experiments, GDOT is able to mitigate the distribution shift regardless of noise levels while other algorithms fail to improve over a standard GCN.



(a) Domain adaptation ROC on syn-cora **(b)** Domain adaptation ROC on syn-products

Figure 6: Domain adaptation on datasets constructed from real graphs. We use homophily ratio 1.0 for training and plot the base GCN performance as well as domain adaption algorithms on three test graphs per interval.

6 Real Data Experiments

In this section, we will compare our UDA method with domain adaptation algorithms designed for neural networks and graph neural networks in both supervised and semi-supervised learning settings. In each setting, we will introduce the specific domain adaptation task and how we apply our approach.

Baselines. In addition to the domain adaptation algorithms used in previous sections, we consider the following methods for comprehensive study under distribution shift: (1) MMD (Long et al., 2015) and (2) DANN (Ganin et al., 2016). For graph-specific methods, we focus on the task of node classification and choose three most representative methods: (1) UDAGCN (Wu et al., 2020) couples domain adversarial learning with graph attention mechanism (2) SRGNN-IW (Zhu et al., 2021) proposes to use instance weighting technique on GNN output embeddings under covariate shift (3) Graph-EERM (Wu et al., 2022) proposes to augment training graph for invariance principles in risk minimization. As for our own ablations, we study two variants of our method where either α or β in Equation 4 is zero indicating optimal transport based on only feature and label distribution distance.

6.1 Semi-supervised Node Classification

Zhu et al. (2021) found biased training data in semi-supervised learning can cause dramatic accuracy loss; they provide the algorithm to generate biased training nodes (refered to as OOD training in Table 1) on three semi-supervised learning benchmarks: Cora, Citeseer and PubMed (Sen et al., 2008). In semi-supervised classification, source data is a small number of training nodes and target data are all of the remaining nodes in the same graph. We choose the best-performed GNN architecture from their paper - APPNP (Klicpera et al., 2018) and report the Micro-F1, and Macro-F1 for each method and the accuracy loss compared with IID training data. We are able to reproduce the performance gap between IID and OOD training data in Table 1. We begin by noting that most of the general domain adaptation algorithms such as CMD, MMD, and DANN can help improve the performance because covariate shift holds in this problem. Among these algorithms, we find that directly optimizing discrepancy metrics seems to be more effective and robust (smaller average loss and deviation over 100 runs) than adversarial methods (CDAN and DANN) which often require more tuning. On three datasets, GDOT always achieves top-2 performances, and our ablation without feature distance ($\alpha = 0$) in pairwise distance $\mathbf{d}$ is usually the second to the best, which illustrates the importance of considering label distribution in OT.

Table 1: Semi-supervised classification on three different citation networks with OOD training samples. Results from the original paper (Zhu et al., 2021) are marked †. We annotate the top-2 and other on-par results.

Method	Cora			Citeseer			PubMed		
	Micro-F1	Macro-F1	ΔAcc	Micro-F1	Macro-F1	ΔAcc	Micro-F1	Macro-F1	ΔAcc
IID training	80.8 ± 1.5	80.1 ± 1.3	0	70.2 ± 1.9	66.8 ± 1.7	0	79.7 ± 1.4	78.8 ± 1.4	0
OOD training	71.3 ± 4.1	69.2 ± 3.4	9.5	63.4 ± 1.8	61.2 ± 1.6	6.9	63.4 ± 4.2	58.7 ± 7.0	16.4
MMD	71.5 ± 4.9	69.5 ± 4.6	9.3	64.4 ± 1.2	62.0 ± 1.1	5.9	66.3 ± 4.2	63.5 ± 5.9	13.4
CMD†	72.1 ± 4.4	69.8 ± 3.7	8.7	63.9 ± 0.7	61.8 ± 0.6	6.4	69.4± 3.4	67.6 ± 4.0	10.4
DANN	71.5 ± 5.0	69.5 ± 4.6	9.3	64.7 ± 1.2	62.3 ± 1.1	5.6	64.5 ± 4.9	60.6 ± 7.8	15.2
CDAN	71.5 ± 5.1	69.5 ± 4.7	9.3	64.6 ± 1.3	62.2 ± 1.2	5.6	64.1 ± 5.0	59.9 ± 7.9	15.6
UDAGCN	36.2 ± 4.5	35.4 ± 4.3	44.6	33.8 ± 5.1	31.5 ± 7.7	36.4	40.6 ± 6.8	34.9 ± 6.8	39.1
EERM	68.3 ± 4.3	66.2 ± 3.9	12.5	62.3 ± 1.0	59.5 ± 1.0	7.9	61.6 ± 4.8	56.8 ± 7.7	18.1
SRGNN-IW†	72.0 ± 3.2	69.5 ± 3.7	8.8	66.1 ± 0.9	63.4 ± 0.9	4.2	66.4 ± 4.0	64.0 ± 5.5	13.4
GDOT ($\alpha = 0$)	71.7 ± 4.7	70.2 ± 2.7	9.1	65.3 ± 0.8	63.3 ± 0.8	4.9	71.5 ± 2.9	70.4 ± 3.1	8.2
GDOT ($\beta = 0$)	71.7 ± 4.7	69.7 ± 4.3	9.1	64.6 ± 1.1	62.2 ± 1.0	5.6	68.3 ± 3.9	66.5 ± 4.7	11.4
GDOT	72.6 ± 3.1	70.7± 3.0	8.2	65.6 ± 0.9	63.5 ± 0.9	4.6	73.0 ± 2.5	71.9 ± 2.5	6.7

6.2 Supervised Node Classification

In a fully-supervised setting, transfer learning is often performed across different domains or time periods on graph structure data. We conduct the domain adaptation experiments on citation graphs provided by ArnetMiner (Tang et al., 2008) with both types of shifts: (1) domain shift, where we adopt two pairs of ACM and DBLP graphs *w.r.t.* graph sizes, (2) time shift, where we use ACM graph prior to 2010 as the source and after 2010 as target data. The details about statistics and the graph creation process can be found in Appendix C. We use a 2-layer graph convolution network as the base model and adopt mini-batch training introduced in Section 4. Specifically, we have two graph samplers on both graphs, training nodes in the source batch and all nodes in the target batch are used for solving the optimal transport. In Table 2, the first observation is that algorithms perform differently under different settings. For example, CMD outperforms baseline GCN on two tasks while its performance on large-scale domain transfer problems is worse. GDOT not only performs best among all domain adaptation algorithms but also is consistently better than the baseline model without domain adaptation. The only other algorithm with this property is SRGNN-IW which does not assume the type of distribution shifts as well. Existing graph domain adaptation algorithms - UDAGCN and EERM exhibit limited or negative transfer improvements.

Table 2: Supervised classification on domain and time transfer. We report the accuracy of each method in this table, both micro and macro F1 can be found in Appendix C.

Method	ACM-DBLP$_{small}$	ACM$_{time}$	ACM-DBLP$_{large}$
Base model	68.1 ± 2.1	78.8 ± 1.0	81.1 ± 0.2
MMD	65.9 ± 2.2	79.0 ± 1.0	81.7 ± 0.3
CMD†	75.5 ± 4.4	79.4 ± 0.7	75.2 ± 0.8
DANN	70.1 ± 1.8	79.6 ± 0.4	81.6 ± 0.4
CDAN	75.3 ± 4.3	79.3 ± 1.3	82.1 ± 0.3
UDAGCN	66.4 ± 5.1	79.3 ± 0.5	78.3 ± 2.6
EERM	64.9 ± 3.5	77.3 ± 0.4	81.0 ± 0.4
SRGNN-IW	69.2 ± 1.6	79.5 ± 1.1	81.4 ± 0.4
GDOT ($\alpha = 0$)	74.0 ± 4.7	80.1 ± 0.5	82.1 ± 0.3
GDOT ($\beta = 0$)	71.6 ± 2.3	80.2 ± 0.4	82.3 ± 0.4
GDOT	78.5 ± 4.0	80.3 ± 0.8	82.5 ± 0.3

7 Conclusion

In this work we establish a theoretical analysis on the effect of graph convolution on out-of-distribution data which illustrates the necessity of domain adaptation when domain shift is large. However, popular domain invariant representation learning algorithms assume covariate shift which is a strong assumption for graph data. We present a novel graph domain adaptation framework based on optimal transport to remedy this. Using a number of synthetic and real data node classification experiments, we demonstrate that our method GDOT results in a robust improvement on different kinds of domain shifts without assuming a covariate shift.

As for future work, we have two notable directions to explore: (1) extend our analysis to multi-layer and other types of graph neural networks (2) demonstrate the effectiveness of GDOT on more tasks such as full graph classification and regression.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. *arXiv preprint arXiv:2102.06966*, 2021.

Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Effects of graph convolutions in deep networks. *arXiv preprint arXiv:2204.09297*, 2022.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1–2), 2010. ISSN 0885-6125. doi: 10.1007/s10994-009-5152-4. URL https://doi.org/10.1007/s10994-009-5152-4.

Franccois Bolley, Arnaud Guillin, and Cédric Villani. Quantitative concentration inequalities for empirical measures on non-compact spaces. *Probability Theory and Related Fields*, 137(3): 541–593, 2007.

Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher RÃ©, and Kevin Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research*, 23(89):1–64, 2022. URL http://jmlr.org/papers/v23/20-852.html.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30, 2017.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 447–463, 2018.

Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31, 2018.

DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franccois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 2016.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. Good: A graph out-of-distribution benchmark. *arXiv preprint arXiv:2206.08452*, 2022.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Shibal Ibrahim, Natalia Ponomareva, and Rahul Mazumder. Newer is not always better: Rethinking transferability metrics, their peculiarities, stability and performance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. URL https://openreview.net/forum?id=iz_Wwmfquno.

L Kantorovich. On the translocation of masses. c. r. *Doklady) Acad. Sci. URSS (NS)*, 37:199–201, 1942.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Trung Le, Tuan Nguyen, Nhat Ho, Hung Bui, and Dinh Phung. Lamda: Label matching deep domain adaptation. In *International Conference on Machine Learning*, pp. 6043–6054. PMLR, 2021.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021a.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021b.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998, 2008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*, 2020.

Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*, 2022.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

Shengyu Zhang, Kun Kuang, Jiezhong Qiu, Jin Yu, Zhou Zhao, Hongxia Yang, Zhongfei Zhang, and Fei Wu. Stable prediction on graphs with agnostic distribution shift. *arXiv preprint arXiv:2110.03865*, 2021.

Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. Dane: Domain adaptive network embedding. *arXiv preprint arXiv:1906.00684*, 2019.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pp. 7523–7532. PMLR, 2019.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *arXiv preprint arXiv:2006.11468*, 2020.

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

A Theory details

A.1 Proof of Theorem 3.1

In theorem 3.1, we made several simplifications on original CSBM model to investigate its OOD generalization *w.r.t.* structure and feature distribution shifts. The original CSBM(μ, ν, p, q) is defined to have two different class means μ and ν. Given training and testing distribution as $\mathcal{G} \sim$ CSBM(μ, ν, p, q) and $\mathcal{G}' \sim$ CSBM(μ', ν', p', q'), we let $\nu = \nu' = -\mu$ in CSBM by making $\vec{0}$ the middle point of original feature mean of two classes. Without loss of generality, we let two graphs have same amount of nodes n and focus on one class (*i.e.* $y_i = 1$) of graph $\mathcal{G}'$.

Theorem A.1. *Suppose we have training graph $\mathcal{G}_1 \sim$ CSBM$(u, -u, p_1, q_1)$ and testing graph $\mathcal{G}_2 \sim$ CSBM$(u', -u, p_2, q_2)$, $\Phi(|\cdot|)$ is the CDF function of multivariate gaussian distribution defined by distance.*

The generalization error of an optimal classifier is for linear layer,

$$\epsilon_x = 1 - \Phi(|\mu' \cdot \mu|)$$

for graph convolution layer,

$$\epsilon_h = 1 - \Phi(|\sqrt{D'_{ii}} \frac{p'\mu' - q'\mu}{p' + q'} \cdot \mu|)$$

Proposition 1. *Through training with hinge loss, the linear model and the linearized graph neural network have the same optimal hyperplane $\mathcal{P} = \{\mathbf{x} | \mathbf{w}^T x + b = 0\}$ characterized by $\mathbf{w}^* = \mu$ and $b^* = 0$.*

Proof. In the following, we discuss the classification performances of original and convoluted feature from train graph $\mathcal{G}$. The training data distribution of two models are,

$$x_i \sim \mathcal{N}(\mu, \mathbf{I}), h_i \sim \mathcal{N}\left(\frac{p-q}{p+q}\mu, \frac{1}{\sqrt{D_{ii}}}\mathbf{I}\right), \text{ for } i \in \mathcal{C}_0. \quad (5)$$

We restate the generalization error as expected error indicated by linear classifier $\mathbf{f}(\mathbf{w}^*, b^*)$,

$$\epsilon = \mathbb{E}_{x,y}\mathbb{I}\left[y \cdot (\mathbf{w}^{*T} x + b^*) < 0\right], \; y \in \{-1, 1\} \quad (6)$$

The CDF of the standard normal distribution is denoted by the Φ function. When $d = 1$ (i.e. 1-dimension case), if we translate the distribution into a standard gaussian $\mathcal{N}(0, \mathbf{I})$ by moving μ, the classification error regarding the optimal hyperplane $\{w^*, 0\}$ is the cumulative probability is $\epsilon_x = 1 - \Phi(u')$.

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(\{-\frac{t^2}{2}\})\, dt \quad (7)$$

In standard multivariate (d > 1) distribution, we define the CDF as a monotonic function regarding the distance to w^* and expected classification error is $\epsilon_x = 1 - \Phi(|x \cdot \mu|)$.

We rescale the gaussian distribution output by graph convolution and the expected error ϵ_x and ϵ_h become comparable between their means.

$$\widetilde{h}_i \sim \mathcal{N}\left(\sqrt{D_{ii}} \cdot \frac{p-q}{p+q}\mu, \mathbf{I}\right), \text{ for } i \in \mathcal{C}_0.$$

On training graph, graph convolution has better linear separability when $\epsilon_h < \epsilon_x$ (*i.e.* $\sqrt{D_{ii}} \cdot \frac{p-q}{p+q} > 1$ due to monotonicity of Φ). Note that previous work Baranwal et al. (2021); Ma et al. (2021b) also result in similar conclusion.

On testing graph, we assume only one class centroid moves or graph structure (p,q) changes,

$$x'_i \sim \mathcal{N}(\mu', \mathbf{I}), h'_i \sim \mathcal{N}\left(\sqrt{D'_{ii}} \cdot \frac{p'\mu' - q'\mu}{p' + q'}, \mathbf{I}\right), \text{ for } i \in \mathcal{C}_0.$$

Similarly, the error of graph convolutional and linear layer are $\epsilon_{h'} = 1 - \Phi(\sqrt{D'_{ii}}\frac{p'\mu'-q'\mu}{p'+q'} \cdot \mu)$ and $\epsilon_{x'} = 1 - \Phi(\mu' \cdot \mu)$, which completes the proof.

Now let's discuss the relative error of a linear layer and graph convolution layer when structure or feature deviates from training, respectively.

When graph structure (p/q) changes on testing graph, the ood generalization is same as on training graph with different p' and q'. The degree of node D_{ii} can be estimated as $D'_{ii} \approx \frac{n(p'+q')}{2}$. By solving $\sqrt{D'_{ii}} \cdot \frac{p'-q'}{p'+q'} < 1$, we have GCN yields larger classification error when $n < 2\frac{p'+q'}{(p'-q')^2}$ as of corollary 3.1.1.

When feature mean of the class has shift (μ' in $\epsilon_{h'}$ and $\epsilon_{x'}$), only horizontal component $\mu'_{\|}$ of $\mu' = \mu'_{\|} + \mu'_{\perp}$ affects the error (*i.e.* $\cos(\mu'_{\|}, \mu) = 1$). Let $\mu'_{\|} = (1-\delta)\mu$, we consider the following condition of GCN underpeforms linear,

$$\sqrt{\frac{n(p+q)}{2}} \cdot \frac{p\mu'_{\|} - q\mu}{p+q} < \mu'_{\|}$$

$$p - \frac{q}{1-\delta} < \sqrt{\frac{2(p+q)}{n}}$$

$$\delta > 1 - \frac{q}{p - \sqrt{\frac{2(p+q)}{n}}}.$$

By solving the above equation, we complete the discussion of corollary 3.1.2. □

A.2 Proof of Lemma 4.1

In theorem 3.1, we prove the output of graph convolution for one class is a Gaussian distribution. We first describes the probability density function of K-class CSBM with k Gaussian distribution $\nu_1, \nu_2, ..., \nu_k$ and probability vector $p_1, p_2, ..., p_k$ is drawn from a multinoulli distribution.

$$\mathbf{P}(\nu, p) = \sum_{i=1}^{k} p_k \nu_k \tag{8}$$

Suppose we have $u^s \sim \mathbf{P}(\nu^s, p^s)$ and $v^t \sim \mathbf{P}(\nu^t, p^t)$ for source and target graph, a discrete optimal transport problem solves,

$$\gamma^* = \underset{\gamma \in \Pi(\mathbf{P}_s, \mathbf{P}_t)}{\arg\min} \sum_{i,j} \mathbf{d}(u_i, v_j)\gamma(i, j) \tag{9}$$

Lemma A.2. *Let $C = \sqrt{\sum_{ij} \mathbf{d}(u_i, v_j)\gamma^*(i, j)}$ and $\mathbf{d}$ be the 2-wasserstein distance $\mathbf{W}_2^2$, then C is a metric on GCN aggregated representation h between source and target CSBM graphs.*

Proof. One important property of CSBM is $p_1, p_2, ..., p_k$ is from a multinoulli (categorical) distribution, such that the 2-wasserstein distance $\mathbf{d}(u_i, v_j)$ is computed on two gaussian distributions. Assume $\nu_a^s \sim \mathcal{N}(m_0, \Sigma_0)$ and $\nu_b^t \sim \mathcal{N}(m_1, \Sigma_1)$ are the two gaussian distribution with $p_a^s = 1$ and $p_a^t = 1$, we have their 2-wasserstein distance Dowson & Landau (1982) as,

$$\mathbf{d}(u_i, v_j) = \mathbf{W}_2(\nu_a^s, \nu_b^t)^2 = \|m_0 - m_1\|^2 + \operatorname{Tr}\left(\Sigma_0 + \Sigma_1 - 2(\Sigma_0^{\frac{1}{2}}\Sigma_1\Sigma_0^{\frac{1}{2}})^{\frac{1}{2}}\right). \tag{10}$$

Apparently positivity and symmetry holds on C: $C \geq 0$ and $C(u^s, v^t) = C(v^t, u^s)$. It is also obvious that $C = 0$ if and only if $\mathbf{d}(u_i, v_j) = 0 \; \forall i, j$, in other words, source and target distribution is identical. We denote u, u', u'' from three different CSBM graphs and p, p', p'' as associnated probability vector and prove the triangle inequality of C,

$$C(u, u'') \leq C(u, u') + C(u', u'') \tag{11}$$

Let γ_{01} and γ_{12} be the solution of Equation 9 between u, u' and u', u'', respectively. We construct a custom transportation plan $\gamma_{02} \in \Pi(u, u'')$ as,

$$\gamma_{02}(i, k) = \sum_j \frac{\gamma_{01}(i, j)\gamma_{12}(j, k)}{p'_j} \tag{12}$$

We have marginal distribution of γ_{02} along i as p''_k,

$$\sum_i \gamma_{02}(i,k) = \sum_{i,j} \frac{\gamma_{01}(i,j)\gamma_{12}(j,k)}{p'_j} = \sum_j \frac{p'_j \gamma_{12}(j,k)}{p'_j} = p''_k \tag{13}$$

and similarly $\sum_k \gamma_{02}(i,k) = p_i$. Now we start consider the $C(u,u'')$,

$$\begin{aligned}
C(u,u'') &\leq \sqrt{\sum_{i,k} \gamma_{02}(i,k)\mathbf{d}(u_i,u''_k)} \\
&= \sqrt{\sum_{i,j,k} \frac{\gamma_{01}(i,j)\gamma_{12}(j,k)}{p'_j} \mathbf{W}_2(u_i,u''_k)^2} \\
&\leq \sqrt{\sum_{i,j,k} \frac{\gamma_{01}(i,j)\gamma_{12}(j,k)}{p'_j} \left(\mathbf{W}_2(u_i,u'_j) + \mathbf{W}_2(u'_j,u''_k)\right)^2} \\
&\leq \sqrt{\sum_{i,j,k} \frac{\gamma_{01}(i,j)\gamma_{12}(j,k)}{p'_j} \mathbf{W}_2(u_i,u'_j)^2} + \sqrt{\sum_{i,j,k} \frac{\gamma_{01}(i,j)\gamma_{12}(j,k)}{p'_j} \mathbf{W}_2(u'_j,u''_k)^2} \\
&= \sqrt{\sum_{i,j} \gamma_{01}(i,j)\mathbf{W}_2(u_i,u'_j)^2} + \sqrt{\sum_{j,k} \gamma_{12}(j,k)\mathbf{W}_2(u'_j,u''_k)^2} \\
&= C(u,u') + C(u',u'')
\end{aligned}$$

The first inequality is the definition of optimal transport. The second inequality is because wasserstein distance is a metric and the last inequality is Minkowski inequality. Now, we have shown the tranportation cost between pair of gaussian mixture distributions from K-class CSBM graphs satisfies all the condition of being a metric. □

In our paper, we mainly use 2-class CSBM as a specific case of the lemma.

A.3 Additional discussion on DIRL

Theorem A.3 (Limits of learning invariant representations under conditional shift)**.** Zhao et al. (2019) *Suppose markov chain $X \xrightarrow{\mathbf{g}} Z \xrightarrow{h} \hat{Y}$ and d_{JS} is the Jensen-Shannon distance,*

$$\varepsilon_{\mathcal{S}}(h \circ \mathbf{g}) + \varepsilon_{\mathcal{T}}(h \circ \mathbf{g}) \geq \frac{1}{2}\left(d_{\text{JS}}(\mathcal{D}^Y_{\mathcal{S}}, \mathcal{D}^Y_{\mathcal{T}}) - d_{\text{JS}}(\mathcal{D}^Z_{\mathcal{S}}, \mathcal{D}^Z_{\mathcal{T}})^2\right)$$

.

When $\mathbf{P}(Y|X)$ is different on source and target, minimizing source risk and $\mathcal{H}\Delta\mathcal{H}$-divergence leads to a small JS distance $d_{\text{JS}}(\mathcal{D}^Z_{\mathcal{S}}, \mathcal{D}^Z_{\mathcal{T}})$. As a consequence, the marginal label shift $d_{\text{JS}}(\mathcal{D}^Y_{\mathcal{S}}, \mathcal{D}^Y_{\mathcal{T}})$ dominating the the lower bound of joint source and target risk. If convariate shift does not hold, DIRL cannot achieve accurate predictions on target.

A.4 Domain Adaptation Bound on Optimal Transport

We restate the existing bound on target error for optimal transport based domain adaptation with 1-wasserstein distance.

Theorem A.4. Courty et al. (2017) *Suppose $\mathbf{g} \in \mathcal{H}$ is the graph neural network in hypothesis space, the optimal $\mathbf{g}^*$ is a Lipschitz function with ϕ-probabilistic transfer lipschitzness (PTL)*[4]*. Let $\{\gamma^*|\gamma \in \Pi(\mu_s,\mu_t)\}$ be the optimal mapping in Equation 4 and $\mathbf{W}_1(\hat{\mathbf{P}}_s, \hat{\mathbf{P}}^{\mathbf{g}}_t)$ corresponds the 1-wasserstein distance between two induced empirical distributions from n samples. If we assume $|\mathbf{g}^*(x_1) - \mathbf{g}^*(x_2)| < M, \forall(x_1,x_2)$ and c' is the concentration factor of wasserstein distance Bolley et al. (2007), with the probability at least $1-\delta, \forall \mathbf{g} \in \mathcal{H}$,*

$$\varepsilon_{\mathcal{T}}(\mathbf{g}) \leq \mathbf{W}_1(\hat{\mathbf{P}}_s, \hat{\mathbf{P}}^{\mathbf{g}}_t) + \lambda^* + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{c' \cdot n}}\right) + kM\phi(c), \tag{14}$$

[4] PTL bounds the probability of source target pairs in $(1/c)$-ball *w.r.t.* Π

By optimizing GDOT, the optimal transportation cost C approximate the 1-wasserstein distance $\mathbf{W}_1(\hat{\mathbf{P}}_s, \hat{\mathbf{P}}_t^{\mathbf{g}})$ in the bound. Compared with DIRL bound in theorem 3.2, both bound contain the joint optimal risk λ^* and the differences are on the assumptions. DIRL assume covariate shift while optimal transport assume the existence of transportation plan γ^*

B Model Details

B.1 Implementations

We implement our method and all other baselines using torch-geometric library. We list the graph neural network specifications used in our experiments,

1. Synthetic node classification - model architecture: Graph Convolutional Networks Kipf & Welling (2017), hidden dimension: 16, activation: SiLU, number of layers: 2, dropout: 0.0
2. Semi-supervised node classification - model architecture: APPNP Klicpera et al. (2018), hidden dimension: 32, number of layers:2, dropout: 0.0,
3. Supervised node classification - model architecture: Graph Convolutional Networks Kipf & Welling (2017), hidden dimension: 128, activation: ReLU, number of layers: 2, dropout: 0.2

In GDOT, we use the RandomWalk GraphSAINT Zeng et al. (2019) sampler and set batch size as 256, step size as 50 and walk length as 2. We indepdentently run experiments 10 times and report the mean and standard deviation in all table and figures. The code for each experiment can be found in separate folder in supplementary materials.

B.2 Baseline Hyperparameters

In our experiments, we use the following baselines with hyperparameters tuning on validation. Specifically, each baseline has hyperparameters as follow,

1. For MMD, $\alpha \in \{0.01, 0.1, 0.5, 1\}$ controls the weight of regularization.
2. For CMD, $k \in \{1, 3, 5, 7, 10\}$ determines the number of central moment. $\alpha \in \{0.01, 0.1, 0.5, 1\}$ controls the weight of regularization.
3. For DANN, α is set in $\{0.1, 0.5, 1\}$ for reverse gradients in backward pass. $\beta \in \{0.01, 0.1, 0.5, 1\}$ controls the weight of regularization.
4. For CDAN, λ is a hyper-parameter between source classifier and conditional domain discriminator. $lo \in \{0.01, 0.1, 1\}$ and $hi \in \{0.1, 1, 2\}$ are the initial value and final value of λ. $\beta \in \{0.01, 0.1, 0.5, 1\}$ controls the weight of regularization.
5. For UDAGCN, the balance parameters γ_1 and γ_2 are adjusted carefully in the searching space $\{0.1, 0.3, 0.5, 0.7, 1.0\}$, respectively. The adaptation rate λ is the following schedule: $\lambda = \min(\frac{2}{1+\exp(-10p)} - 1, 0.1)$, and the p is changing from 0 to 1 within the training process as Wu et al. (2020).
6. For EERM, we search the best learning rate $\alpha_f \in \{0.0001, 0.0002, 0.001, 0.005, 0.01\}$ for GNN backbone, the learning rate $\alpha_g \in \{0.0001, 0.001, 0.005, 0.01\}$ for graph editers, the weight $\beta \in \{0.2, 0.5, 1.0, 2.0, 3.0\}$ for combination, the number of edge editing for each node $s \in \{1, 5, 10\}$, the number of iterations $T \in \{1, 5\}$ for inner update before one-step outer update.
7. For SRGNN-IW†, the main hyper parameters in the sampler PPR-S are $\alpha \in \{0.01, 0.1, 0.5, 1\}, \gamma \in \{10, 50, 100, 200, 500\}$. When the graph is large, $\epsilon = 0.001$ is set in the local algorithm for sparse PPR approximation. $\lambda \in \{0.1, 0.5, 1, 2\}$ is the penalty parameter for the discrepancy regularizer. The lower bound for the instance weight B_l is in $\{0.1, 0.2, 0.5, 1.0\}$.
8. Hyperparameters of GDOT α and β are selected between $\{0.01, 0.1, 1\}$.

Table 3: Dataset Statistics. Number of experiments show how many different train and test pairs are tested.

	syn-csbm	syn-cora	syn-products	cora	citeseer	pubmed	DBLP	ACM
# Experiments	500	30	30	100	100	100	2	2
# Nodes	128	1,490	10,000	2,708	3,327	19,717	78,509	23,343
# Edges	1,280	2,965	59,640	5,278	4,614	44,325	1,001,300	162,106
# Classes	2	5	10	7	6	3	5	5

Table 4: Mean ROC and standard deviation per method (with structure shift on `syn-csbm-pq` (Fig. 4a).)

Method					syn-csbm-pq					
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
2-GCN	62.5 ± 5.8	79.5 ± 8.3	88.6 ± 6.8	92.0 ± 9.3	94.8 ± 5.4	97.4 ± 3.0	97.0 ± 6.1	98.2 ± 3.8	98.9 ± 2.0	98.0 ± 3.7
1-GCN	57.2 ± 4.3	78.4 ± 5.4	86.1 ± 6.3	89.3 ± 6.1	89.9 ± 6.0	92.8 ± 5.1	94.3 ± 4.4	94.2 ± 4.5	95.0 ± 3.9	94.6 ± 3.9
MLP	74.8 ± 7.6	75.8 ± 7.1	78.1 ± 5.6	74.8 ± 6.8	74.3 ± 8.0	74.8 ± 7.4	76.3 ± 6.1	75.7 ± 6.7	77.0 ± 6.4	74.8 ± 6.6

C Experiment Details

C.1 DBLP-ACM dataset

We conduct the transfer learning experiments in domain shift and time shift. These experiments use three sets of citation networks, which are constructed on the datasets provided by ArnetMiner Tang et al. (2008). Specifically, for domain shift, we adopt two sets of ACM-DBLP citation networks of different sizes. The small set namely ACM-DBLP$_{\text{small}}$ is proposed by Wu et al. (2020). It includes the papers extracted from ACMv9 (between years 2000 and 2010) and DBLPv8 (after year 2010). The large set, ACM-DBLP$_{\text{large}}$ is constructed on DBLPv12 (before 2017) and ACMv8 (before 2017). As to time shift, we utilize ACMv9 across different time periods, specifically, before or after 2010, to build two citation networks, ACM$_{\text{time}}$. In our experiments, we consider these datasets as undirected graphs and each edge representing a citation relation between two papers. The papers are classified to some of the predefined categories according to its research topics. ACM-DBLP$_{\text{small}}$ has six categories including“Database”, “Data mining”, “Artificial intelligent”, “Computer vision”, “Information Security” and "High Performance Computing". For ACM-DBLP$_{\text{large}}$ and ACM$_{\text{time}}$, there are five categories including “Database”, “Data mining”, “Artificial intelligent”, “Computer vision”, and “Natural Language Processing". We evaluate our proposed methods by conducting multi-label classification on these three sets of citation networks. The dataset statistics are shown in Table 3.

C.2 CSBM dataset generation

In this section, we describe the generation process of `syn-csbm-pq` and `syn-csbm-`δ. According to the definition of CSBM 3.1, we fix the size and average degree of graph (*i.e.* 128 and 10).

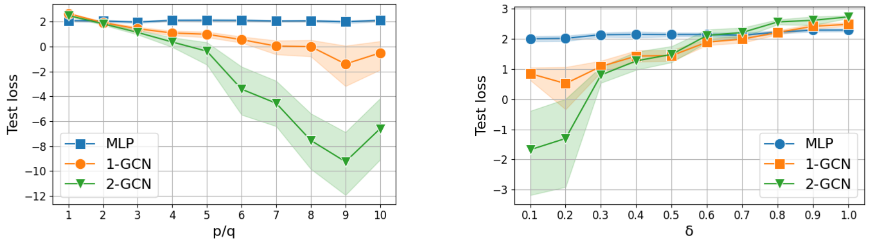
For structure shift, each time we sample a feature meanμ, generate source graph with fixed $p/q = 5$ and sample a p/q between $\{1, ..., 10\}$ to generate the target graph. Such that we ensure the feature of both graph are generated with same gaussian distribution and their homophily ratios are different.

For feature shift, we generate μ' by scale μ by $1 - \delta$ and rotate μ' from 0 to 60 degrees. In corollary 3.1.2, we use the same δ to describe the classification error. When δ is small, feature shift is small and test feature mean μ' is close to original feature mean. The rotation is added to avoid trival adaptation like translation.

The dataset generation code can be found in uploaded code named cSBM_gendata.py.

C.3 More results on OOD generalization

In section 5.1, we show the testing ROC AUC between MLP and graph convolution networks in Figure 4. Here, we provide its numerical results in Table 4 and Table 5. Besides, we provide the testing loss of these methods *w.r.t.* structure and feature shifts between pair of CSBM graphs.



(a) structure shift on CSBM

(b) feature shift on CSBM

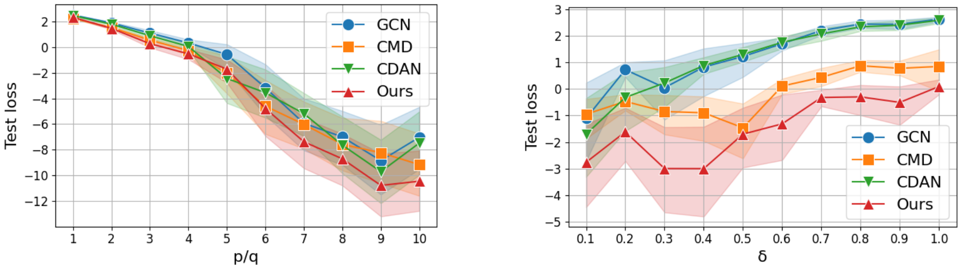
Figure 7: Out-of-distribution generalization of GCNs and a perceptron. We report the test logloss as of Fig. 4 in the main paper.

Table 5: Mean ROC and standard deviation per method (with feature shift on `syn-csbm-`δ (Fig. 4b).)

Method	syn-csbm-δ									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
2-GCN	95.3 ± 6.5	92.4 ± 8.8	89.7 ± 10.0	84.4 ± 11.8	80.8 ± 14.4	70.2 ± 12.7	68.5 ± 14.1	59.7 ± 9.6	56.1 ± 10.4	54.5 ± 6.0
1-GCN	90.6 ± 6.1	90.2 ± 6.3	89.3 ± 6.0	85.3 ± 7.5	84.7 ± 8.2	78.3 ± 7.9	76.2 ± 8.6	71.6 ± 6.8	65.2 ± 8.8	63.6 ± 7.2
MLP	76.6 ± 7.3	76.0 ± 7.1	74.1 ± 5.8	73.5 ± 7.0	74.0 ± 5.7	73.7 ± 6.4	74.0 ± 6.9	71.9 ± 5.6	70.0 ± 5.4	69.8 ± 5.3

C.4 Domain adaptation on synthetic dataset

In Figure 8, we provide the test logloss plot and numerical results of Figure 5b and 5b.



(a) Testing loss of different DA algorithms

(b) Testing Loss of different DA algorithms

Figure 8: Domain adaptation on datasets constructed from real graphs. We use homophily ratio 1.0 for training and plot the base GCN performance as well as domain adaption algorithms on three test graphs per interval.

Table 6: syn-csbm-p/q (Fig. 5a). Mean ROC and standard deviation per method (with structure shift p/q)

Method					syn-csbm-p/q					
	1	2	3	4	5	6	7	8	9	10
GCN	62.6 ± 6.3	78.7 ± 8.2	87.9 ± 8.8	93.1 ± 8.1	94.9 ± 5.7	97.1 ± 3.6	97.6 ± 4.4	98.8 ± 1.7	98.4 ± 3.0	97.8 ± 4.6
CMD	66.0 ± 5.0	83.7 ± 4.0	93.1 ± 3.3	96.2 ± 2.6	97.9 ± 1.5	98.5 ± 1.4	98.8 ± 1.4	99.1 ± 1.2	99.3 ± 0.9	99.3 ± 1.0
CDAN	62.7 ± 5.9	79.2 ± 8.1	90.0 ± 6.5	94.6 ± 5.4	96.0 ± 4.7	97.9 ± 2.0	98.4 ± 3.6	99.1 ± 1.1	99.1 ± 1.4	98.6 ± 2.7
Ours	68.1 ± 5.4	85.9 ± 4.0	94.7 ± 3.0	96.9 ± 2.1	98.4 ± 1.3	98.9 ± 1.0	99.4 ± 0.7	99.5 ± 0.5	99.7 ± 0.5	99.6 ± 0.5

Table 7: syn-csbm-δ (Fig.5b). Mean ROC and standard deviation per method (with feature shift δ)

Method					syn-csbm-δ					
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
GCN	91.5 ± 10.7	89.6 ± 9.9	87.9 ± 11.2	82.9 ± 14.3	80.2 ± 13.8	78.1 ± 13.6	69.1 ± 12.8	61.8 ± 13.1	61.1 ± 13.4	56.8 ± 10.2
CMD	97.5 ± 1.9	97.0 ± 1.8	96.9 ± 2.6	97.1 ± 2.5	96.2 ± 5.1	94.2 ± 5.5	89.5 ± 18.5	87.3 ± 15.5	87.2 ± 19.2	80.1 ± 20.2
CDAN	93.5 ± 7.5	90.2 ± 9.3	87.1 ± 11.4	84.4 ± 12.6	79.0 ± 13.5	72.6 ± 12.4	66.6 ± 12.1	60.8 ± 13.0	59.4 ± 12.2	55.5 ± 8.0
Ours	98.1 ± 1.5	97.8 ± 1.5	98.0 ± 1.8	98.1 ± 1.5	97.4 ± 4.1	96.9 ± 1.8	96.3 ± 2.4	95.3 ± 3.3	95.2 ± 3.9	94.0 ± 5.4

Table 8: Full result of supervised node classification. We report mean and standard deviation on Micro and Macro F1.

Method	ACM-DBLP$_{\text{small}}$		ACM$_{\text{time}}$		ACM-DBLP$_{\text{large}}$	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Base model	68.1 ± 2.1	68.2 ± 2.4	78.8 ± 1.0	76.1 ± 0.7	81.1 ± 0.2	79.1 ± 0.2
MMD	65.9 ± 2.2	65.3 ± 3.1	79.0 ± 1.0	76.1 ± 1.0	81.7 ± 0.3	79.6 ± 0.3
CMD†	75.5 ± 4.4	71.9 ± 6.8	79.4 ± 0.7	75.9 ± 0.7	75.2 ± 0.8	74.7 ± 0.7
DANN	70.1 ± 1.8	70.5 ± 1.7	79.6 ± 0.4	76.9 ± 0.4	81.6 ± 0.4	80.0 ± 0.4
CDAN	75.3 ± 4.3	75.2 ± 4.6	79.3 ± 1.3	76.4 ± 0.9	82.1 ± 0.3	80.0 ± 0.2
UDAGCN	66.4 ± 5.1	64.1 ± 6.2	79.3 ± 0.5	74.6 ± 0.4	78.3 ± 2.6	74.5 ± 2.7
EERM	64.9 ± 3.5	60.0 ± 3.2	77.3 ± 0.4	74.5 ± 0.3	81.0 ± 0.4	78.1 ± 0.4
SRGNN-IW†	69.2 ± 1.6	69.9 ± 1.7	79.5 ± 1.1	76.7 ± 0.8	81.4 ± 0.4	79.5 ± 0.3
GDOT ($\alpha = 0$)	74.0 ± 4.7	73.3 ± 4.9	80.1 ± 0.5	77.2 ± 0.4	82.1 ± 0.3	80.0 ± 0.3
GDOT ($\beta = 0$)	71.6 ± 2.3	71.2 ± 2.6	80.2 ± 0.4	77.3 ± 0.3	82.3 ± 0.4	80.2 ± 0.4
GDOT	78.5 ± 4.0	78.1 ± 4.3	80.3 ± 0.8	77.3 ± 0.6	82.5 ± 0.3	80.4 ± 0.3