Fine-Tuning Graph Neural Networks via Graph Topology Induced Optimal Transport

Jiying Zhang[1,2] , **Xi Xiao**[2*] , **Long-Kai Huang**[1] , **Yu Rong**[1] and **Yatao Bian**[1*]
[1]Tencent AI Lab
[2]Shenzhen International Graduate School, Tsinghua University
{zhangjiy20@mails,xiaox@sz}.tsinghua.edu.cn, yu.rong@hotmail.com,
{hlongkai,yatao.bian}@gmail.com

Abstract

Recently, the pretrain-finetuning paradigm has attracted tons of attention in graph learning community due to its power of alleviating the lack of labels problem in many real-world applications. Current studies use existing techniques, such as weight constraint, representation constraint, which are derived from images or text data, to transfer the invariant knowledge from the pre-train stage to fine-tuning stage. However, these methods failed to preserve invariances from graph structure and Graph Neural Network (GNN) style models. In this paper, we present a novel optimal transport-based fine-tuning framework called GTOT-Tuning, namely, Graph Topology induced Optimal Transport fine-Tuning, for GNN style backbones. GTOT-Tuning is required to utilize the property of graph data to enhance the preservation of representation produced by fine-tuned networks. Toward this goal, we formulate graph local knowledge transfer as an Optimal Transport (OT) problem with a structural prior and construct the GTOT regularizer to constrain the fine-tuned model behaviors. By using the adjacency relationship amongst nodes, the GTOT regularizer achieves node-level optimal transport procedures and reduces redundant transport procedures, resulting in efficient knowledge transfer from the pre-trained models. We evaluate GTOT-Tuning on eight downstream tasks with various GNN backbones[1] and demonstrate that it achieves state-of-the-art fine-tuning performance for GNNs.

1 Introduction

Learning from limited number of training instances is a fundamental problem in many real-word applications. A popular approach to address this issue is fine-tuning a model that pretrains on a large dataset. In contrast to training from scratch, fine-tuning usually requires fewer labeled data, allows for faster training, and generally achieves better performance [Li *et al.*, 2018b; He *et al.*, 2019; Peng *et al.*, 2020].

Conventional fine-tuning approaches can be roughly divided into two categories: (i) weight constraint [Xuhong *et al.*, 2018], i.e. directly constraining the distance of the weights between pretrained and finetuned models. Obviously, they would fail to utilize the topological information of graph data. (ii) representation constraint [Li *et al.*, 2018b]. This type of approach constrains the distance of representations produced from pretrained and finetuned models, preserving the outputs or intermediate activations of finetuned networks. Therefore, both types of approaches fail to take good account of the topological information implied by the middle layer embeddings. However, it has been proven that GNNs explicitly look into the topological structure of the data by exploring the local and global semantics of the graph [Xu *et al.*, 2018; Hu *et al.*, 2020; Xu and et al., 2021; Peng *et al.*, 2020; Zhang *et al.*, 2022], which means that the implicit structure between the node embeddings is very significant. As a result, these fine-tuning methods, which cover only weights or layer activations and ignore the topological context of the input data, are no longer capable of obtaining comprehensive knowledge transfer.

To preserve the local information of finetuned network from pretrained models, in this paper, we explore a principled representation regularization approach. i) Masked Optimal Transport (MOT) is formalized and used as a knowledge transfer procedure between pretrained and finetuned models. Compared with the Typical OT distance [Peyré and Cuturi, 2019], which considers all pair-wise distance between two domains [Courty *et al.*, 2016], MOT allows for choosing the specific node pairs to sum in the final OT distance due to the introduced mask matrix. ii) The topological information of graphs is incorporated into the Masked Wasserstein distance (MWD) via setting the mask matrix as an adjacency matrix, leading to a GTOT distance within the node embedding space. The embedding distance between finetuned and pretrained models is minimized by penalizing the GTOT distance, preserving the local information of finetuned models from pretrained models. Finally, we propose a new fine-tuning framework: GTOT-Tuning as illustrated in Fig. 1.

Using the adjacency relationship between nodes, the proposed GTOT regularizer achieves precise node-level optimal transport procedures and omits unnecessary transport procedures, resulting in efficient knowledge transfer from the pretrained models (Fig. 2). Moreover, thanks to the OT opti-

[*]Correspondence to: Yatao Bian and Xi Xiao

[1]Code: https://github.com/youjibiying/GTOT-Tuning.

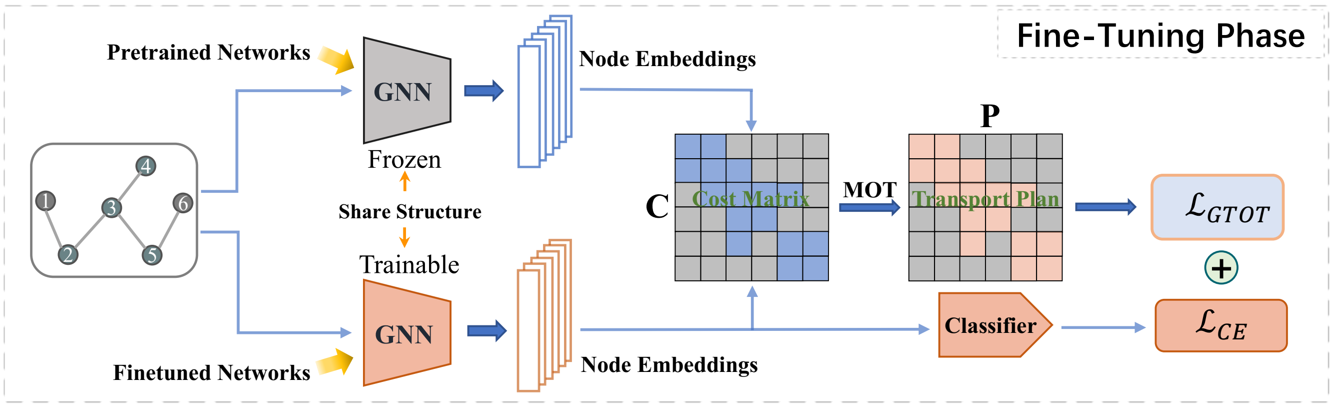

Figure 1: Overall framework of GTOT-Tuning, where $\mathcal{L}_{GTOT}$ denotes GTOT regularizer loss and $\mathcal{L}_{CE}$ represents Cross Entropy loss. The gray lattice of $\mathbf{P}$ indicates that $\mathbf{P}_{ij} = 0$ when the vertex pair (v_i, v_j) is not adjacent. Assume that the input graph has self-loops.

mization that dynamically updates the transport map (weights to summing cosine dissimilarity) during training, the proposed regularizer is able to adaptively and implicitly adjust the distance between fine-tuned weights and pre-trained weights according to the downstream task. The experiments conducted on eight different datasets with various GNN backbones show that GTOT-Tuning achieves the best performance among all baselines, validating the effectiveness and generalization ability of our method.

Contributions: 1) We propose a masked OT problem as an extension of the classical Optimal Transport problem by introducing a mask matrix to constrain the transport procedures. Specially, we define *Masked Wasserstein distance (MWD)* for providing a flexible metric to compare two distributions. 2) We propose a fine-tuning framework called GTOT-Tuning tailored for GNNs, based on the proposed MWD. The core component of this framework, *GTOT Regularizer*, has the ability to utilize graph structure to preserve the local feature invariances between finetuned and pretrained models. To the best of our knowledge, it is the first fine-tuning method tailored for GNNs. 3) Empirically, we conduct extensive experiments on various benchmark datasets, and the results demonstrate a competitive performance of our method.

2 Related Work

Pretraining GNNs. Pre-training techniques have been shown to be effective for improving the generalization ability of GNN models. The existing methods for pre-training GNNs are mainly based on self-supervised paradigms. Some self-supervised tasks, e.g. context prediction [Hu *et al.*, 2020; Rong *et al.*, 2020], edge/attribute generation [Hu *et al.*, 2020] and contrastive learning ([You *et al.*, 2020; Xu and et al., 2021]), have been designed to obtain knowledge from unlabeled graphs. However, most of these methods only use the vanilla fine-tuning methods, i.e. the pretrained weights act as the initial weights for downstream tasks. It remains open to exploiting the optimal performance of the pre-trained GNN models. Our work is expected to utilize the graph structure to achieve better performance on the downstream tasks.

Fine-Tuning in Transfer Learning. Fine-tuning a pre-trained model to downstream tasks is a popular paradigm in transfer learning (TL). [Donahue and Jia, 2014; Oquab *et al.*, 2014] indicate that transferring features extracted by pre-trained AlexNet model to downstream tasks yields better performance than hand-crafted features. Further studies by [Yosinski *et al.*, 2014; Agrawal *et al.*, 2014] show that fine-tuning the pre-trained networks is more effective than fixed pre-trained representations. Recent research primarily focuses on how to better tap into the prior knowledge of pre-trained models from various perspectives. i) Weights: L2_SP [Xuhong *et al.*, 2018] propose a L_2 distance regularization that penalizes L_2 distance between the fine-tuned weights and pre-trained weights. ii) Features: DELTA [Li *et al.*, 2018b] constrains feature maps with the pre-trained activations selected by channel-wise attention. iii) Architecture: BSS [Chen *et al.*, 2019] penalizes smaller singular values to suppress untransferable spectral components to prevent negative transfer. StochNorm [Kou *et al.*, 2020] uses Stochastic Normalization to replace the classical batch normalization of pre-trained models. Despite the encouraging progress, it still lacks a fine-tuning method specifically for GNNs.

Optimal Transport. Optimal Transport is frequently used in many applications of deep learning, including domain adaption [Courty *et al.*, 2016; Xu and et al., 2020], knowledge distillation [Chen and Wang, 2021], sequence-to-sequence learning [Chen and Zhang, 2019], graph matching [Xu *et al.*, 2019], cross-domain alignment [Chen and Gan, 2020], rigid protein docking [Ganea *et al.*, 2022], and GNN architecture design [Bécigneul *et al.*, 2020]. Some classical solutions to OT problem like Sinkhorn Algorithm can be found in [Peyré and Cuturi, 2019]. The work closely related to us may be [Li *et al.*, 2020], which raises a fine-tuning method based on typical OT. The significant differences are that our approach is i) based on the proposed MOT, ii) tailored for GNNs, and iii) able to exploit the structural information of graphs.

3 Preliminaries

Notations. We define the inner product $\langle \cdot, \cdot \rangle$ for matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{m \times n}$ by $\langle \mathbf{U}, \mathbf{V} \rangle = \mathrm{tr}(\mathbf{U}^\top \mathbf{V}) = \sum_{i,j} \mathbf{U}_{ij} \mathbf{V}_{ij}$. $\mathbf{I} \in \mathbb{R}^{n \times n}$ denotes the identity matrix, and $\mathbf{1}_n \in \mathbb{R}^n$ represents the vector with ones in each component of size n. We use boldface letter $\mathbf{x} \in \mathbb{R}^n$ to indicate an n-dimensional vector, where $\mathbf{x}_i$ is the i^{th} entry of $\mathbf{x}$. Let $G(\mathcal{V}, \mathcal{E})$ be a graph with vertices $\mathcal{V}$ and edges $\mathcal{E}$. We denote by $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ the adjacency matrix of G and $\odot$ the Hadamard product. For convenience, we follow the terms of transfer learning and call the signal graph output from fine-tuned models (target networks) as **target graph**, and correspondingly, the output from pre-trained models (source networks) is called **source graph**. Note that these two graphs have the same adjacency matrix in fine-tuning setting.

Wasserstein Distance. Wasserstein distance (WD) [Peyré and Cuturi, 2019] is commonly used for matching two empirical distributions (e.g., two sets of node embeddings in a graph) [Courty *et al.*, 2016]. The WD can be defined below.

Definition 1. *Let $\alpha = \sum_i^n \mathbf{a}_i \delta_{\mathbf{x}_i}$ and $\beta = \sum_i^m \mathbf{b}_i \delta_{\mathbf{y}}$ be two discrete distributions with $\delta_{\mathbf{x}_i}$ as the Dirac function concentrated at location $\mathbf{x}$. $\Pi(\alpha,\beta)$ denotes all the joint distributions $\gamma(\mathbf{x},\mathbf{y})$, with marginals $\alpha(\mathbf{x})$ and $\beta(\mathbf{y})$. $\mathbf{a} \in \mathbb{R}_+^n$ and $\mathbf{b} \in \mathbb{R}_+^m$ are weight vectors satisfying $\sum_{i=1}^n \mathbf{a}_i = \sum_{i=1}^m \mathbf{b}_i = 1$. The definition of Wasserstein distance between the two discrete distributions α, β is:*

$$\mathcal{D}_w(\alpha,\beta) = \inf_{\gamma\in\Pi(\alpha,\beta)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} c(\mathbf{x},\mathbf{y}) \tag{1}$$

or

$$\mathbf{L}_w(\mathbf{a},\mathbf{b}) = \min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \langle \mathbf{P},\mathbf{C}\rangle = \min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{ij} \mathbf{P}_{ij}\mathbf{C}_{ij} \tag{2}$$

*where $\mathbf{U}(\mathbf{a},\mathbf{b}) = \{\mathbf{P} \in \mathbb{R}^{n\times m} \mid \mathbf{P}\mathbf{1}_m = \mathbf{a}, \mathbf{P}^\top \mathbf{1}_n = \mathbf{b}\}$ and $\mathbf{C}_{ij} = c(\mathbf{x}_i,\mathbf{y}_j)$ is a cost scalar representing the distance between $\mathbf{x}_i$ and $\mathbf{y}_j$. The $\mathbf{P} \in \mathbb{R}^{n\times m}$ is called as **transport plan** or **transport map**, and $\mathbf{P}_{ij}$ represents the amount of mass to be moved from $\mathbf{a}_i$ to $\mathbf{b}_j$. $\mathbf{a},\mathbf{b}$ are also known as marginal distributions of $\mathbf{P}$.*

The discrepancy between each pair of samples across the two domains can be measured by the optimal transport distance $\mathcal{D}_w(\alpha,\beta)$. This seems to imply that $\mathbf{L}_w(\mathbf{a},\mathbf{b})$ is a natural choice as a representation distance between source graph and target graph for GNN fine-tuning. However, the local dependence exists between source and target graph, especially when the graph is large (see section 5). Therefore, it is not appropriate for WD to sum the distances of all node pairs. Inspired by this observation, we propose a masked optimal transport problem, as an extension of typical OT (section 4).

4 Masked Optimal Transport

Recall that in typical OT (definition 1), $\mathbf{a}_i$ can be transported to any $\mathbf{b}_j \in \{\mathbf{b}_k\}_{k=1}^m$ with amount $\mathbf{P}_{ij}$. Here, we assume that the $\mathbf{a}_i$ can only be transported to $\mathbf{b}_j \in \mathcal{U}$ where $\mathcal{U} \subseteq \{\mathbf{b}_k\}_{k=1}^m$ is a subset. This constraint can be implemented by limiting the transport plan: $\mathbf{P}_{ij} = 0$ if $\mathbf{b}_j \notin \mathcal{U}$. Thus, the problem is formalized as follows by introducing the mask matrix.

Definition 2 (Masked Wasserstein Distance)**.** *Following the same notation of definition 1 and given a mask matrix $\mathbf{M}$[2] $\in \{0,1\}^{n\times m}$ where every row or column is not all zeros, the masked Wasserstein distance (MWD) is defined as*

$$\mathbf{L}_{mw}(\mathbf{M},\mathbf{a},\mathbf{b}) = \min_{\mathbf{P}\in\mathbf{U}(\mathbf{M},\mathbf{a},\mathbf{b})} \langle \mathbf{M}\odot\mathbf{P},\mathbf{C}\rangle, \tag{3}$$

[2]In this paper, $\mathbf{M}_{ij} = 0$ represents the ij-th element of the matrix being masked. Full Paper: https://arxiv.org/abs/2203.10453

Algorithm 1 Computing Masked Wasserstein Distance

```
Input: Cost matrix $\mathbf{C}$, Mask matrix $\mathbf{M} \in \{0,1\}^{n\times m}$,
Marginals $\mathbf{a} \in \mathbb{R}_+^n$, $\mathbf{b} \in \mathbb{R}_+^m$, Iteration number $\mathcal{K}$.
Initialize: $\mathbf{u} = \mathbf{v} = \mathbf{0}$.
for $i = 1,2,3,\cdots,\mathcal{K}$ do
    $\mathbf{u} = \epsilon(\log(\mathbf{a}) - \log(\mathbf{M}\odot\exp(-\mathbf{C} + \mathbf{u}\mathbf{1}_m^\top + \mathbf{1}_n\mathbf{v}^\top)\mathbf{1}_m)) + \mathbf{u}$
    $\mathbf{v} = \epsilon(\log(\mathbf{b}) - \log((\mathbf{M}\odot\exp(-\mathbf{C} + \mathbf{u}\mathbf{1}_m^\top + \mathbf{1}_n\mathbf{v}^\top))^\top\mathbf{1}_n)) + \mathbf{v}$
end for
$\mathbf{P} = \mathbf{M}\odot\exp(-\mathbf{M}\odot\mathbf{C} + \mathbf{u}\mathbf{1}_m^\top + \mathbf{1}_n\mathbf{v}^\top)$
$\mathcal{D}_{mw} = \langle\mathbf{P},\mathbf{C}\rangle$
Output: $\mathbf{P}, \mathcal{D}_{mw}$
```

where $\mathbf{U}(\mathbf{M},\mathbf{a},\mathbf{b}) := \{\mathbf{P} \in \mathbb{R}_+^{n\times m} \mid (\mathbf{M}\odot\mathbf{P})\mathbf{1}_m = \mathbf{a}, (\mathbf{M}\odot\mathbf{P})^\top\mathbf{1}_n = \mathbf{b}, \mathbf{P}\odot(\mathbf{1}_{n\times m} - \mathbf{M}) = \mathbf{0}_{n\times m}\}$ and $\mathbf{C} \in \mathbb{R}^{n\times m}$ is a cost matrix.

From Eq. (3), the mask matrix $\mathbf{M}$ indicates that the elements of $\mathbf{P}$ need to be optimized, in other words, the costs need to be involved in the summation when calculating the inner product. Notably, different mask matrices $\mathbf{M}$ lead to different transport maps and obtain the OT distance associated with $\mathbf{M}$. One can design $\mathbf{M}$ carefully to get a specific WD. Moreover, the defined MWD can recover WD by setting $\mathbf{M} = \mathbf{1}_{n\times m}$ and it is obvious that $\mathbf{L}_{mw}(\mathbf{M},\mathbf{a},\mathbf{b}) \geq \mathbf{L}_w(\mathbf{a},\mathbf{b})$. This problem can obtain approximate solutions by adding an entropic regularization penalty, which is essential for deriving algorithms suitable for parallel iterations [Peyré and Cuturi, 2019].

Proposition 1. *The solution to definition 2 with entropic regularization $\epsilon\mathbf{H}(\mathbf{M}\odot\mathbf{P})$[3] is unique and has the form*

$$\mathbf{P}_{ij} = \mathbf{u}_i\mathbf{M}_{ij}\mathbf{K}_{ij}\mathbf{v}_j \tag{4}$$

where $\mathbf{K}_{ij} = \exp(-\mathbf{C}_{ij}/\epsilon)$ and $(\mathbf{u},\mathbf{v}) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$ are two unknown scaling variables.

It is clear from the result that MWD is not equivalent to weighting the distance matrix $\mathbf{C}$ directly [Xu and et al., 2020], so the masked OT problem is nontrivial. We provide the proof in Appendix and the key to it is the observation that $\exp(\mathbf{M}\odot\mathbf{X}) = \mathbf{M}\odot\exp(\mathbf{X}) + \mathbf{1}_{n\times m} - \mathbf{M}$, where $\mathbf{X} \in \mathbb{R}^{n\times m}$ is an arbitrary given matrix.

A conceptually simple method to compute the solution would be through the Sinkhorn Knopp iterations. However, as we know, the Sinkhorn algorithm suffers from numerical overflow when the regularization parameter ϵ is too small compared to the entries of the cost matrix $\mathbf{C}$ [Peyré and Cuturi, 2019]. This problem may be more severe when the sparsity of the mask matrix is large. Fortunately, this concern can be somewhat alleviated by performing computations in the log domain. Therefore, for numerical stability and speed,

[3]Namely, $\min_{\mathbf{P}\in\mathbf{U}(\mathbf{M},\mathbf{a},\mathbf{b})} \langle\mathbf{M}\odot\mathbf{P},\mathbf{C}\rangle - \epsilon\mathbf{H}(\mathbf{M}\odot\mathbf{P})$, where $\mathbf{H}(\cdot)$ is Entropy function. Assume that $0\log 0 = 0$ to ensure that $\mathbf{H}(\mathbf{M}\odot\mathbf{P})$ is well defined.

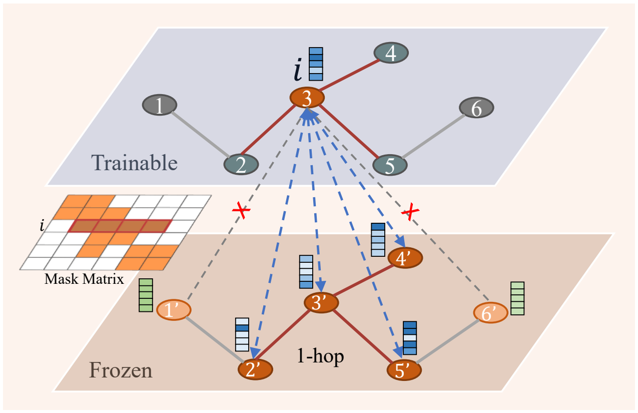

Figure 2: An example of calculating GTOT distance. When preserving the i-th node representation in target graph, the corresponding vertices in the source graph within 1-hop vertices distance (i.e.$\{2', 3', 4', 5'\}$) would be considered. This implies that GTOT regularizer is a local knowledge transfer regularizer.

we propose the Log-domain masked Sinkhorn algorithm (the derivation can be seen in Appendix). Algorithm 1 provides the whole process.

5 Fine-Tuning GNNs via Masked OT Distance

In our proposed framework, we use Masked Optimal Transport (MOT) for GNNs fine-tuning, where a transport plan $\mathbf{P} \in \mathbb{R}^{n\times m}$ is learned to optimize the knowledge transfer between pretrained and fine-tuned models. There are several distinctive characteristics of MOT that make it an ideal tool for fine-tuning GNNs. (i) **Self normalization**: All the elements of $\mathbf{P}$ sum to 1. (ii) **Sparsity**: The introduced mask matrix can effectively limit the sparsity of the transport map, leading to a more interpretable and robust representation regularizer for fine-tuning (See Fig. 12 in Appendix.). (iii) **Efficiency**: Our solution can be easily obtained by Algorithm 1 that only requires matrix-vector products, therefore is readily applicable to GNNs. (iv) **Flexibility**: The Masked matrix can assign exclusive transport plans for specific transport tasks and reduce the unnecessary optimal transport procedure.

5.1 GTOT Regularizer

Given the node embeddings $\{\mathbf{x}_i^S\}_{i=1}^{|\mathcal{V}|}$ and $\{\mathbf{x}_i^T\}_{i=1}^{|\mathcal{V}|}$ extracted from pre-trained GNN and fine-tuned GNN message passing period, respectively, we calculate the **cosine dissimilarity** $\mathbf{C}_{ij} = \frac{1}{2}(1 - \cos(\mathbf{x}_i^S, \mathbf{x}_j^T))$ as the cost matrix of MWD. Indeed, cosine dissimilarity is a popular choice that used in many OT application [Chen and Gan, 2020; Xu and et al., 2020]. Intuitively, in most cases, the more (geodesic) distant two vertices in a graph are, the less similar their features are. This implies that when the MWD is used for fine-tuning, the adjacency of the graph should be taken into account, rather than summing all pairwise distances of the cost matrix. Therefore, we set the mask matrix as the adjacency matrix $\mathbf{A}$ (with self-loop) here, based on the assumption of 1-hop dependence, i.e., the vertices in the target graph are assumed to be related only to the vertices within 1-hop of the corresponding vertices in the source graph (Fig. 2). This assumption is reasonable because the neighboring node embeddings extracted through the (pretrained) GNN are somewhat similar [Li *et al.*, 2018a], which also reveals that considering only the node-to-node distance but without the neighbors is sub-optimal. We call this MOT with graph topology as *Graph Topology induced Optimal Transport (GTOT)*. With marginal distributions being uniform, the **GTOT regularizer** is formally defined as

$$\mathbf{L}_{mw}(\mathbf{A}, \mathbf{q}, \mathbf{q}) = \min_{\mathbf{P}\in\mathbf{U}(\mathbf{A},\mathbf{q},\mathbf{q})} \langle \mathbf{A} \odot \mathbf{P}, \mathbf{C}\rangle \tag{5}$$

where $\mathbf{q}$ is defined as uniform distribution $\mathbf{1}_{|\mathcal{V}|}/|\mathcal{V}|$. Noted that the inner product in Eq. (5) can be rewritten as $\sum_i \sum_{j\in\mathcal{N}(i)\bigcup\{i\}} \mathbf{P}_{ij}\mathbf{C}_{ij}$, where $\mathcal{N}(i)$ denotes the set of neighbors of node i. Moreover, our method has the potential to use sparse matrix acceleration when the adjacency matrix is sparse. Our method is easy to extend to i) the weighted graph by element-wise multiplying the edge-weighted matrix $\mathbf{W}$ with cost matrix, i.e. $\langle \mathbf{A}\odot\mathbf{P}, \mathbf{W}\odot\mathbf{C}\rangle$ or ii) the k-hop dependence assumption by replacing $\mathbf{A}$ with $\mathbf{A}^k$ or $g(\mathbf{A})$, where g is a polynomial function. Similarly, using MOT we can define the MGWD-based Regularizer for regularizing the representations on edge-level (see appendix).

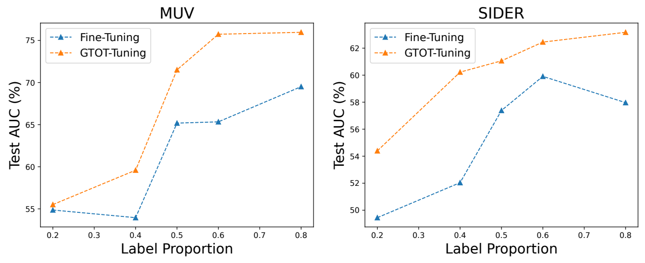

Figure 3: Test AUC with different proportion of labeled data.

5.2 GTOT-Tuning Framework

Unlike the OT representation regularizer proposed by [Li *et al.*, 2020] which calculates the Wasserstein distance between mini-batch samples of training data, GTOT regularizer in our framework focus on the single sample, that is, the GTOT distance between the node embeddings of one source graph and the corresponding target graph. Considering the GTOT distance between nodes allows for knowledge transfer at the node level. This makes the fine-tuned model able to output representations that are more appropriate for the downstream task, i.e., node representations that are as identical as possible to these output from the pre-trained model, but with specific differences in the graph-level representations.

Overall Objective. Given N training samples $\{(G_1, y_1), \cdots, (G_N, y_N)\}$, the overall objective of GTOT-Tuning is to minimize the following loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} l(f, G_i, y_i) \tag{6}$$

where $l(f, G_i, y_i) := \phi(f(G_i), y_i) + \lambda \mathbf{L}_{mw}(\mathbf{A}^{(i)}, \mathbf{q}^{(i)}, \mathbf{q}^{(i)})$, f denotes a given GNN backbone, λ is a hyper-parameter for

Datasets	BBBP	Tox21	Toxcast	SIDER	ClinTox	MUV	HIV	BACE	**Average**
Methods					GIN (contexpred)				
Fine-Tuning (baseline)	68.0±2.0	75.7±0.7	63.9±0.6	60.9±0.6	65.9±3.8	75.8±1.7	77.3±1.0	79.6±1.2	70.85
L2_SP [Xuhong *et al.*, 2018]	68.2±0.7	73.6±0.8	62.4±0.3	61.1±0.7	68.1±3.7	76.7±0.9	75.7±1.5	82.2±2.4	70.25
DELTA[Li *et al.*, 2018b]	67.8±0.8	75.2±0.5	63.3±0.5	62.2±0.4	**73.4±3.0**	**80.2±1.1**	77.5±0.9	81.8±1.1	72.68
Feature(DELTA w/o ATT)	61.4±0.8	71.1±0.1	61.5±0.2	62.4±0.3	64.0±3.4	78.4±1.1	74.0±0.5	76.3±1.1	68.64
BSS[Chen *et al.*, 2019]	68.1±1.4	**75.9±0.8**	63.9±0.4	60.9±0.8	70.9±5.1	78.0±2.0	77.6±0.8	82.4±1.8	72.21
StochNorm [Kou *et al.*, 2020]	69.3±1.6	74.9±0.6	63.4±0.5	61.0±1.1	65.5±4.2	76.0±1.6	77.6±0.8	80.5±2.7	71.03
GTOT-Tuning (Ours)	**70.0±2.3**	75.6±0.7	**64.0±0.3**	**63.5±0.6**	72.0±5.4	80.0±1.8	**78.2±0.7**	**83.4±1.9**	**73.34**
Methods					GIN (supervised contexpred)				
Fine-Tuning (baseline)	68.7±1.3	78.1±0.6	65.7±0.6	62.7±0.8	72.6±1.5	81.3±2.1	79.9±0.7	84.5±0.7	74.19
L2_SP [Xuhong *et al.*, 2018]	68.5±1.0	78.7±0.3	65.7±0.4	**63.8±0.3**	71.8±1.6	85.0±1.1	77.5±0.9	84.5±0.9	74.44
DELTA[Li *et al.*, 2018b]	68.4±1.2	77.9±0.2	65.6±0.2	62.9±0.8	72.7±1.9	**85.9±1.3**	75.6±0.4	79.0±1.1	73.50
Feature(DELTA w/o ATT)	68.6±0.9	77.9±0.2	65.7±0.2	63.0±0.6	72.7±1.5	85.6±1.0	75.7±0.3	78.4±0.7	73.45
BSS[Chen *et al.*, 2019]	70.0±1.0	78.3±0.4	65.8±0.3	62.8±0.6	73.7±1.3	78.6±2.1	79.9±1.4	84.2±1.0	74.16
StochNorm [Kou *et al.*, 2020]	69.8±0.9	78.4±0.3	66.1±0.4	62.2±0.7	73.2±2.1	82.5±2.6	80.2±0.7	84.2±2.3	74.58
GTOT-Tuning (Ours)	**71.5±0.8**	**78.6±0.3**	**66.6±0.4**	63.3±0.6	**77.9±3.2**	85.0±0.9	**81.1±0.5**	**85.3±1.5**	**76.16**

Table 1: Test ROC-AUC (%) of GIN (contexpred) and GIN (supervised contexpred) on downstream molecular property prediction benchmarks.

balancing the regularization with the main loss function, and $\phi(\cdot)$ is Cross Entropy loss function.

6 Theoretical Analysis

We provide some theoretical analysis for GTOT-Tuning.

Algorithm Stability and Generalization Bound. We analyze the generalization bound of GTOT-Tuning and expect to find the key factors that affect its generalization ability. We first give the uniform stability below.

Lemma 1 (Uniform stability for GTOT-Tuning)**.** *Let* $S := \{z_1 = (G_1, y_1), z_2 = (G_2, y_2), \cdots, z_{i-1} = (G_{i-1}, y_{i-1}), z_i = (G_i, y_i), z_{i+1} = (G_{i+1}, y_{i+1}), \cdots, z_N = (G_N, y_N)\}$ *be a training set with* N *graphs,* $S^i := \{G_1, G_2, ..., G_{i-1}, G'_i, G_{i+1}, ..., G_N\}$ *be the training set where graph* i *has been replaced. Assume that the number of vertices* $|\mathcal{V}_{G_j}| \leq B$ *for all* j *and* $0 \leq \phi(f_S, z) \leq M$*, then*

$$|l(f_S, z) - l(f_{S^i}, z)| \leq 2M + \lambda\sqrt{B} \tag{7}$$

where λ *is the hyper-parameter used in Eq.* (6)*.*

Based on Lemma 1 and following the conclusion of [Bousquet and Elisseeff, 2002], the generalization error bound of GTOT-Tuning is obtained as follows.

Proposition 2. *Assume that a GNN with GTOT regularization satisfies* $0 \leq l(f_S, z) \leq \mathcal{Q}$*. For any* $\delta \in (0, 1)$*, the following bound holds with probability at least* $1 - \delta$ *over the random draw of the sample* S*,*

$$\begin{aligned} R(f_S) \leq & R_m(f_S) + 4M + 2\lambda\sqrt{B} \\ & + (8NM + 4N\lambda\sqrt{B} + \mathcal{Q})\sqrt{\frac{\ln 1/\delta}{2N}} \end{aligned} \tag{8}$$

where $R(f_S)$ denotes the generalization error and $R_m(f_S)$ represents empirical error. Proof is deferred to Appendix. This result shows that the generalization bound of GNN with GTOT regularizer is affected by the maximum number of vertices (B) in the training dataset.

7 Empirical Studies

We conduct experiments on graph classification tasks to evaluate our method.

7.1 Comparison of Various Fine-Tuning Strategies

We compare our method with other Fine-Tuning Strategies as follows.

Settings. We reuse two pretrained models released by [Hu *et al.*, 2020] as backbones: GIN (contextpred) [Xu *et al.*, 2018], which is only pretrained via self-supervised task *Context Prediction*, and GIN (supervised_contextpred), an architecture that is pretrained by *Context Prediction* + *Graph Level* multi-task supervised strategy. Both networks are pre-trained on the Chemistry dataset (with 2 million molecules). In addition, eight binary classification datasets in MoleculeNet [Wu *et al.*, 2018] serve as downstream tasks for evaluating the fine-tuning strategies, where the scaffold split scheme is used for dataset split. More details can be found in Appendix.

Baselines. Since we have not found related works about fine-tuning GNNs, we extend several typical baselines tailored for Convolutional networks to GNNs, including L2_SP, DELTA, BSS, SotchNorm.

Results. The results with different fine-tuning strategies are shown in Table 1. Observation (1): GTOT-Tuning gains a competitive performance on different datasets and outperforms other methods on average. Observation (2): Weights regularization (L2_SP) can't obtain improvement on pure self-supervised tasks. This implies L2_SP may require the pretrained task to be similar to the downstream task. Fortunately, our method can consistently boost the performance of

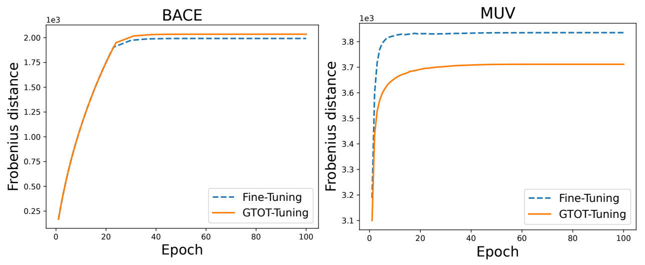

Figure 4: Weights distance between pre-trained initialization weights and fine-tuned weights.

both supervised and self-supervised pretrained models. Observation (3): The performance of the Euclidean distance regularization (Features(DELTA w/o ATT)) is worse than vanilla fine-tuning, indicating that directly using the node representation regularization may cause negative transfer.

7.2 Ablation Studies

Effects of the Mask Matrix. We conduct the experiments on GNNs fine-tuning with GTOT regularizer to verify the efficiency of the introduced adjacency matrix. The results shown in Table 2 suggest that when using adjacency matrix as mask matrix, the performance on most downstream tasks will be better than using classic WD directly. Besides, the competitive performance when the mask matrix is identity also implies that we can select possible pre-designed mask matrices, such as polynomials of $\mathbf{A}$, for specific downstream tasks. This also indicates that our MOT can be flexibly used for fine-tuning. (MWD(A) is GTOT distance.)

Effects of Different Proportion of Labeled Data. We also investigate the performance of the proposed method with different proportions of labeled data on MUV and SIDER datasets. As illustrated in Fig. 3, MWD outperforms the baseline (vanilla Fine-Tuning) given different amounts of labeled data consistently, indicating the generalization of our method.

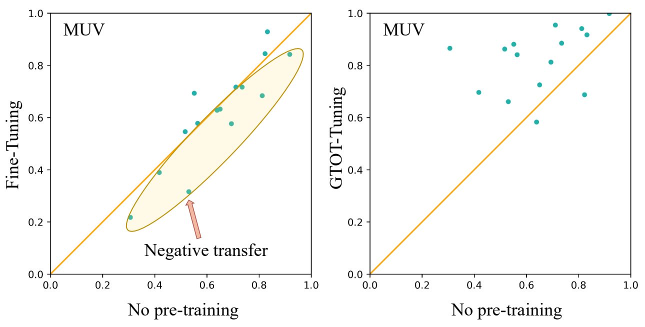

Figure 5: Scatter plot comparisons of ROC-AUC scores for a pair of fine-tuning strategies on the multi-task datasets (MUV). Each point represents a particular downstream task. There are many downstream tasks where vanilla fine-tuning performs worse than non-pre-trained model, indicating negative transfer. Meanwhile, negative transfer is alleviated when the GTOT regularizer is used.

Methods	BBBP	Tox21	Toxcast	SIDER
# $\mathcal{A}$_distance	1.57	1.56	1.56	1.41
w/o MWD	68.7±3.4	75.9±0.5	63.1±0.6	60.2±0.9
w/ MWD($\mathbf{1}_{n\times n}$)	66.2±3.3	75.3±0.9	63.6±0.6	62.7±0.7
w/ MWD($\mathbf{A}$)	**69.6±2.6**	**75.7±0.5**	63.8±0.4	63.5±0.6
w/ MWD($\mathbf{I}$)	68.6±3.5	75.4±0.7	**64.1±0.3**	**63.7±0.5**
Methods	ClinTox	MUV	HIV	BACE
# $\mathcal{A}$_distance	1.41	1.19	1.65	1.65
w/o MWD	69.5±5.0	69.5±1.3	78.2±1.2	82.5±1.7
w/ MWD($\mathbf{1}_{n\times n}$)	69.5±5.0	74.3±1.3	78.2±0.8	**83.7±1.9**
w/ MWD($\mathbf{A}$)	**70.9±5.8**	**80.7±0.6**	**78.5±1.5**	83.1±1.9
w/ MWD($\mathbf{I}$)	69.7±4.0	80.2±0.9	78.3±1.3	82.6±2.5

Table 2: Test ROC-AUC (%) of GIN(contextpred) on downstream tasks. (The parentheses indicate the masked matrix.)

7.3 Why GTOT-Tuning Works

Adaptive Adjustment of Model Weights to Downstream Tasks. The $\mathcal{A}$-distance [Ben-David *et al.*, 2007], $d_A(D^S, D^T) = 2(1 - 2\xi(h))$ is adopted to measure the discrepancy between the pretrained and finetuned domains, i.e. the domain gap between pre-trained data D^S and fine-tuned data D^T. $\xi(h)$ is the error of a linear SVM classifier h discriminating the two domains [Xu and et al., 2020]. We calculate the d_A with representations extracted by GIN (contextpred) as input and show the results in Table 2. From Fig. 4, it can be observed that GTOT-Tuning constrains the weights distance between fine-tuned and pre-trained model when domain gap is relatively small (MUV). Conversely, when the domain gap is large (BACE), our method does not necessarily increase the distance between weights, but rather increases the distance between weights. This reveals that GOT-Tuning can adaptively and implicitly adjust the distance between fine-tuned weights and pre-trained weights according to the downstream task, yielding powerful fine-tuned models.

Mitigating Negative Transfer under Multi-task. Fig. 5 shows that GTOT-Tuning boosts the performance of most tasks on multi-task datasets, demonstrating the ability of our method in mitigating negative transfer.

8 Discussions and Future Work

Despite the good results, there are some aspects which worth further explorations: i) The MOT-based methods require relatively high computational cost, which calls for designing a more efficient algorithm to solve it. ii) MWD has the potential to perform even better when used in conjunction with MGWD. iii) The proposed method can be potentially extended for more challenging settings that needs advanced knowledge transfer techniques, such as the graph out of distribution learning [Ji *et al.*, 2022].

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61972219), the Research and Development Program of Shenzhen (JCYJ20190813174403598, SGDX20190918101201696), the National Key Research and Development Program of China (2018YFB1800601), and the Overseas Research Cooperation Fund of Tsinghua Shenzhen International Graduate School (HW2021013). Besides, the authors would like to thank Ximei Wang, Guoji Fu and Guanzi Chen for their sincere and selfless help.

References

[Agrawal *et al.*, 2014] Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 2014.

[Bécigneul *et al.*, 2020] Gary Bécigneul, Octavian-Eugen Ganea, and et al. Optimal transport graph neural networks. *arXiv preprint arXiv:2006.04804*, 2020.

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *NeurIPS*, 2007.

[Bousquet and Elisseeff, 2002] Olivier Bousquet and André Elisseeff. Stability and generalization. *JMLR*, 2002.

[Chen and Gan, 2020] Liqun Chen and Zhe et al. Gan. Graph optimal transport for cross-domain alignment. In *ICML*, pages 1542–1553, 2020.

[Chen and Wang, 2021] Liqun Chen and Dong et al. Wang. Wasserstein contrastive representation distillation. In *CVPR*, pages 16296–16305, 2021.

[Chen and Zhang, 2019] Liqun Chen and Yizhe et al. Zhang. Improving sequence-to-sequence learning via optimal transport. In *ICLR*, 2019.

[Chen *et al.*, 2019] Xinyang Chen, Sinan Wang, and et al. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *NeurIPS*, 32:1908–1918, 2019.

[Courty *et al.*, 2016] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *TPAMI*, 39(9):1853–1865, 2016.

[Donahue and Jia, 2014] Jeff Donahue and Yangqing et al. Jia. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[Ganea *et al.*, 2022] Octavian-Eugen Ganea, Xinyuan Huang, and et al. Independent SE(3)-equivariant models for end-to-end rigid protein docking. In *ICLR*, 2022.

[He *et al.*, 2019] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, pages 4918–4927, 2019.

[Hu *et al.*, 2020] W Hu, B Liu, J Gomes, M Zitnik, P Liang, V Pande, and J Leskovec. Strategies for pre-training graph neural networks. In *ICLR (ICLR)*, 2020.

[Ji *et al.*, 2022] Yuanfeng Ji, Lu Zhang, and et al. DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery – A Focus on Affinity Prediction Problems with Noise Annotations. *arXiv e-prints*, page arXiv:2201.09637, January 2022.

[Kou *et al.*, 2020] Zhi Kou, Kaichao You, and et al. Stochastic normalization. *NeurIPS*, 33, 2020.

[Li *et al.*, 2018a] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018.

[Li *et al.*, 2018b] Xingjian Li, Haoyi Xiong, and et al. Delta: Deep learning transfer using feature map with attention for convolutional networks. In *ICLR*, 2018.

[Li *et al.*, 2020] Xuhong Li, Yves Grandvalet, Rémi Flamary, Nicolas Courty, and Dejing Dou. Representation transfer by optimal transport. *arXiv preprint arXiv:2007.06737*, 2020.

[Oquab *et al.*, 2014] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724, 2014.

[Peng *et al.*, 2020] Zhen Peng, Wenbing Huang, and et al. Graph representation learning via graphical mutual information maximization. In *The WebConf*, 2020.

[Peyré and Cuturi, 2019] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11 (5-6):355–602, 2019.

[Rong *et al.*, 2020] Yu Rong, Yatao Bian, and et al. Self-supervised graph transformer on large-scale molecular data. In *NeurIPS*, 2020.

[Wu *et al.*, 2018] Zhenqin Wu, Bharath Ramsundar, and et al. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[Xu and et al., 2020] Renjun Xu and et al. Reliable weighted optimal transport for unsupervised domain adaptation. In *CVPR*, pages 4394–4403, 2020.

[Xu and et al., 2021] Minghao Xu and et al. Self-supervised graph-level representation learning with local and global structure. *ICML*, 2021.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2018.

[Xu *et al.*, 2019] Hongteng Xu, Dixin Luo, and et al. Gromov-wasserstein learning for graph matching and node embedding. In *ICML*. PMLR, 2019.

[Xuhong *et al.*, 2018] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, 2018.

[Yosinski *et al.*, 2014] Jason Yosinski, Jeff Clune, and et al. How transferable are features in deep neural networks? *NeurIPS*, 27:3320–3328, 2014.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *NeurIPS*, 2020.

[Zhang *et al.*, 2022] Jiying Zhang, Fuyang Li, Xi Xiao, Tingyang Xu, Yu Rong, Junzhou Huang, and Yatao Bian. Hypergraph convolutional networks via equivalency between hypergraphs and undirected graphs. *ICML TAG-ML Workshop*, 2022.