

Domain Adaptation on Graphs by Learning Graph Topologies: Theoretical Analysis and an Algorithm

Elif Vural^{1*}

¹Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey

Abstract:

Traditional machine learning algorithms assume that the training and test data have the same distribution, while this assumption can be easily violated in real applications. Learning by taking into account the changes in the data distribution is called domain adaptation. In this work, we treat the domain adaptation problem in a graph setting. We consider a source and a target data graph that are constructed with samples drawn from a source and a target data manifold. We study the problem of estimating the unknown labels on the target graph by employing the label information in the source graph and the similarity between the two graphs. We particularly focus on a setting where the target label function is learnt such that its spectrum (frequency content when regarded as a graph signal) is similar to that of the source label function. We first present an overview of the recent field of graph signal processing and introduce concepts such as the Fourier transform on graphs. We then propose a theoretical analysis of domain adaptation over graphs, and present performance bounds relating the target classification error to the properties of the graph topologies and the manifold geometries. Finally, we propose a graph domain adaptation algorithm inspired by our theoretical findings, which estimates the label functions while learning the source and target graph topologies at the same time. Experiments on synthetic and real data sets suggest that the proposed method outperforms baseline approaches.

Key words: Domain adaptation, data classification, graph Fourier basis, graph Laplacian, performance bounds.

1. Introduction

Classical machine learning methods are based on the assumption that the training and test data have the same distribution. A classifier is learnt on the training data, which is then applied to the test data to estimate unknown class labels. Domain adaptation methods, on the other hand, focus on settings where the distribution of the test data is different from that of the training data [1], [2], [3], [4]. Given many labeled samples in a source domain and much fewer labeled samples in a target domain, the purpose of domain adaptation is to exploit the information available in both domains in order to improve the performance of classification in the target domain. Meanwhile, many machine learning problems nowadays involve inference on graph domains, such as in social and communication networks. Moreover, in problems where the data is observed in a physical ambient space, data samples often conform to a low-dimensional manifold model, e.g., the face images of a person captured from different viewpoints lie on a low-dimensional manifold. In such problems, graphs models are widely used as they provide very convenient tools for approximating the actual data manifold. Hence, the development of domain adaptation methods for graph-modeled data arises as an interesting problem. In this study, we consider

*Correspondence: velif@metu.edu.tr. This work has been supported by the TÜBİTAK 2232 program, project no. 117C007.

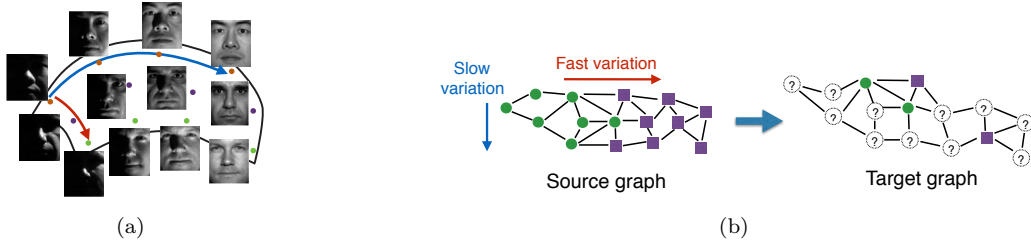


Figure 1. (a) Illustration of a face manifold where the label function has different speeds of variation along different directions (b) Illustration of domain adaptation on graphs

the problem of domain adaptation on graphs. We first present performance bounds for transferring knowledge between a pair of graphs. We then use these theoretical results to develop an algorithm that learns appropriate graph topologies and estimates class labels in a joint manner.

Many graph-based learning methods are based on the prior that the label function has a slow variation on the data graph. However, this does not need to hold in the general sense as illustrated in Figure 1(a). A face manifold is depicted in Figure 1(a). Each manifold point corresponds to a face image here, belonging to one of the three different subjects. In this example, the class label function has a slow variation along the blue direction, where the samples belong to the same subject. On the other hand, along the red direction the face images of different subjects get critically close to each other due to the extreme illumination conditions. Hence, the label function has a fast variation along the red direction on the manifold.

Although it is common practice to assume that the label function varies slowly in problems with a single graph, in a problem with more than one graph it is possible to learn the speed of variation of the label function and transfer this information across different graphs. This is illustrated in Figure 1(b), where the characteristics of the speed of variation of the label function can be learnt on a source graph with many available class labels. The purpose of graph domain adaptation is then to transfer this information to a target graph with very few labels in order to estimate the unknown class labels, by assuming that the target and source graphs have similar structures. We have studied this problem in our previous work [5], where we proposed a method called Spectral Domain Adaptation (SDA) for domain adaptation on graphs. The SDA method learns the spectrum, i.e., the frequency content, of the label function on the source graph, and then transfers the spectrum information to the target graph in order to improve the performance of classification over the target graph.

In order to achieve the best possible performance with graph domain adaptation methods, it is important to theoretically study their performance. The performance of graph-based domain adaptation significantly depends on the source and target graph structures and their resemblance. In particular, in problems where the graphs are constructed from data collections sampled from data manifolds, graph properties such as the edge locations and weights, and the number of neighbors of graph nodes largely influence the performance of learning. A thorough characterization of the effect of such factors in conjunction with the geometry of the data manifolds is necessary to understand the performance limits of graph domain adaptation techniques.

Our contribution in this study is twofold. We first propose a theoretical study of domain adaptation on graphs. We consider a source graph and a target graph constructed from samples obtained from a source manifold and a target manifold, defined via functions from a parameter domain to the ambient space. We theoretically analyze the performance of classification on the target graph. In particular, we focus on the estimation error of the target label function, i.e., the difference between the estimated and the true target label functions, and analyze how this estimation error varies with the graph properties, the sampling density of data, and the geometric properties of the data manifolds. Our theoretical analysis suggests that the construction of

very sparse graphs with too few edges or too dense graphs with too many edges should be avoided, as well as too small edge weights. The smoothness of the label functions is shown to positively influence the performance of learning. We show under certain assumptions that the estimation error of the target label function decreases with the sampling density of the manifolds at a rate of $O(N^{-1/d})$, where N is the number of samples and d is the intrinsic dimension of the manifolds. Next, we use these theoretical findings in order to propose a graph domain adaptation algorithm that jointly estimates the class labels of the source and target data together with a pair of source and target graph topologies that are consistent with the estimated class labels. In particular, we optimize the source and target graph weight matrices, which fully characterize the graph topologies, and the label functions so as to control factors such as the number of neighbors, minimum edge weights, and the smoothness of the label functions on the graphs. The experimental results on synthetic and real data sets show that the proposed method with learnt graph topologies outperforms reference domain adaptation methods with fixed graph topologies as well as some other baseline algorithms.

The rest of the paper is organized as follows. In Section 2, we briefly overview the related literature. In Section 3, we first overview frequency analysis on graphs and then present our theoretical bounds for graph domain adaptation. In Section 4, we present a graph domain adaptation algorithm that is motivated by our theoretical findings. We experimentally evaluate the proposed method in Section 5, and conclude in Section 6.

2. Related Work

We first overview some common solution approaches for domain adaptation. Several previous works have studied the covariate shift problem, where the two distributions with the same conditional distributions of class labels are matched with reweighting [6], [7]. The studies in [8], [9] propose to train a common classifier after mapping the data to a higher dimensional domain via feature augmentation. Another common approach is to align or match the two domains by mapping them to a common domain with projections or transformations [10], [11], [12], [13], [14], [15], [16], [17].

Several domain adaptation methods model data with a graph and make use of the assumption that the data varies smoothly on the graph [3], [18], [19]. The algorithms in [20] and [21] aim to compute a pair of bases on the source and target graphs that approximate the Fourier bases and jointly diagonalize the two graph Laplacians, which is applied to problems such as clustering and 3D shape analysis. Our recent work [5] relies also on representations with graph Fourier bases, however, in the context of domain adaptation. Our study in this paper provides perspective for the performance of such graph-based methods, whenever notions such as the smoothness of functions on graphs and representations with graph bases are relevant.

Some previous studies analyzing the domain adaptation problem from a theoretical perspective are the following. Performance bounds for importance reweighting have been proposed in [6] and [22]. The studies in [23], [24], [25], [26], and [27] bound the target loss in terms of the deviation between the source and the target distributions. While such studies propose a theoretical analysis of domain adaptation, none of them treat the domain adaptation problem in a graph setting. To the best of our knowledge, our theoretical analysis is the first to focus particularly on graph domain adaptation.

3. Theoretical Analysis of Graph Domain Adaptation

3.1. Overview of Signal Processing on Graphs

We first briefly overview some basic concepts regarding spectral graph theory and signal processing on graphs [28], [29]. Let $G = (V, E)$ be a graph consisting of N vertices denoted by $V = \{x_i\}_{i=1}^N$ and edges E . The

$N \times N$ symmetric matrix W consisting of nonnegative edge weights is called the weight matrix, where W_{ij} is the weight of the edge between the nodes x_i and x_j . If there is no edge between x_i and x_j , then $W_{ij} = 0$. The degree $d_i = \sum_{j=1}^N W_{ij}$ of a vertex x_i is defined as the total weight of the edges linked to it. Then, the diagonal matrix D with $D_{ii} = d_i$ is called the degree matrix.

The graph Laplacian matrix defined as $L = D - W$ is very important in spectral graph theory [28], [29]. A graph signal $f : V \rightarrow \mathbb{R}$ is a function taking a real value on each vertex. Hence, a graph signal f on a graph with N vertices can simply be regarded as an N -dimensional vector as $f = [f(x_1) \dots f(x_N)]^T \in \mathbb{R}^N$. The graph Laplacian can then be seen as an operator acting on the function f via the matrix multiplication Lf . It has been shown that the graph Laplacian L is in fact the graph equivalent of the Laplace operator in the Euclidean domain, or the Laplace-Beltrami operator on manifold domains [28], [30], [31].

Let us now understand why the graph Laplacian matrix is important for frequency analysis on graphs. First, recall that the complex exponentials (or simply, sinusoids) fundamental in classical signal processing have the special property that they are the eigenfunctions of the Laplacian operator. This can be easily seen by observing that the Laplacian operator Δ is equivalent to the second derivative $\Delta f = \partial^2 f / \partial t^2$ for a one-dimensional signal $f(t)$, and then checking that the complex exponentials $f(t) = e^{j\Omega t}$ satisfy the eigenfunction property $\Delta f = \lambda f$ since $\Delta(e^{j\Omega t}) = -\Omega^2 e^{j\Omega t}$. This observation is critical as it allows the extension of the notion of frequency to graph domains: Relying on the analogy between the Laplacian operator Δ and the graph Laplacian L , one can define a Fourier basis on graphs, consisting simply of the eigenvectors of L .

Let u_1, \dots, u_N denote the eigenvectors of the graph Laplacian, so that $Lu_k = \lambda_k u_k$ for $k = 1, \dots, N$. Then each u_k is a graph Fourier basis vector with frequency λ_k . The eigenvector u_1 with the smallest eigenvalue $\lambda_1 = 0$ is always a constant function on the graph, and the speed of variation of u_k on the graph increases for increasing k . The eigenvalues $\lambda_1, \dots, \lambda_N$ of the graph Laplacian correspond to frequencies such that λ_k gives a measure of the speed of variation of the signal u_k on the graph. Some Fourier basis vectors on an example graph are illustrated in Figure 2. In particular, the speed of variation of a signal f over the graph is given by

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^N W_{ij} (f(x_i) - f(x_j))^2,$$

which takes larger values if the function f varies more abruptly between neighboring graph vertices. Notice that the above term becomes the corresponding eigenvalue λ_k of L when f is taken as u_k , since $u_k^T L u_k = \lambda_k$.

This definition of a graph Fourier basis allows the extension of the Fourier transform to graph domains as follows. Let $U = [u_1 u_2 \dots u_N] \in \mathbb{R}^{N \times N}$ be the matrix consisting of the graph Fourier basis vectors. Then, for a graph signal $f \in \mathbb{R}^N$, the Fourier transform of f can simply be obtained as $\alpha = U^T f$, where $\alpha = [\alpha_1 \dots \alpha_N]^T$ is the vector consisting of the Fourier coefficients, with the k -th Fourier coefficient $\alpha_k = u_k^T f$ given by the inner product of f and the Fourier basis vector u_k . Note that the graph Fourier basis U is orthonormal as in classical signal processing; hence, the signal f can be reconstructed from the Fourier coefficients as $f = U\alpha$.

3.2. Notation and Setting

We now discuss the problem of domain adaptation on graphs and set the notation used in this paper. We consider a source graph $G^s = (V^s, E^s)$ with vertices $V^s = \{x_i^s\}_{i=1}^{N_s}$ and edges E^s ; and a target graph $G^t = (V^t, E^t)$ with vertices $V^t = \{x_i^t\}_{i=1}^{N_t}$ and edges E^t . Let W^s and W^t denote the weight matrices, and L^s and L^t the

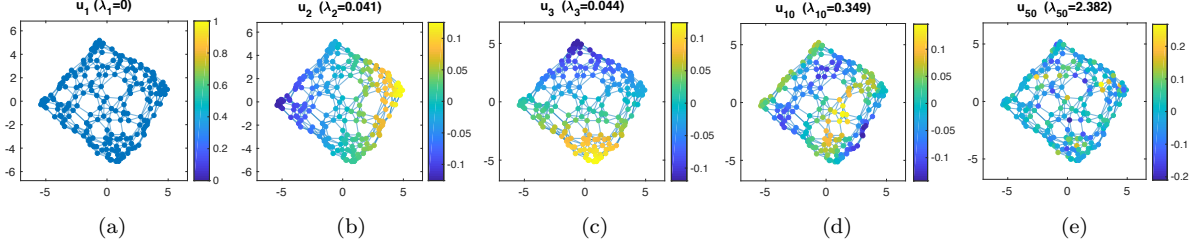


Figure 2. Fourier basis vectors of an example graph. The eigenvectors $u_1, u_2, u_3, u_{10}, u_{50}$ of the graph Laplacian are plotted as graph signals in panels (a)-(e), where yellow and dark blue tones respectively indicate positive and negative values. The first Fourier basis vector u_1 has constant amplitude as its frequency is $\lambda_1 = 0$. The signals u_2 and u_3 have small frequencies around 0.04 and they slowly oscillate along different directions on the graph. The signal u_{10} has a larger frequency $\lambda_{10} = 0.349$, hence its speed of oscillation is higher. Among the five signals, u_{50} has the highest frequency $\lambda_{50} = 2.382$ and it has the fastest variation on the graph.

Laplacians of the source and target graphs. Let f^s and f^t be the label functions on the source and target graphs, representing class labels in a classification problem and continuously varying entities in a regression problem. We assume that the labels are known at some vertices as $y_i^s = f^s(x_i^s)$ with $i \in I_L^s \subset \{1, \dots, N_s\}$, and $y_i^t = f^t(x_i^t)$ with $i \in I_L^t \subset \{1, \dots, N_t\}$ for the index sets I_L^s, I_L^t . One often has many labeled samples in the source domain and much fewer labeled samples in the target domain, i.e., $|I_L^t| \ll |I_L^s|$. Given the available labels $\{y_i^s\}_{i \in I_L^s}, \{y_i^t\}_{i \in I_L^t}$, the purpose of graph domain adaptation is to compute accurate estimates \hat{f}^s, \hat{f}^t of f^s and f^t .

All domain adaptation methods rely on the existence of a certain relationship between the source and the target domains. In this study, we consider a setting where a relationship can be established between the source and the target graphs via the frequency content of the label functions. Let $f^s = U^s \alpha^s$ and $f^t = U^t \alpha^t$ denote the decompositions of the label functions over the source Fourier basis $U^s = [u_1^s \dots u_{N_s}^s] \in \mathbb{R}^{N_s \times N_s}$ and the target Fourier basis $U^t = [u_1^t \dots u_{N_t}^t] \in \mathbb{R}^{N_t \times N_t}$. We then assume a setting where the source and target label functions have similar spectra, hence, similar Fourier coefficients.

We have observed in [5] that, when computing the label function estimates \hat{f}^s and \hat{f}^t , it is useful to represent them over the reduced bases $\bar{U}^s \in \mathbb{R}^{N_s \times R}$ and $\bar{U}^t \in \mathbb{R}^{N_t \times R}$, which consist of the first R Fourier basis vectors with smallest frequencies. This not only reduces the complexity of the problem but also has a regularization effect since very high-frequency components are excluded from the estimates. The estimates of the label functions are then obtained in the form $\hat{f}^s = \bar{U}^s \bar{\alpha}^s, \hat{f}^t = \bar{U}^t \bar{\alpha}^t$ where $\bar{\alpha}^s \in \mathbb{R}^R$ and $\bar{\alpha}^t \in \mathbb{R}^R$ are reduced Fourier coefficient vectors. The idea in [5] is to estimate the label functions such that their Fourier coefficients $\bar{\alpha}^s$ and $\bar{\alpha}^t$ are close to each other and the estimates \hat{f}^s, \hat{f}^t are consistent with the available labels.

In our theoretical analysis of graph domain adaptation, we consider a setting where graphs nodes are sampled from data manifolds. Let $\{x_i^s\}_{i=1}^{N_s} \subset \mathcal{M}^s$ and $\{x_i^t\}_{i=1}^{N_t} \subset \mathcal{M}^t$ denote the source and target graph nodes sampled from a source data manifold \mathcal{M}^s and a target data manifold \mathcal{M}^t . We assume that the source and target data manifolds are defined via a pair of functions $g^s : \Gamma \rightarrow \mathcal{M}^s$ and $g^t : \Gamma \rightarrow \mathcal{M}^t$ defined on a common parameter space Γ . Each manifold sample can then be expressed as $x_i^s = g^s(\gamma_i^s)$ and $x_i^t = g^t(\gamma_i^t)$, where $\gamma_i^s \in \Gamma$ and $\gamma_i^t \in \Gamma$ are the parameter vectors as illustrated in Figure 3. The parameter vectors are assumed to capture the source of variation generating the data manifolds. For instance, in a face recognition problem, γ_i^s and

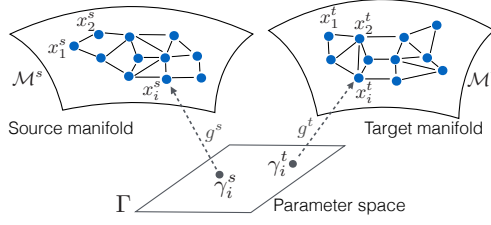


Figure 3. Illustration of the domain adaptation setting considered in our study

γ_i^t may represent the rotation angles of the camera viewing the subjects; and the difference between \mathcal{M}^s and \mathcal{M}^t may result from the change in the illumination conditions. Note that in domain adaptation no relation is assumed to be known between γ_i^s and γ_i^t . Moreover, the parameters $\{\gamma_i^s\}$, $\{\gamma_i^t\}$ and the functions g^s , g^t are often not known in practice. Although we employ these concepts in our theoretical analysis for convenience, the practical algorithm we propose in Section 4 will not require the knowledge of these parameters.

3.3. Performance Bounds for Graph Domain Adaptation

In this section, we analyze the error between the estimated target label function \hat{f}^t and the true target label function f^t . We thus would like to derive an upper bound for the estimation error

$$E = \|\hat{f}^t - f^t\|^2 = \sum_{i=1}^{N_t} (\hat{f}^t(x_i^t) - f^t(x_i^t))^2 = \sum_{i=1}^{N_t} (\hat{f}_i^t - f_i^t)^2$$

where $\hat{f}_i^t = \hat{f}^t(x_i^t)$ and $f_i^t = f^t(x_i^t)$ simply denote the estimated and the true label functions at the sample x_i^t .

We first define some parameters regarding the properties of the data manifolds. For the convenience of analysis, we assume that the source and target graphs contain equally many samples¹, i.e., $N_s = N_t = N$. We assume that the source and target manifolds $\mathcal{M}^s \subset \mathcal{H}^s$, $\mathcal{M}^t \subset \mathcal{H}^t$ are embedded in some Hilbert spaces \mathcal{H}^s , \mathcal{H}^t ; the parameter space Γ is a Banach space, and the manifolds \mathcal{M}^s and \mathcal{M}^t have (intrinsic) dimension d .

We assume that the functions $g^s : \Gamma \rightarrow \mathcal{M}^s$, $g^t : \Gamma \rightarrow \mathcal{M}^t$ are Lipschitz-continuous, respectively with constants M_s , M_t , so that for any two parameter vectors $\gamma_1, \gamma_2 \in \Gamma$, we have

$$\|g^s(\gamma_1) - g^s(\gamma_2)\| \leq M_s \|\gamma_1 - \gamma_2\|, \quad \|g^t(\gamma_1) - g^t(\gamma_2)\| \leq M_t \|\gamma_1 - \gamma_2\|$$

where $\|\cdot\|$ denotes the usual norm in the space of interest. The constants M_s and M_t thus provide a smoothness measure for the manifolds \mathcal{M}^s and \mathcal{M}^t . We further assume that there exist two constants A_l , A_u such that for any $\gamma_1 \neq \gamma_2$ in Γ ,

$$A_l \leq \frac{\|g^t(\gamma_1) - g^t(\gamma_2)\|}{\|g^s(\gamma_1) - g^s(\gamma_2)\|} \leq A_u. \quad (1)$$

The constants A_l , A_u indicate the similarity between the geometry of the two manifolds \mathcal{M}^s , \mathcal{M}^t : As the variations of the manifold-generating functions g^s , g^t over Γ get more similar, the constants A_l and A_u get closer to 1. Let $A = \max(|1 - A_l|, |A_u - 1|)$ denote a bound on the deviations of A_l and A_u from 1.

¹In fact, this does not only simplify the analysis, but also suggests a condition towards improving the similarity between the graph Laplacians. In practice, one can always match the number of samples between the two graphs by sample removal or generation.

We consider a setting where the graph weight matrices are obtained via a kernel ϕ such that $W_{ij}^s = \phi(\|x_i^s - x_j^s\|)$ and $W_{ij}^t = \phi(\|x_i^t - x_j^t\|)$ for neighboring samples on the graph. We assume that the kernel $\phi : \mathbb{R}^+ \cup \{0\} \rightarrow \mathbb{R}^+$ is an L_ϕ -Lipschitz nonincreasing function with $|\phi(u) - \phi(v)| \leq L_\phi|u - v|$ for any $u, v \in \mathbb{R}^+ \cup \{0\}$. Let us denote the maximum value of the kernel function as $\phi_0 := \phi(0)$.

Next, we define some parameters related to the properties of the target graph. First, recall from Section 3.2 that we consider a setting where very few labels $y_i^t = f^t(x_i^t)$ are known on the target graph, with indices $i \in I_L^t \subset \{1, \dots, N_t\}$ in the index set I_L^t . In our analysis, we consider domain adaptation algorithms that compute an estimate \hat{f}^t of f^t whose values at the labeled nodes agree with the given labels, i.e., $\hat{f}_i^t = f_i^t$ for $i \in I_L^t$. Now let us denote the set of indices of the unlabeled nodes of the target graph as $I_U^t = I^t \setminus I_L^t$, where $I^t = \{1, \dots, N_t\}$ denotes the set of all target node indices and $\cdot \setminus \cdot$ denotes the set difference. Moreover, assuming that the target graph is connected, or otherwise each connected component contains at least one labeled node, one can partition the node indices as $I^t = I_L^t \cup I_{U_1}^t \cup \dots \cup I_{U_Q}^t$, where $I_{U_q}^t$ consists of the indices of unlabeled nodes that are connected to the nearest labeled node through a shortest path of length q (with q hops). Q is then the longest possible length of the shortest path between an unlabeled and a labeled node. Let $I_{U_0}^t := I_L^t$ simply stand for the index set of labeled nodes. We thus partition the target graph nodes in sets of several “layers” $q = 0, 1, \dots, Q$, with respect to their proximity to the closest labeled node.

Let $\mathcal{N}_j^q := \{i \in I_{U_q}^t : x_i^t \sim x_j^t\}$ denote the set of indices of the neighbors of a node x_j^t within the nodes of layer q , where \sim denotes neighboring nodes. We can define the parameters $K_q^{\min} := \min_{j \in I_{U_q}^t} |\mathcal{N}_j^{q-1}|$ and $K_q^{\max} := \max_{j \in I_{U_q}^t} |\mathcal{N}_j^{q+1}|$, where $|\cdot|$ denotes the cardinality. K_q^{\min} is thus the minimum number of neighbors between a layer q and its preceding layer $q - 1$, and K_q^{\max} is the maximum number of neighbors between a layer q and its succeeding layer $q + 1$. Let us also define the minimum weight of an edge between any two nodes from consecutive layers as $w^{\min} := \min_{q=1, \dots, Q} w_q^{\min}$, where $w_q^{\min} = \min\{W_{ij}^t : x_i^t \sim x_j^t, i \in I_{U_q}^t, j \in I_{U_{q-1}}^t\}$ is the smallest edge weight between layers q and $q - 1$.

We are now ready to state our main result in the following theorem.

Theorem 1 *Consider a graph domain adaptation algorithm estimating source and target label functions $\hat{f}^s = \bar{U}^s \bar{\alpha}^s$ and $\hat{f}^t = \bar{U}^t \bar{\alpha}^t$ such that the difference between their Fourier coefficients is bounded as $\|\bar{\alpha}^s - \bar{\alpha}^t\| \leq \Delta_\alpha$, the norms of the Fourier coefficients are bounded as $\|\bar{\alpha}^s\|, \|\bar{\alpha}^t\| \leq C$, and \hat{f}^s and \hat{f}^t are band-limited on the graphs so as not to contain any components with frequencies larger than λ_R . Assume that the estimate \hat{f}^s is equal to the true source label function f^s (e.g. in a setting where all source samples are labeled).*

Then, the target label estimation error can be upper bounded as

$$\|\hat{f}^t - f^t\|^2 \leq \frac{\kappa}{w^{\min}} (\sqrt{B} + \sqrt{\hat{B}})^2 \quad (2)$$

where B is an upper bound on the rate of variation of the true label function

$$(f^t)^T L^t f^t \leq B, \quad (3)$$

the parameter κ is defined as

$$\kappa := \sum_{q=1}^Q \frac{|I_{U_q}^t|}{K_q^{\min}} \left(1 + \sum_{l=1}^{q-1} \prod_{m=l}^{q-1} |I_{U_m}^t| \frac{K_m^{\max}}{K_m^{\min}} \right), \quad (4)$$

\hat{B} is an upper bound on the speed of variation of the target label estimate given by

$$(\hat{f}^t)^T L^t \hat{f}^t \leq \hat{B} := (f^s)^T L^s f^s + C^2 \rho_{\max} + 2C \lambda_R \Delta_\alpha, \quad (5)$$

ρ_{\max} is a geometry-dependent parameter varying at rate

$$\rho_{\max} := O(L_\phi (A M_s + M_s + M_t) \epsilon_\Gamma + \phi_0), \quad (6)$$

and ϵ_Γ is proportional to the largest parameter-domain distance between neighboring graph nodes.

A more precise statement of Theorem 1 is provided in Appendix A along with its proof.

Theorem 1 can be interpreted as follows. First, observe that the estimation error increases linearly with the bound Δ_α on the deviation between the source and target Fourier coefficients. This suggests that it is favorable to estimate source and target label functions with similar spectra in graph domain adaptation. The theorem also has several implications regarding the smoothness of label functions and their estimates. It is common knowledge that graph-based learning methods perform better if label functions vary smoothly on the graph. This is formalized in Theorem 1 via the assumption that the estimates \hat{f}^s and \hat{f}^t are band-limited so that the highest frequency present in their spectrum (computed via the graph Fourier transform) does not exceed some threshold λ_R , which limits their speed of variation on the graphs. Notice also that the rates of variation $(f^s)^T L^s f^s$, $(f^t)^T L^t f^t$ of the true source and target label functions affect the estimation error (via the terms B and \hat{B}), which is in line with the common intuition.

Next, we observe from (2) that the estimation error depends on the geometric properties of data manifolds as follows: The error increases linearly with ρ_{\max} (via the term \hat{B}), while in (6), ρ_{\max} is seen to depend linearly on the parameters A , M_s , and M_t . Recalling that M_s and M_t are the Lipschitz constants of the functions g^s , g^t defining the data manifolds, the theorem suggests that the estimation error will be smaller when the data manifolds are smoother. The fact that the error increases linearly with A is intuitive as the parameter A captures the dissimilarity between the geometries of the source and target manifolds. We also notice that ρ_{\max} is proportional to the parameter ϵ_Γ . We give a precise definition of the parameter ϵ_Γ in Appendix A, which can be roughly described as a parameter that upper bounds the parameter-domain distance (measured in Γ) between neighboring samples on the graphs. As the number of samples N increases, ϵ_Γ decreases at rate $\epsilon_\Gamma = O(N^{-1/d})$ with N , where d is the intrinsic dimension of the manifolds. Since ρ_{\max} is linearly proportional to ϵ_Γ , we conclude that the estimation error of the target label function decreases with N at the same rate $O(N^{-1/d})$. This can be intuitively interpreted with the reasoning that, when the data manifolds are sampled more densely, the difference between the source and target graph topologies due to finite sampling effects is reduced.

The result in Theorem 1 also leads to the following important conclusions about the effect of the graph properties on the performance of learning. First, the estimation error is observed to increase linearly with the

parameter-domain distance ϵ_Γ between neighboring points on the graphs. This suggests that when constructing the graphs, two samples too distant from each other in the parameter space should rather not be connected with an edge. Then, we notice that the parameter κ decreases when the ratio K_m^{\max}/K_m^{\min} between the maximum and minimum number of neighbors is smaller. Hence, more “balanced” graph topologies influence the performance positively; more precisely, the number of neighbors of different graph nodes should not be disproportionate. At the same time, the term K_q^{\min} in the denominator in (4) implies that nodes with too few neighbors should rather be avoided. A similar observation can be made about the term w^{\min} in the expression of the error bound in (2). The minimal edge weight term w^{\min} in the denominator suggests that graph edges with too small weights have the tendency to increase the error. From all these observations, we draw the conclusion that when constructing graphs, rather balanced graph topologies should be preferred, without significant differences between the number of neighbors of different nodes, too isolated nodes, and too weak edges.

4. Learning Graph Topologies for Domain Adaptation

In this section, we propose an algorithm for jointly learning graph topologies and label functions for domain adaptation, based on the theoretical findings presented in Section 3. We first formulate our graph domain adaptation problem and then propose a method for solving it.

4.1. Problem Formulation

Given the source and target samples $\{x_i^s\}_{i=1}^{N_s} \subset \mathbb{R}^n$ and $\{x_i^t\}_{i=1}^{N_t} \subset \mathbb{R}^n$, we consider the problem of constructing a source and a target graph with respective vertices $\{x_i^s\}$ and $\{x_i^t\}$, while obtaining estimates $\hat{f}^s = \bar{U}^s \bar{\alpha}^s$ and $\hat{f}^t = \bar{U}^t \bar{\alpha}^t$ of the label functions at the same time. The problem of learning the graph topologies is equivalent to the problem of learning the weight matrices W^s and W^t .

The bound (2) on the target error suggests that when learning a pair of graphs, the parameters κ , B and \hat{B} should be kept small, whereas small values for w^{\min} should be avoided. The expression in (5) shows that the parameters λ_R and Δ_α should be kept small. Meanwhile, in the expression of ρ_{\max} in (6), we observe that the terms A , M_s , and M_t are dictated by the geometry of the data manifolds and are independent of the constructed graphs. On the other hand, the parameter ϵ_Γ depends on the graph topology and can be controlled more easily. Thus, in view of the interpretation of Theorem 1, we propose to learn the label functions \hat{f}^s , \hat{f}^t along with the weight matrices W^s and W^t based on the following optimization problem.

$$\begin{aligned}
& \text{minimize}_{\bar{\alpha}^s, \bar{\alpha}^t, W^s, W^t} \|S^s \bar{U}^s \bar{\alpha}^s - y^s\|^2 + \|S^t \bar{U}^t \bar{\alpha}^t - y^t\|^2 + \mu \|\bar{\alpha}^s - \bar{\alpha}^t\|^2 \\
& \quad + (\hat{f}^s)^T L^s \hat{f}^s + (\hat{f}^t)^T L^t \hat{f}^t + \mu_s \sum_{i,j=1}^{N_s} W_{ij}^s \|x_i^s - x_j^s\|^2 + \mu_t \sum_{i,j=1}^{N_t} W_{ij}^t \|x_i^t - x_j^t\|^2 \\
& \text{subject to } W_{ij}^s \geq W^{\min}, \forall i, j \in \{1, \dots, N_s\} \text{ with } W_{ij}^s \neq 0; \quad W_{ij}^t \geq W^{\min}, \forall i, j \in \{1, \dots, N_t\} \text{ with } W_{ij}^t \neq 0; \\
& \quad d_{\min} \leq d_i^s \leq d_{\max}, \forall i \in \{1, \dots, N_s\}; \quad d_{\min} \leq d_i^t \leq d_{\max}, \forall i \in \{1, \dots, N_t\}.
\end{aligned} \tag{7}$$

Here μ , μ_s , and μ_t are positive weight parameters, and y^s and y^t are vectors consisting of all the available source and target labels. The first two terms $\|S^s \bar{U}^s \bar{\alpha}^s - y^s\|^2$ and $\|S^t \bar{U}^t \bar{\alpha}^t - y^t\|^2$ in (7) enforce the estimated

label functions \hat{f}^s and \hat{f}^t to be consistent with the available labels via the binary selection matrices S^s and S^t consisting of 0's and 1's to properly select the indices of labeled data. The third term $\|\bar{\alpha}^s - \bar{\alpha}^t\|^2$ aims to reduce the parameter Δ_α in (5). Note that choosing a representation of \hat{f}^s and \hat{f}^t in terms of the first R Fourier basis vectors in \bar{U}^s and \bar{U}^t serves to keep the parameter λ_R small.

Next, the minimization of the terms $(\hat{f}^s)^T L^s \hat{f}^s$ and $(\hat{f}^t)^T L^t \hat{f}^t$ imposes the label functions \hat{f}^s and \hat{f}^t to have a slow variation on the graphs, which aims to reduce the parameters \hat{B} and B in (5) and (3). Then, remember from Theorem 1 that in order to make the parameter ϵ_Γ small, graph edges between distant points should be avoided. The terms $\sum_{i,j} W_{ij}^s \|x_i^s - x_j^s\|^2$ and $\sum_{i,j} W_{ij}^t \|x_i^t - x_j^t\|^2$ aim to achieve this by penalizing large edge weights between distant samples. The inequality constraints $W_{ij}^s \geq W^{\min}$ and $W_{ij}^t \geq W^{\min}$ on nonzero edge weights aim to ensure that the minimum edge weight w^{\min} is above some predetermined threshold W^{\min} .

Finally, we recall from Theorem 1 that in order to minimize κ , the ratio K_m^{\max}/K_m^{\min} must be kept small while avoiding too small K_q^{\min} values. However, incorporating the number of neighbors directly in the objective function would lead to an intractable optimization problem. Noticing that the node degrees are typically expected to be proportional to the number of neighbors, we prefer to relax this to the constraints $d_{\min} \leq d_i^s \leq d_{\max}$ and $d_{\min} \leq d_i^t \leq d_{\max}$ on the node degrees, where d_i^s and d_i^t respectively denote the degrees of x_i^s and x_i^t ; and d_{\min} and d_{\max} are some predefined degree threshold parameters with $0 < d_{\min} \leq d_{\max}$.

4.2. Proposed Method: Domain Adaptive Graph Learning

Analyzing the optimization problem in (7), we observe that the matrices \bar{U}^s and \bar{U}^t are nonconvex and highly nonlinear functions of the optimization variables W^s and W^t as they consist of the eigenvectors of the graph Laplacians L^s and L^t . Moreover, due to the multiplicative terms such as $\bar{U}^s \bar{\alpha}^s$, the problem is even not jointly convex in $\bar{\alpha}^s$, $\bar{\alpha}^t$, and \bar{U}^s , \bar{U}^t . Hence, it is quite difficult to solve the problem (7). In our method, we propose a heuristic iterative solution approach that relaxes the ideal problem (7) into more tractable subproblems and alternatively updates the coefficients and the weight matrices in each iteration as follows.

We first initialize the weight matrices W^s , W^t with a typical strategy; e.g., by connecting each node to its K nearest neighbors and assigning edge weights with a Gaussian kernel. We use the normalized versions of the graph Laplacians given by $L^s = (D^s)^{-1/2}(D^s - W^s)(D^s)^{-1/2}$ and $L^t = (D^t)^{-1/2}(D^t - W^t)(D^t)^{-1/2}$.

In the first step of each iteration, we optimize $\bar{\alpha}^s$, $\bar{\alpha}^t$ by fixing the weight matrices W^s , W^t . This gives the following optimization problem:

$$\text{minimize}_{\bar{\alpha}^s, \bar{\alpha}^t} \|S^s \bar{U}^s \bar{\alpha}^s - y^s\|^2 + \|S^t \bar{U}^t \bar{\alpha}^t - y^t\|^2 + \mu \|\bar{\alpha}^s - \bar{\alpha}^t\|^2. \quad (8)$$

The simplified objective² in (8) is in fact the same as the objective of the SDA algorithm proposed in [5]. As the problem is quadratic and convex in $\bar{\alpha}^s$ and $\bar{\alpha}^t$, its solution can be analytically found by setting the gradient as equal to 0, which gives [5]:

$$\bar{\alpha}^s = (\mu^{-1} A^t A^s + A^t + A^s)^{-1} (\mu^{-1} A^t B^s y^s + B^s y^s + B^t y^t), \quad \bar{\alpha}^t = (\mu^{-1} A^s \bar{\alpha}^s + \bar{\alpha}^s - \mu^{-1} B^s y^s)$$

²Note that the dependence of \hat{f}^s and \hat{f}^t on $\bar{\alpha}^s$ and $\bar{\alpha}^t$ is neglected in (8). The reason is that since \bar{U}^s and \bar{U}^t consist of the eigenvectors of L^s and L^t , the terms $(\hat{f}^s)^T L^s \hat{f}^s$ and $(\hat{f}^t)^T L^t \hat{f}^t$ would contribute to the objective only as regularization terms on the weighted norms of $\bar{\alpha}^s$ and $\bar{\alpha}^t$. We prefer to exclude such a regularization in order to prioritize fitting the coefficients $\bar{\alpha}^s$, $\bar{\alpha}^t$ to each other and to the available labels.

Algorithm 1 Spectral Domain Adaptation via Domain Adaptive Graph Learning (SDA-DAGL)

- 1: **Input:**
 $\{x_i^s\}, \{x_i^t\}$: Source and target samples
 y^s, y^t : Available source and target labels
 - 2: **Initialization:**
Initialize the weight matrices W^s, W^t with a sufficiently large number of edges, e.g., as K-NN graphs.
 - 3: **repeat**
 - 4: Compute the graph Laplacians L^s, L^t and Fourier bases \bar{U}^s, \bar{U}^t with weight matrices W^s, W^t .
 - 5: Update coefficients $\bar{\alpha}^s, \bar{\alpha}^t$ by solving (8).
 - 6: Update the weight matrices W^s, W^t by solving (9) via linear programming.
 - 7: Prune the graph edges by setting the edge weights with $W_{ij}^s < W^{\min}$ and $W_{ij}^t < W^{\min}$ to 0.
 - 8: **until** the maximum number of iterations is attained
 - 9: **Output:**
 $f^t = \bar{U}^t \bar{\alpha}^t$: Estimated target label function
 $f^s = \bar{U}^s \bar{\alpha}^s$: Estimated source label function
-

where $A^s = (\bar{U}^s)^T (S^s)^T S^s \bar{U}^s$, $B^s = (\bar{U}^s)^T (S^s)^T$, $A^t = (\bar{U}^t)^T (S^t)^T S^t \bar{U}^t$, $B^t = (\bar{U}^t)^T (S^t)^T$.

Then, in the second step of an iteration, we fix the coefficients $\bar{\alpha}^s, \bar{\alpha}^t$ and optimize the weight matrices W^s and W^t . As the dependence of the Fourier basis matrices \bar{U}^s and \bar{U}^t on W^s and W^t is quite intricate, we fix \bar{U}^s and \bar{U}^t to their values from the preceding iteration and neglect this dependence when reformulating our objective for learning the weight matrices. Defining the vectors $\hat{h}^s = (D^s)^{-1/2} \hat{f}^s$ and $\hat{h}^t = (D^t)^{-1/2} \hat{f}^t$, the fourth and fifth terms in (7) can be rewritten as

$$(\hat{f}^s)^T L^s \hat{f}^s + (\hat{f}^t)^T L^t \hat{f}^t = (\hat{h}^s)^T (D^s - W^s) \hat{h}^s + (\hat{h}^t)^T (D^t - W^t) \hat{h}^t.$$

If the node degrees are fixed, the minimization of the above term corresponds to the maximization of $(\hat{h}^s)^T W^s \hat{h}^s + (\hat{h}^t)^T W^t \hat{h}^t$. However, we have observed that letting the node degrees vary in an appropriate interval gives better results than strictly fixing them. We thus propose the following objective for optimizing W^s and W^t .

$$\begin{aligned} \text{minimize}_{W^s, W^t} \quad & \mu_s \sum_{i,j=1}^{N_s} W_{ij}^s \|x_i^s - x_j^s\|^2 - (\hat{h}^s)^T W^s \hat{h}^s + \mu_t \sum_{i,j=1}^{N_t} W_{ij}^t \|x_i^t - x_j^t\|^2 - (\hat{h}^t)^T W^t \hat{h}^t \\ \text{subject to} \quad & d_{\min} \leq \sum_{j=1}^{N_s} W_{ij}^s \leq d_{\max}, \forall i = 1, \dots, N_s; \quad d_{\min} \leq \sum_{j=1}^{N_t} W_{ij}^t \leq d_{\max}, \forall i = 1, \dots, N_t. \end{aligned} \tag{9}$$

The objective function and the constraints of the problem (9) are linear in the entries of W^s and W^t . Hence, (9) can be posed as a linear programming (LP) problem and can be solved with an LP solver. Notice, however, that the constraints $W_{ij}^s \geq W^{\min}$, $W_{ij}^t \geq W^{\min}$ in the original problem (7) are excluded from the subproblem in (9). This is for the following reason: While the problem (7) imposes the nonzero edge weights to be larger than a threshold and attempts to control the number of neighbors via degree constraints, it does not strictly impose any structure in the graph topology such as the sparsity of the weight matrices. Meanwhile, we have experimentally observed that the solution to (9) often tends to decrease the number of nonzero edge weights, i.e., gradually improve the sparsity of the weight matrices. Hence, instead of directly incorporating the sparsity of the weight matrices in the optimization problem and imposing a lower bound on positive edge weights, we prefer to initialize the graphs with a sufficiently high number of edges, solve the LP problem (9) by optimizing only the nonzero edge weights, and then at the end of each iteration apply a graph pruning step so that the edge weights smaller than W^{\min} are set to 0. After each iteration, the graph Laplacians L^s, L^t and the Fourier

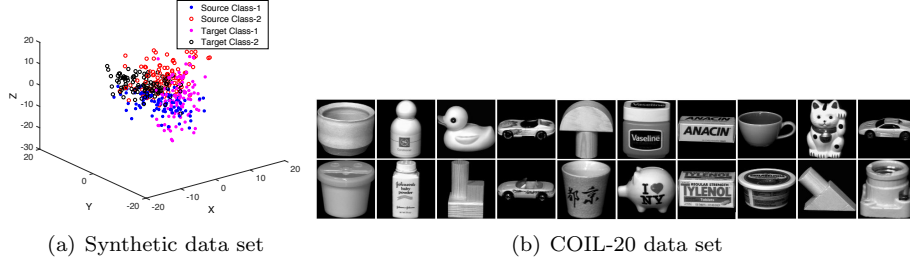


Figure 4. (a) Synthetic data set with two classes drawn from normal distributions. (b) Sample images from the COIL-20 data set. Each source domain object in the upper row is assigned the same class label as the matching target domain object right below it.

bases \bar{U}^s , \bar{U}^t are updated and the same procedure continues with the optimization of the Fourier coefficients as in (8). We call this algorithm Spectral Domain Adaptation via Domain Adaptive Graph Learning (SDA-DAGL) and summarize it in Algorithm 1. Due to the various relaxations made in the employed subproblems, it is not possible to guarantee the convergence of the solution in the general sense. In practice, we have found it useful to terminate the algorithm when a suitably set number of iterations is attained.

5. Experimental Results

We now evaluate the proposed method with experiments on a synthetic and a real data set. The synthetic data set shown in Figure 4(a) consists of two classes with a total of 400 normally distributed samples in \mathbb{R}^3 . The two classes in each domain have different means, and the source and target domains differ by a rotation of 90° . The variance of the normal distributions is chosen relatively large for increasing the difficulty of the problem. The COIL-20 object database [32] shown in Figure 4(b) consists of a total of 1440 images of 20 objects. Each object has 72 images taken from different viewpoints rotating around it. We downsample the images to a resolution of 32×32 pixels. The 20 objects in the data set are divided into two groups and each object in the first group is matched to the object in the second group with the highest similarity to it. Each group of 10 objects is taken as a different domain and the matched object pairs are considered to have the same class label in the experiments.

We first compare our SDA-DAGL algorithm with the SDA algorithm in order to study the efficiency of the proposed domain adaptive graph learning approach. In both data sets, we first independently construct the source and target graphs by connecting each sample to its K nearest neighbors and form weight matrices W^s and W^t with a Gaussian kernel. The SDA algorithm uses the fixed graph topology imposed by these weight matrices. In the proposed SDA-DAGL method, these weight matrices are used to initialize the algorithm as in Step 2 of Algorithm 1, and then they are refined gradually with the proposed joint graph learning and label estimation framework. All class labels are known in the source domain, whereas a small number of labels are known in the target domain. The class labels of the unlabeled target samples are estimated with the two algorithms in comparison. Figure 5 shows the variation of the misclassification rate of target samples with the number of graph neighbors (K) for different numbers of labeled samples (N) in the target domain. The results are averaged over around 10 random repetitions of the experiment with different selections of labeled samples.

The results in Figure 5 show that the SDA-DAGL algorithm with the proposed graph learning strategy performs better than the SDA algorithm with the fixed graph in almost all cases. This suggests that even if the SDA-DAGL algorithm is initialized with nonoptimal edge weights, it can successfully learn suitable source and target graph topologies together with the edge weights and accurately estimate the target labels. The performance of SDA-DAGL is seen to be not much sensitive to the choice of the initial number of neighbors K

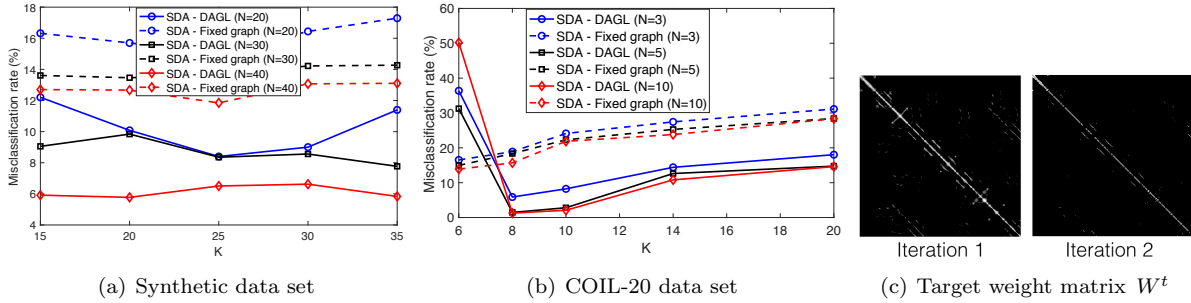


Figure 5. (a), (b) Variation of the misclassification rates of target samples with the number of nearest neighbors K in graphs. The curves with dashed lines are obtained with a fixed graph topology (SDA), whereas the corresponding curves with solid lines are obtained with the proposed graph learning method (SDA-DAGL). (c) Evolution of W^t during two consecutive iterations. Black color represents 0 weight (no edge) and brighter tones indicate larger edge weights.

in the synthetic data set in Figure 5(a), whereas it is more affected by the choice of K in the COIL-20 data set in Figure 5(b). This is because the COIL-20 data set conforms quite well to a low-dimensional manifold structure due to the regularly sampled camera rotation generating the data set. Initializing the weight matrices with too high K values leads to the loss of the information of the geometric structure from the beginning and makes it more difficult for the algorithm to recover the correct graph topologies along with the label estimates. The fact that too small K values also yield large error in Figure 5(b) can be explained with the incompatibility of this choice with the graph pruning strategy employed in our method. We also show in Figure 5(c) the evolution of the weight matrix W^t during two consecutive iterations of the SDA-DAGL method for the COIL-20 data set. Data samples are ordered with respect to their classes, hence, ideally one would like a block-diagonal weight matrix with only within-class edges. The update on W^t is seen to remove some of the between-class edges observable in the off-block-diagonal entries, thus, the learnt graph topology is progressively improved.

We finally present an overall comparison of the proposed SDA-DAGL method with some baseline domain adaptation methods in the literature representing different approaches. We compare our method to the Easy Adapt ++ (EA++) [8] algorithm based on feature augmentation; the Domain Adaptation using Manifold Alignment (DAMA) [10] algorithm which is a graph-based method learning a supervised embedding; and the Geodesic Flow Kernel (GFK) [11] and Subspace Alignment (SA) [12] methods, which align the PCA bases of the two domains via unsupervised projections. The misclassification rates of the algorithms on target samples are plotted with respect to the ratio of known target labels in Figures 6(a) and 6(b), respectively for the synthetic and the COIL-20 data sets. The misclassification error decreases as the ratio of known target labels increases as expected. The proposed SDA-DAGL algorithm is observed to often outperform the baseline approaches and the SDA method using a fixed graph topology in both data sets. This suggests that the proposed graph learning strategy provides an effective solution for improving the performance of domain adaptation on graphs.

6. Conclusion

In this paper, we have studied the problem of domain adaptation on graphs both from a theoretical and a methodological perspective. We have first proposed a theoretical analysis of the performance of graph domain adaptation methods. We have considered a setting where a pair of graphs are constructed from data samples drawn from a source manifold and a target manifold. We have focused on a graph domain adaptation framework where the source and target label functions are estimated such that they have similar spectrum when regarded as graph signals. We have proposed an upper bound on the estimation error of the target label function and

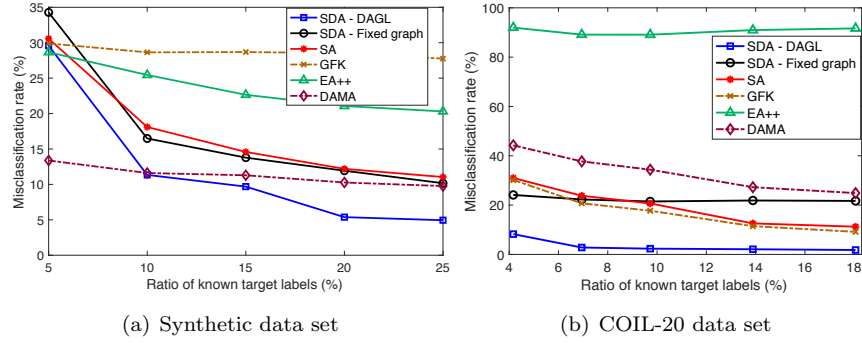


Figure 6. The variation of the misclassification rate of the target samples with the ratio of labeled target samples

studied its dependence on the number of data samples, the geometric properties of the data manifolds, and graph parameters such as edge weights and the number of neighbors of graph nodes. In particular, as far as the graph properties are concerned, our theoretical results suggest that a “balanced” graph topology improves the performance of learning where the numbers of neighbors are proportionate across different nodes, and too weak edge weights as well as edges between too distant samples are avoided. Based on these theoretical insights, we have then proposed a graph domain adaptation algorithm that learns the source and target graphs while jointly estimating the source and target label functions. Experimental results on synthetic and real data sets suggest that the proposed method yields promising performance in problems concerning learning on graph domains.

References

- [1] M. Rohrbach, S. Ebert, and B. Schiele, “Transfer learning in a transductive setting,” in *Proc. Advances in Neural Information Processing Systems*, 2013, pp. 46–54.
- [2] C. Wang and S. Mahadevan, “Manifold alignment using procrustes analysis,” in *Proc. 25th Int. Conf. Machine Learning*, 2008, pp. 1120–1127.
- [3] M. Xiao and Y. Guo, “Feature space independent semi-supervised domain adaptation via kernel matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54–66, 2015.
- [4] Z. Fang and Z. Zhang, “Discriminative transfer learning on manifold,” in *Proc. 13th SIAM Int. Conf. Data Mining*, 2013, pp. 539–547.
- [5] M. Pilancı and E. Vural, “Domain adaptation via transferring spectral properties of label functions on graphs,” in *IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop*, 2016, pp. 1–5.
- [6] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, “Correcting sample selection bias by unlabeled data,” in *Proc. Advances in Neural Information Processing Systems 19*, 2006, pp. 601–608.
- [7] S. Khalighi, B. Ribeiro, and U. Nunes, “Importance weighted import vector machine for unsupervised domain adaptation,” *IEEE Trans. Cybernetics*, vol. 47, no. 10, pp. 3280–3292, 2017.
- [8] H. Daumé, III, A. Kumar, and A. Saha, “Frustratingly easy semi-supervised domain adaptation,” in *Proc. 2010 Workshop on Domain Adaptation for Natural Language Processing*, 2010, pp. 53–59.
- [9] L. Duan, D. Xu, and I. W. Tsang, “Learning with augmented features for heterogeneous domain adaptation,” in *Proc. 29th International Conference on Machine Learning*, 2012.
- [10] C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in *Proc. 22nd Int. Joint Conf. on Artificial Intelligence*, 2011, pp. 1541–1546.

- [11] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.
- [12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in *Proc. IEEE International Conference on Computer Vision*, 2013, ICCV ’13, pp. 2960–2967.
- [13] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, “Domain adaptation with conditional transferable components,” in *Proc. 33rd International Conference on Machine Learning*, 2016, pp. 2839–2848.
- [14] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.
- [15] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, “Optimal transport for domain adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, 2017.
- [16] M. Jiang, W. Huang, Z. Huang, and G. G. Yen, “Integration of global and local metrics for domain adaptation learning via dimensionality reduction,” *IEEE Trans. Cybernetics*, vol. 47, no. 1, pp. 38–51, 2017.
- [17] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, “Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation,” in *IEEE Conf. CVPR*, 2017, pp. 945–954.
- [18] T. Yao, Y. Pan, C. Ngo, H. Li, and T. Mei, “Semi-supervised domain adaptation with subspace learning for visual recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2142–2150.
- [19] L. Cheng and S. J. Pan, “Semi-supervised domain adaptation on manifolds,” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 12, pp. 2240–2249, 2014.
- [20] D. Eynard, A. Kovnatsky, M. M. Bronstein, K. Glashoff, and A. M. Bronstein, “Multimodal manifold analysis by simultaneous diagonalization of Laplacians,” *IEEE Trans. PAMI.*, vol. 37, no. 12, pp. 2505–2517, 2015.
- [21] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, “Partial functional correspondence,” *Comput. Graph. Forum*, vol. 36, no. 1, pp. 222–236, 2017.
- [22] C. Cortes, Y. Mansour, and M. Mohri, “Learning bounds for importance weighting,” in *Proc. Advances in Neural Information Processing Systems 23*, 2010, pp. 442–450.
- [23] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in *Proc. Advances in Neural Information Processing Systems 19*, 2006, pp. 137–144.
- [24] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, “Learning bounds for domain adaptation,” in *Proc. Advances in Neural Information Processing Systems 20*, 2007, pp. 129–136.
- [25] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation: Learning bounds and algorithms,” in *The 22nd Conference on Learning Theory*, 2009.
- [26] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, “A theory of learning from different domains,” *Machine Learning*, vol. 79, no. 1-2, pp. 151–175, 2010.
- [27] C. Zhang, L. Zhang, and J. Ye, “Generalization bounds for domain adaptation,” *arXiv preprint*, 2013.
- [28] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.
- [29] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.
- [30] M. Hein, J. Y. Audibert, and U. von Luxburg, “From graphs to manifolds - weak and strong pointwise consistency of graph laplacians,” in *18th Annual Conference on Learning Theory, COLT*, 2005, pp. 470–485.
- [31] A. Singer, “From graph to manifold Laplacian: The convergence rate,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 128 – 134, 2006.
- [32] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia Object Image Library (COIL-20),” Tech. Rep., Feb 1996.
- [33] M. Pilanci and E. Vural, “Domain adaptation on graphs by learning aligned graph bases,” *arXiv preprint*, 2018, [Online]. Available: <https://arxiv.org/abs/1803.05288>.

A. Proof of Theorem 1

In order to prove Theorem 1, we proceed as follows. We first propose an upper bound on the difference between the rates of variation of the source and target label functions in Section A.1. Next, we study the deviation between the eigenvalues of the source and target graph Laplacians in Section A.2. Finally, we combine these results in Section A.3 and present our upper bound on the estimation error of the target label function to finalize the proof.

A.1. Analysis of the difference between the rates of variation of the source and target label functions

We propose in Lemma 1 an upper bound on the difference between the rates of variation $(\hat{f}^s)^T L^s \hat{f}^s$ and $(\hat{f}^t)^T L^t \hat{f}^t$ of the estimated source and label functions on the graphs.

Lemma 1 *Let $0 = \lambda_1^s \leq \lambda_2^s \leq \dots \leq \lambda_R^s$ and $0 = \lambda_1^t \leq \lambda_2^t \leq \dots \leq \lambda_R^t$ respectively denote the R smallest eigenvalues of the source and target graph Laplacians L^s and L^t . Assume that the deviation between the corresponding eigenvalues of the two graph Laplacians are bounded as $|\lambda_i^s - \lambda_i^t| \leq \delta$, for all $i = 1, \dots, R$. Let the bandwidth parameter $\lambda_R = \max(\lambda_R^s, \lambda_R^t)$ indicate an upper bound for the frequencies of the first R source and target Fourier basis vectors.*

In the estimates $\hat{f}^s = \bar{U}^s \bar{\alpha}^s$, $\hat{f}^t = \bar{U}^t \bar{\alpha}^t$ of the source and target label functions, let the difference between the Fourier coefficients of the label functions be bounded as $\|\bar{\alpha}^s - \bar{\alpha}^t\| \leq \Delta_\alpha$, and let C be a bound on the norms of the coefficients such that $\|\bar{\alpha}^s\|, \|\bar{\alpha}^t\| \leq C$.

Then the difference between the rates of variation of the source and target label function estimates on the graphs can be bounded as

$$|(\hat{f}^s)^T L^s \hat{f}^s - (\hat{f}^t)^T L^t \hat{f}^t| \leq C^2 \delta + 2C \lambda_R \Delta_\alpha.$$

The proof of Lemma 1 is an adaptation of the proof of [33, Proposition 1] to our case and it is given in Appendix B.1. Lemma 1 can be interpreted as follows: When the source and the target graphs are sufficiently similar, their graph Laplacians will have similar spectra, and the difference between their i -th eigenvalues can be bounded as $|\lambda_i^s - \lambda_i^t| \leq \delta$ for a relatively small constant δ . If in addition, the label functions vary sufficiently slowly to have limited bandwidth, then the source and target label functions have a similar rate of variation on the two graphs.

A.2. Analysis of the difference between the spectra of the source and target graphs

In Lemma 1 we have assumed that the source and target graph Laplacians have similar spectra, so that the difference $|\lambda_i^s - \lambda_i^t|$ between their eigenvalues can be suitably bounded. We now determine under which conditions this is possible. We aim to develop an upper bound on $|\lambda_i^s - \lambda_i^t|$ for all i in terms of the properties of the data manifolds and the constructed graphs.

In domain adaptation problems, a one-to-one correspondence between the source and target data samples is often not available, in contrast to multi-view or multi-modal learning problems. Hence, we do not assume that there exists a particular relation between the samples $x_i^s = g^s(\gamma_{s_i})$ and $x_i^t = g^t(\gamma_{t_i})$, such as being generated from the same parameter vector $\gamma_{s_i} = \gamma_{t_i}$ in the parameter space. Nevertheless, if the manifolds \mathcal{M}^s and

\mathcal{M}^t defined over the same parameter space Γ are sampled under similar conditions (although independently), one may assume that the independently formed sample sets $\{x_i^s\}$ and $\{x_i^t\}$ can be reordered so that their corresponding parameter vectors $\{\gamma_i^s\}$ and $\{\gamma_i^t\}$ “fall” into nearby regions of the parameter space Γ . We state this assumption with the help of a cover

$$\mathcal{C} = \bigcup_{m=1}^M B_{\epsilon_m}(\gamma_m) \subset \Gamma$$

in the parameter space, where $B_{\epsilon}(\gamma)$ denotes an open ball of radius ϵ around the parameter vector γ

$$B_{\epsilon}(\gamma) = \{\gamma' \in \Gamma : \|\gamma' - \gamma\| < \epsilon\}$$

and the parameter vector of any data sample is in at least one ball in \mathcal{C}

$$\{\gamma_i^s\} \cup \{\gamma_i^t\} \subset \mathcal{C} = \bigcup_{m=1}^M B_{\epsilon_m}(\gamma_m).$$

We assume without loss of generality that the source and target samples $\{x_i^s\}_{i=1}^N$ and $\{x_i^t\}_{i=1}^N$ are ordered such that for each $i = 1, \dots, N$,

$$\gamma_i^s \in B_{\epsilon_m}(\gamma_m), \quad \gamma_i^t \in B_{\epsilon_m}(\gamma_m)$$

the samples γ_i^s and γ_i^t are in the same ball $B_{\epsilon_m}(\gamma_m)$ for some $m \in \{1, \dots, M\}$. This condition imposes an ordering of the samples such that source and target samples with nearby indices correspond to nearby parameter vectors in the parameter space Γ .

Remarks: 1. First, let us note that when applying a graph-based domain adaptation algorithm, as the parameter-domain representation of the data samples is often unknown, it will not be possible to actually find such a reordering of the data. Nevertheless, our assumption of such a reordering is just for the purpose of theoretical analysis and is not needed in practice. This is because reordering the graph nodes will result in a permutation of the rows and the columns of the graph Laplacian. Since the permutations of the rows and columns with the same indices can be represented as the left and right multiplications of the graph Laplacian with the same symmetric rotation matrix, it will not change its eigenvalues. Hence, our analysis of the deviation $|\lambda_i^s - \lambda_i^t|$ between the eigenvalues under the reordering assumption is still valid even if the nodes are not reordered in practice.

2. Second, notice that for a finite sampling of the source and target data, it is always possible to find such covers as mentioned above. However, the radii $\epsilon_1, \dots, \epsilon_M$ of the open balls depend on the number of samples. In particular, the radii of these open sets decrease proportionally to the typical inter-sample distance between neighboring manifold samples as the number of samples N increases, i.e., $\epsilon_m = O(N^{-1/d})$ where d is the dimension of the manifolds $\mathcal{M}^s, \mathcal{M}^t$.

Next, we define some parameters regarding the properties of the graph. Let the source node x_i^s have K_i^s neighbors in the source graph G^s , where we denote $x_i^s \sim x_j^s$ when x_i^s and x_j^s are connected with an edge. Similarly let K_i^t denote the number of neighbors of x_i^t in the target graph. Among the K_i^s neighbors x_j^s of x_i^s in the source graph, some of their “correspondences” x_j^t (with respect to the ordering discussed above) will be

neighbors of the target node x_i^t corresponding to x_i^s . Let β_i^s be the proportion of such nodes, which is given by

$$\beta_i^s = \frac{1}{K_i^s} |\{j : x_j^s \sim x_i^s, x_j^t \sim x_i^t\}|$$

where $|\cdot|$ denotes the cardinality of a set. Similarly, let

$$\beta_i^t = \frac{1}{K_i^t} |\{j : x_j^t \sim x_i^t, x_j^s \sim x_i^s\}|.$$

The more similar the source and target graphs are, the closer the parameters $\beta_i^s \leq 1$ and $\beta_i^t \leq 1$ will be to 1.

Finally, we define the parameter $\epsilon_{\mathcal{N}}$ as

$$\epsilon_{\mathcal{N}} = \max \left(\max_{x_i^s \sim x_j^s} \|\gamma_i^s - \gamma_j^s\|, \max_{x_i^t \sim x_j^t} \|\gamma_i^t - \gamma_j^t\| \right)$$

which gives the largest possible parameter domain distance between two neighboring nodes in the source or target graphs. Also, due to our assumption on the ordering of the samples, for any i , there exists some $m \in \{1, \dots, M\}$ such that $\gamma_i^s, \gamma_i^t \in B_{\epsilon_m}(\gamma_m)$. Let

$$\epsilon_{m_i} = \min\{\epsilon_m : \gamma_i^s, \gamma_i^t \in B_{\epsilon_m}(\gamma_m)\}$$

and also let

$$\epsilon_{\Gamma} = \epsilon_{\mathcal{N}} + 2 \max_i \epsilon_{m_i}$$

represent a generic upper bound on the parameter domain distance between “within-domain” and “cross-domain” neighboring samples.

We are now ready to state our bound on the deviation $|\lambda_i^s - \lambda_i^t|$ between the corresponding eigenvalues of the source and target graph Laplacians in Lemma 2.

Lemma 2 *Let L^s and L^t be the Laplacian matrices of the source and target graphs with respective eigenvalues ordered as $0 = \lambda_1^s \leq \lambda_2^s \leq \dots \leq \lambda_N^s$, and $0 = \lambda_1^t \leq \lambda_2^t \leq \dots \leq \lambda_N^t$. Based on the above definitions of the graph parameters, let us denote*

$$\Delta_W := L_{\phi} (A M_s + M_s + M_t) \epsilon_{\Gamma}.$$

Then for all $i = 1, \dots, N$ the deviation $|\lambda_i^s - \lambda_i^t|$ between the corresponding eigenvalues of L^s and L^t is upper bounded as

$$|\lambda_i^s - \lambda_i^t| \leq \rho_{\max} := \max_{i=1, \dots, N} 2 \left(\beta_i^s K_i^s \Delta_W + (1 - \beta_i^s) K_i^s \phi_0 + (1 - \beta_i^t) K_i^t \phi_0 \right).$$

The proof of Lemma 2 is presented in Appendix B.2. Lemma 2 can be interpreted as follows. First, recall that the parameters β_i^s and β_i^t give the ratio of the source and target neighbors of a graph node that have correspondences in the other graph. Hence, they provide a measure of the similarity between the source and the target graphs. When the resemblance between the source and target graphs gets stronger, these parameters get closer to 1. If we consider the asymptotic case where β_i^s and β_i^t approach 1, the upper bound ρ_{\max} on the deviation between the eigenvalues becomes

$$\lim_{\beta_i^s \rightarrow 1, \beta_i^t \rightarrow 1} \rho_{\max} = \max_i 2 K_i^s \Delta_W.$$

In order to understand how the eigenvalue deviation changes with the sampling density of the graphs, we can study the variation of this term with the number N of graph nodes. Assuming that the number of neighbors K_i^s are of $O(1)$ with respect to N , i.e., if the number of neighbors is not increased proportionally to N in the graph construction, the rate of variation of ρ_{\max} with N is given by that of the term Δ_W . As the number N of data points sampled from the manifolds $\mathcal{M}^s, \mathcal{M}^t$ increases, we have

$$\epsilon_{m_i}, \epsilon_{\mathcal{N}} = O(N^{-1/d})$$

and consequently, $\epsilon_{\Gamma} = O(N^{-1/d})$, where d is the intrinsic dimension of the manifolds. This yields $\Delta_W = O(N^{-1/d})$, hence, we obtain

$$|\lambda_i^s - \lambda_i^t| \leq \rho_{\max} = O(N^{-1/d})$$

for the asymptotic case $\beta_i^s \rightarrow 1, \beta_i^t \rightarrow 1$.

Next, we can also interpret Lemma 2 from the perspective of the similarity between the manifolds. As the geometric structures of the source and target manifolds $\mathcal{M}^s, \mathcal{M}^t$ get more similar, the constants A_l and A_u will approach 1, and the constant A will approach 0. We observe that this reduces the deviation $|\lambda_i^s - \lambda_i^t|$ as Δ_W is proportional to A . Also, as the Lipschitz regularity of the functions g^s, g^t defining the source and target manifolds improves, the constants M_s and M_t will decrease, hence, Δ_W will also decrease. We can summarize these with the observation that as $M_s, M_t \rightarrow 0, A \rightarrow 0$, and $\beta_i^s, \beta_i^t \rightarrow 1$, the eigenvalue difference converges to 0 as $|\lambda_i^s - \lambda_i^t| \rightarrow 0$ for all $i = 1, \dots, N$.

A.3. Bounding the estimation error of the target label function

Putting together the results from Lemmas 1 and 2, we arrive at the following observation: Assuming that the conditions of Lemma 2 hold, the spectrum perturbation parameter δ in Lemma 1 can be upper bounded as $\delta \leq \rho_{\max}$. In this case, the rates of variations of the source and target function estimates differ as

$$|(\hat{f}^s)^T L^s \hat{f}^s - (\hat{f}^t)^T L^t \hat{f}^t| \leq C^2 \rho_{\max} + 2C \lambda_R \Delta_{\alpha}.$$

Due to our assumption $f^s = \hat{f}^s$, the above inequality implies

$$(\hat{f}^t)^T L^t \hat{f}^t \leq \hat{B} = (f^s)^T L^s f^s + C^2 \rho_{\max} + 2C \lambda_R \Delta_{\alpha}.$$

The parameter \hat{B} thus gives a bound on the rate of variation of the label function estimate \hat{f}^t on the graph. Recall also the assumption on the true target label function in Theorem 1

$$(f^t)^T L^t f^t \leq B.$$

Under these assumptions, we would like to find an upper bound on the target label estimation error $\|\hat{f}^t - f^t\|$.

Our derivation of an upper bound on the target label estimation error is based on the following decomposition

$$\|\hat{f}^t - f^t\|^2 = \sum_{q=0}^Q \|\hat{f}_{U_q}^t - f_{U_q}^t\|^2$$

where the vectors $f_{U_q}^t$ and $\hat{f}_{U_q}^t$ are respectively obtained by restricting the vectors f^t and \hat{f}^t to their entries in the index set $I_{U_q}^t$. Our strategy for bounding the target error is to bound the error of each layer in terms of

the error of the previous layer, and use the observation that the error of layer $q = 0$ is 0 as it consists of the labeled samples. In the following result, we first provide an upper bound on the error $\|\hat{f}_{U_q}^t - f_{U_q}^t\|^2$ of the q -th layer in terms of the error $\|\hat{f}_{U_{q-1}}^t - f_{U_{q-1}}^t\|^2$ of the preceding layer $q - 1$.

Lemma 3 *Let*

$$B_q := \sum_{j \in I_{U_q}^t} \sum_{x_i^t \sim x_j^t, i \in I_{U_{q-1}}^t} W_{ij}^t (f_i^t - f_j^t)^2$$

denote the total rate of variation of the true target label function f^t over the edges between two consecutive layers q and $q - 1$. Similarly define

$$\hat{B}_q := \sum_{j \in I_{U_q}^t} \sum_{x_i^t \sim x_j^t, i \in I_{U_{q-1}}^t} W_{ij}^t (\hat{f}_i^t - \hat{f}_j^t)^2$$

for the estimate \hat{f}^t of the target label function. We can then bound the estimation error of the q -th layer in terms of the estimation error of the preceding layer $q - 1$ as

$$\|\hat{f}_{U_q}^t - f_{U_q}^t\|^2 \leq \frac{|I_{U_q}^t|}{K_q^{\min}} \left(\frac{\sqrt{B_q} + \sqrt{\hat{B}_q}}{\sqrt{w_q^{\min}}} + \sqrt{K_{q-1}^{\max}} \|\hat{f}_{U_{q-1}}^t - f_{U_{q-1}}^t\| \right)^2$$

for $q = 1, \dots, Q$.

The proof of Lemma 3 is given in Appendix B.3.

Lemma 3 provides an upper bound on the error of each layer in terms of the error of the previous layer. We can now use this result to obtain an upper bound on the overall target label estimation error, which is presented in the following lemma.

Lemma 4 *The estimation error of the target label function can be bounded as*

$$\|f^t - \hat{f}^t\|^2 \leq \frac{\kappa}{w_{\min}} (\sqrt{B} + \sqrt{\hat{B}})^2$$

where

$$\kappa = \sum_{q=1}^Q \frac{|I_{U_q}^t|}{K_q^{\min}} \left(1 + \sum_{l=1}^{q-1} \prod_{m=l}^{q-1} |I_{U_m}^t| \frac{K_m^{\max}}{K_m^{\min}} \right).$$

The proof of Lemma 4 is given in Appendix B.4.

The results stated in Lemmas 1-4 provide a complete characterization of the performance of estimating the target label function in a graph domain adaptation setting. We are now ready to combine these results in the following main result.

Theorem 2 *Consider a graph-based domain adaptation algorithm matching the spectra of the source and target label functions $\hat{f}^s = \bar{U}^s \bar{\alpha}^s$ and $\hat{f}^t = \bar{U}^t \bar{\alpha}^t$ such that the difference between the Fourier coefficients of the label functions are bounded as $\|\bar{\alpha}^s - \bar{\alpha}^t\| \leq \Delta_\alpha$, the norms of the Fourier coefficients are bounded as $\|\bar{\alpha}^s\|, \|\bar{\alpha}^t\| \leq C$,*

and \hat{f}^s and \hat{f}^t are band-limited on the graphs so as not to contain any components with frequencies larger than λ_R .

Assume the source and target graphs are constructed independently from equally many data samples by setting the graph weights via the kernel ϕ , where the source samples $\{x_i^s\}_{i=1}^N$ and target samples $\{x_i^t\}_{i=1}^N$ are drawn from the manifolds \mathcal{M}^s and \mathcal{M}^t defined via the functions g^s and g^t over a common parameter domain Γ . Let

$$\Delta_W = L_\phi (A M_s + M_s + M_t) \in \Gamma. \quad (10)$$

and

$$\rho_{\max} = \max_{i=1, \dots, N} 2 (\beta_i^s K_i^s \Delta_W + (1 - \beta_i^s) K_i^s \phi_0 + (1 - \beta_i^t) K_i^t \phi_0). \quad (11)$$

Then, the difference between the rates of variation of the source and target function estimates is upper bounded as

$$|(\hat{f}^s)^T L^s \hat{f}^s - (\hat{f}^t)^T L^t \hat{f}^t| \leq C^2 \rho_{\max} + 2C \lambda_R \Delta_\alpha.$$

Moreover, if $\hat{f}^s = f^s$ and if the target label function estimates \hat{f}_i^t are equal to the true labels f_i^t at labeled nodes, the target label estimation error can be bounded as

$$\|\hat{f}^t - f^t\|^2 \leq \frac{\kappa}{w_{\min}} (\sqrt{B} + \sqrt{\hat{B}})^2 \quad (12)$$

where

$$\hat{B} = (f^s)^T L^s f^s + C^2 \rho_{\max} + 2C \lambda_R \Delta_\alpha, \quad (13)$$

B is an upper bound on the rate of variation of the true label function

$$(f^t)^T L^t f^t \leq B, \quad (14)$$

and

$$\kappa = \sum_{q=1}^Q \frac{|I_{U_q}^t|}{K_q^{\min}} \left(1 + \sum_{l=1}^{q-1} \prod_{m=l}^{q-1} |I_{U_m}^t| \frac{K_m^{\max}}{K_m^{\min}} \right).$$

We finally conclude the proof of Theorem 1 by observing that it is simply a summarizing restatement of the result in Theorem 2.

B. Proofs of the Lemmas in Appendix A

B.1. Proof of Lemma 1

Proof The rates of variation of \hat{f}^s and \hat{f}^t on the source and target graphs are given by

$$\begin{aligned} (\hat{f}^s)^T L^s \hat{f}^s &= (\bar{\alpha}^s)^T (\bar{U}^s)^T L^s \bar{U}^s \bar{\alpha}^s = (\bar{\alpha}^s)^T \Lambda^s \bar{\alpha}^s \\ (\hat{f}^t)^T L^t \hat{f}^t &= (\bar{\alpha}^t)^T (\bar{U}^t)^T L^t \bar{U}^t \bar{\alpha}^t = (\bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t \end{aligned}$$

where Λ^s and Λ^t are the diagonal matrices consisting of the R smallest eigenvalues of respectively L^s and L^t , such that $\Lambda_{ii}^s = \lambda_i^s$ and $\Lambda_{ii}^t = \lambda_i^t$, for $i = 1, \dots, R$.

The difference between the rates of variations of \hat{f}^s and \hat{f}^t can then be bounded as

$$\begin{aligned}
|(\hat{f}^s)^T L^s \hat{f}^s - (\hat{f}^t)^T L^t \hat{f}^t| &= |(\bar{\alpha}^s)^T \Lambda^s \bar{\alpha}^s - (\bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t| \\
&= |(\bar{\alpha}^s)^T \Lambda^s \bar{\alpha}^s - (\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^s + (\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^s - (\bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t| \\
&\leq |(\bar{\alpha}^s)^T (\Lambda^s - \Lambda^t) \bar{\alpha}^s| + |(\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^s - (\bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t|.
\end{aligned} \tag{15}$$

In the following, we derive an upper bound for each one of the two terms at the right hand side of the inequality in (15). The first term is bounded as

$$|(\bar{\alpha}^s)^T (\Lambda^s - \Lambda^t) \bar{\alpha}^s| \leq \|\bar{\alpha}^s\|^2 \|\Lambda^s - \Lambda^t\| \leq C^2 \delta.$$

Here the first inequality is due to the Cauchy-Schwarz inequality, and the second inequality follows from the fact that the operator norm of the diagonal matrix $\Lambda^s - \Lambda^t$ is given by the magnitude of its largest eigenvalue, which cannot exceed δ due to the assumption $|\lambda_i^s - \lambda_i^t| \leq \delta$ for all i .

Next, we bound the second term in (15) as

$$\begin{aligned}
|(\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^s - (\bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t| &= |(\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^s - (\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^t + (\bar{\alpha}^s)^T \Lambda^t \bar{\alpha}^t - (\bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t| \\
&\leq |(\bar{\alpha}^s)^T \Lambda^t (\bar{\alpha}^s - \bar{\alpha}^t)| + |(\bar{\alpha}^s - \bar{\alpha}^t)^T \Lambda^t \bar{\alpha}^t| \\
&\leq \|\bar{\alpha}^s\| \|\Lambda^t\| \|\bar{\alpha}^s - \bar{\alpha}^t\| + \|\bar{\alpha}^s - \bar{\alpha}^t\| \|\Lambda^t\| \|\bar{\alpha}^t\| \leq 2C\lambda_R \Delta_\alpha
\end{aligned}$$

where the last equality follows from the fact that the matrix norm $\|\Lambda^t\|$ is given by the largest eigenvalue of Λ^t , which is smaller than λ_R by our assumption.

Putting together the upper bounds for both terms in (15), we get the stated result

$$|(\hat{f}^s)^T L^s \hat{f}^s - (\hat{f}^t)^T L^t \hat{f}^t| \leq C^2 \delta + 2C\lambda_R \Delta_\alpha.$$

□

B.2. Proof of Lemma 2

Proof In order to show that the stated upper bound holds on the difference between the eigenvalues, we first examine the difference $|W_{ij}^s - W_{ij}^t|$ between the corresponding entries of the source and target weight matrices. We propose an upper bound on $|W_{ij}^s - W_{ij}^t|$ for three different cases below where at least one of W_{ij}^s and W_{ij}^t is nonzero.

Case 1. When the source samples $x_i^s \sim x_j^s$ are neighbors on the source graph and the corresponding target samples $x_i^t \sim x_j^t$ are also neighbors on the target graph at the same time, we bound $|W_{ij}^s - W_{ij}^t|$ as

$$|W_{ij}^s - W_{ij}^t| = |\phi(\|x_i^s - x_j^s\|) - \phi(\|x_i^t - x_j^t\|)| \leq L_\phi \|\|x_i^s - x_j^s\| - \|x_i^t - x_j^t\|\|. \tag{16}$$

We proceed by examining each one of the terms $\|x_i^s - x_j^s\|$ and $\|x_i^t - x_j^t\|$. We have

$$\begin{aligned}
\|x_i^s - x_j^s\| &= \|g^s(\gamma_i^s) - g^s(\gamma_j^s)\| \\
&= \|g^s(\gamma_i^s) - g^s(\gamma_{m_i}) + g^s(\gamma_{m_i}) - g^s(\gamma_{m_j}) + g^s(\gamma_{m_j}) - g^s(\gamma_j^s)\|
\end{aligned}$$

which implies

$$\begin{aligned} \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| - \Delta_i^s - \Delta_j^s &\leq \|x_i^s - x_j^s\| \\ &\leq \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| + \Delta_i^s + \Delta_j^s \end{aligned} \quad (17)$$

where

$$\Delta_i^s := \|g^s(\gamma_i^s) - g^s(\gamma_{m_i})\|, \quad \Delta_j^s := \|g^s(\gamma_j^s) - g^s(\gamma_{m_j})\|.$$

With a similar derivation, we get

$$\begin{aligned} \|g^t(\gamma_{m_i}) - g^t(\gamma_{m_j})\| - \Delta_i^t - \Delta_j^t &\leq \|x_i^t - x_j^t\| \\ &\leq \|g^t(\gamma_{m_i}) - g^t(\gamma_{m_j})\| + \Delta_i^t + \Delta_j^t \end{aligned} \quad (18)$$

where

$$\Delta_i^t := \|g^t(\gamma_i^t) - g^t(\gamma_{m_i})\|, \quad \Delta_j^t := \|g^t(\gamma_j^t) - g^t(\gamma_{m_j})\|.$$

From (17) and (18), we get

$$\begin{aligned} \left| \|x_i^s - x_j^s\| - \|x_i^t - x_j^t\| \right| &\leq \left| \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| - \|g^t(\gamma_{m_i}) - g^t(\gamma_{m_j})\| \right| \\ &\quad + \Delta_i^s + \Delta_j^s + \Delta_i^t + \Delta_j^t. \end{aligned} \quad (19)$$

From the definition (1) of A_l and A_u , we have

$$A_l \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| \leq \|g^t(\gamma_{m_i}) - g^t(\gamma_{m_j})\| \leq A_u \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\|$$

which yields the following bound on the first term of the right hand side of the inequality in (19):

$$\begin{aligned} &\left| \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| - \|g^t(\gamma_{m_i}) - g^t(\gamma_{m_j})\| \right| \\ &\leq \max(|1 - A_l|, |A_u - 1|) \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| \\ &= A \|g^s(\gamma_{m_i}) - g^s(\gamma_{m_j})\| \leq A M_s \|\gamma_{m_i} - \gamma_{m_j}\| \\ &\leq A M_s (\|\gamma_{m_i} - \gamma_i^s\| + \|\gamma_i^s - \gamma_j^s\| + \|\gamma_j^s - \gamma_{m_j}\|) \\ &\leq A M_s (\epsilon_{m_i} + \epsilon_{\mathcal{N}} + \epsilon_{m_j}). \end{aligned} \quad (20)$$

The other terms in (19) can be bounded as

$$\Delta_i^s = \|g^s(\gamma_i^s) - g^s(\gamma_{m_i})\| \leq M_s \|\gamma_i^s - \gamma_{m_i}\| \leq M_s \epsilon_{m_i}$$

since $\gamma_i^s \in B_{\epsilon_{m_i}}(\gamma_{m_i})$. Similarly,

$$\Delta_j^s \leq M_s \epsilon_{m_j}, \quad \Delta_i^t \leq M_t \epsilon_{m_i}, \quad \Delta_j^t \leq M_t \epsilon_{m_j}.$$

Using these bounds in (19) together with the bound in (20), we get

$$\left| \|x_i^s - x_j^s\| - \|x_i^t - x_j^t\| \right| \leq A M_s (\epsilon_{m_i} + \epsilon_{\mathcal{N}} + \epsilon_{m_j}) + (M_s + M_t) (\epsilon_{m_i} + \epsilon_{m_j})$$

which gives the following bound on $|W_{ij}^s - W_{ij}^t|$ from (16)

$$\begin{aligned} |W_{ij}^s - W_{ij}^t| &\leq L_\phi A M_s (\epsilon_{m_i} + \epsilon_{\mathcal{N}} + \epsilon_{m_j}) + L_\phi (M_s + M_t) (\epsilon_{m_i} + \epsilon_{m_j}) \\ &\leq L_\phi A M_s \epsilon_\Gamma + L_\phi (M_s + M_t) \epsilon_\Gamma = \Delta_W. \end{aligned} \quad (21)$$

Case 2. When $x_i^s \sim x_j^s$ are neighbors on the source graph but $x_i^t \not\sim x_j^t$ are not neighbors on the target graph, $W_{ij}^t = 0$, and hence we have

$$|W_{ij}^s - W_{ij}^t| = |W_{ij}^s| = \phi(\|x_i^s - x_j^s\|) \leq \phi(0) = \phi_0.$$

Case 3. Similarly to Case 2, when $x_i^t \sim x_j^t$ are neighbors on the target graph but $x_i^s \not\sim x_j^s$ are not neighbors on the source graph, it is easy to obtain

$$|W_{ij}^s - W_{ij}^t| \leq \phi_0.$$

Having examined all three cases, we can now derive a bound on the difference $|\lambda_i^s - \lambda_j^s|$ between the corresponding source and target Laplacian eigenvalues. Defining

$$P = L^t - L^s$$

we can write $L^t = L^s + P$, where the difference P can be seen as a ‘‘perturbation’’ on the source Laplacian matrix L^s . The spectral radius (the largest eigenvalue magnitude) of P can be bounded as

$$\rho(P) \leq \max_i \sum_j |P_{ij}|.$$

The diagonal entries of the perturbation matrix are given by

$$P_{ii} = L_{ii}^t - L_{ii}^s = D_{ii}^t - D_{ii}^s = \sum_{j \neq i} (W_{ij}^t - W_{ij}^s)$$

whereas the off-diagonal entries are given by

$$P_{ij} = L_{ij}^t - L_{ij}^s = W_{ij}^s - W_{ij}^t$$

for $j \neq i$. Then, for $i = 1, \dots, N$, we have

$$\begin{aligned} \sum_j |P_{ij}| &= |P_{ii}| + \sum_{j \neq i} |P_{ij}| \\ &= \left| \sum_{j \neq i} (W_{ij}^t - W_{ij}^s) \right| + \sum_{j \neq i} |W_{ij}^s - W_{ij}^t| \leq 2 \sum_{j \neq i} |W_{ij}^s - W_{ij}^t| \\ &= 2 \left(\sum_{x_i^s \sim x_j^s, x_i^t \not\sim x_j^t} |W_{ij}^s - W_{ij}^t| + \sum_{x_i^s \not\sim x_j^s, x_i^t \sim x_j^t} |W_{ij}^s - W_{ij}^t| + \sum_{x_i^s \not\sim x_j^s, x_i^t \not\sim x_j^t} |W_{ij}^s - W_{ij}^t| \right). \end{aligned} \quad (22)$$

Using the bounds on the term $|W_{ij}^s - W_{ij}^t|$ for each one of the three studied cases in the above expression, we get

$$\sum_j |P_{ij}| \leq 2 (\beta_i^s K_i^s \Delta_W + (1 - \beta_i^s) K_i^s \phi_0 + (1 - \beta_i^t) K_i^t \phi_0) \leq \rho_{\max} \quad (23)$$

where the last inequality follows from the definition of ρ_{\max} in Lemma 2. We can then bound the spectral radius of the perturbation matrix as

$$\rho(P) \leq \max_i \sum_j |P_{ij}| \leq \rho_{\max}.$$

Finally, from Weyl's inequality, the difference between the corresponding eigenvalues λ_i^s and λ_i^t of the matrices L^s and L^t are upper bounded by the spectral radius of the perturbation matrix. Hence, we have

$$|\lambda_i^s - \lambda_i^t| \leq \rho(P) \leq \rho_{\max}$$

which proves the stated result. □

B.3. Proof of Lemma 3

Proof Let us first define the following parameter

$$a_j^q := \left(\sum_{x_i^t \sim x_j^t, i \in I_{U_q}^t} (f_i^t - f_j^t)^2 \right)^{1/2}$$

which gives the total difference of the target label function between a node x_j^t and its neighbors from the q -th layer. Similarly, let

$$\hat{a}_j^q := \left(\sum_{x_i^t \sim x_j^t, i \in I_{U_q}^t} (\hat{f}_i^t - \hat{f}_j^t)^2 \right)^{1/2}$$

for the estimate of the target label function. Let us also define vectors $A^q \in \mathbb{R}^{|I_{U_q}^t|}$ and $\hat{A}^q \in \mathbb{R}^{|I_{U_q}^t|}$, respectively consisting of the values a_j^{q-1} and \hat{a}_j^{q-1} in their entries, where j varies in the index set $I_{U_q}^t$. We can then lower bound the parameter B_q as

$$\begin{aligned} B_q &= \sum_{j \in I_{U_q}^t} \sum_{x_i^t \sim x_j^t, i \in I_{U_{q-1}}^t} W_{ij}^t (f_i^t - f_j^t)^2 \\ &\geq \sum_{j \in I_{U_q}^t} \sum_{x_i^t \sim x_j^t, i \in I_{U_{q-1}}^t} w_q^{\min} (f_i^t - f_j^t)^2 \\ &\geq \sum_{j \in I_{U_q}^t} w_q^{\min} (a_j^{q-1})^2 = w_q^{\min} \|A^q\|^2 \end{aligned} \quad (24)$$

which gives

$$\|A^q\| \leq \left(\frac{B_q}{w_q^{\min}} \right)^{1/2}.$$

With similar derivations, we also get

$$\|\hat{A}^q\| \leq \left(\frac{\hat{B}_q}{w_q^{\min}} \right)^{1/2}.$$

Now, let $f_{\mathcal{N}_j^q}^t$ and $\hat{f}_{\mathcal{N}_j^q}^t$ respectively denote the restrictions of the vectors f^t and \hat{f}^t to the indices in \mathcal{N}_j^q . Let us fix a node index $j \in I_{U_q}^t$ in the q -th layer. Defining $\vec{1}$ to be a vector of appropriate size consisting of 1's in all its entries, we obtain the following relation by applying the triangle inequality

$$\begin{aligned} \|f_j^t \vec{1} - \hat{f}_j^t \vec{1}\| &\leq \|f_j^t \vec{1} - f_{\mathcal{N}_j^{q-1}}^t\| + \|f_{\mathcal{N}_j^{q-1}}^t - \hat{f}_{\mathcal{N}_j^{q-1}}^t\| + \|\hat{f}_{\mathcal{N}_j^{q-1}}^t - \hat{f}_j^t \vec{1}\| \\ &= a_j^{q-1} + \hat{a}_j^{q-1} + e_j^{q-1} \end{aligned} \quad (25)$$

where the equality follows from the definitions of the parameters a_j^q, \hat{a}_j^q ; and the definition

$$e_j^{q-1} := \|f_{\mathcal{N}_j^{q-1}}^t - \hat{f}_{\mathcal{N}_j^{q-1}}^t\|$$

of the total estimation error at the neighbors of node x_j^t at layer $q-1$. Observing that the constant vector at the left hand side of the inequality in (25) consists of $|\mathcal{N}_j^{q-1}|$ entries, we get

$$|f_j^t - \hat{f}_j^t| \leq \frac{a_j^{q-1} + \hat{a}_j^{q-1} + e_j^{q-1}}{\sqrt{|\mathcal{N}_j^{q-1}|}}.$$

We can then bound the estimation error of the q -th layer as

$$\begin{aligned} \|\hat{f}_{U_q}^t - f_{U_q}^t\|^2 &= \sum_{j \in I_{U_q}^t} (f_j^t - \hat{f}_j^t)^2 \leq \sum_{j \in I_{U_q}^t} \frac{(a_j^{q-1} + \hat{a}_j^{q-1} + e_j^{q-1})^2}{|\mathcal{N}_j^{q-1}|} \\ &\leq \left(\sum_{j \in I_{U_q}^t} (a_j^{q-1} + \hat{a}_j^{q-1} + e_j^{q-1})^2 \right) \left(\sum_{j \in I_{U_q}^t} \frac{1}{|\mathcal{N}_j^{q-1}|} \right) \end{aligned} \quad (26)$$

We proceed by upper bounding each one of the terms at the right hand side of the above inequality. In order to bound the first term, let us first define the vector $E^q \in \mathbb{R}^{|I_{U_q}^t|}$, which is made up of the entries e_j^{q-1} , where j varies in the set $I_{U_q}^t$. We then have

$$\begin{aligned} \sum_{j \in I_{U_q}^t} (a_j^{q-1} + \hat{a}_j^{q-1} + e_j^{q-1})^2 &= \|A^q + \hat{A}^q + E^q\|^2 \\ &\leq \left(\sqrt{\frac{B_q}{w_q^{\min}}} + \sqrt{\frac{\hat{B}_q}{w_q^{\min}}} + \|E^q\| \right)^2. \end{aligned} \quad (27)$$

We can relate the term $\|E^q\|$ to the estimation error of the preceding layer as

$$\begin{aligned}
\|E^q\|^2 &= \sum_{j \in I_{U_q}^t} (e_j^{q-1})^2 = \sum_{j \in I_{U_q}^t} \|f_{\mathcal{N}_j^{q-1}}^t - \hat{f}_{\mathcal{N}_j^{q-1}}^t\|^2 \\
&= \sum_{j \in I_{U_q}^t} \sum_{x_i^t \sim x_j^t, i \in I_{U_{q-1}}^t} (f_i^t - \hat{f}_i^t)^2 = \sum_{i \in I_{U_{q-1}}^t} \sum_{x_j^t \sim x_i^t, j \in I_{U_q}^t} (f_i^t - \hat{f}_i^t)^2 \\
&\leq \sum_{i \in I_{U_{q-1}}^t} K_{q-1}^{\max} (f_i^t - \hat{f}_i^t)^2 = K_{q-1}^{\max} \|f_{U_{q-1}}^t - \hat{f}_{U_{q-1}}^t\|^2.
\end{aligned} \tag{28}$$

This gives in (27)

$$\sum_{j \in I_{U_q}^t} (a_j^{q-1} + \hat{a}_j^{q-1} + e_j^{q-1})^2 \leq \left(\sqrt{\frac{B_q}{w_q^{\min}}} + \sqrt{\frac{\hat{B}_q}{w_q^{\min}}} + \sqrt{K_{q-1}^{\max}} \|f_{U_{q-1}}^t - \hat{f}_{U_{q-1}}^t\| \right)^2.$$

Next, we bound the second term in (26) as

$$\sum_{j \in I_{U_q}^t} \frac{1}{|\mathcal{N}_j^{q-1}|} \leq \sum_{j \in I_{U_q}^t} \frac{1}{K_q^{\min}} = \frac{|I_{U_q}^t|}{K_q^{\min}}$$

where the inequality follows from the definition of K_q^{\min} . Combining this with the bound on the first term in (26), we get

$$\|\hat{f}_{U_q}^t - f_{U_q}^t\|^2 \leq \frac{|I_{U_q}^t|}{K_q^{\min}} \left(\sqrt{\frac{B_q}{w_q^{\min}}} + \sqrt{\frac{\hat{B}_q}{w_q^{\min}}} + \sqrt{K_{q-1}^{\max}} \|\hat{f}_{U_{q-1}}^t - f_{U_{q-1}}^t\| \right)^2$$

which proves the lemma. \square

B.4. Proof of Lemma 4

Proof The proof of Lemma 4 is based on using the recursive relation provided in Lemma 3 between the errors of consecutive layers. For brevity of notation, let us define

$$\begin{aligned}
c_q &= \frac{|I_{U_q}^t|}{K_q^{\min}} \\
b_q &= \sqrt{\frac{B_q}{w_q^{\min}}} + \sqrt{\frac{\hat{B}_q}{w_q^{\min}}}.
\end{aligned} \tag{29}$$

Then, Lemma 3 states that for $q = 1, \dots, Q$,

$$\|\hat{f}_{U_q}^t - f_{U_q}^t\| \leq \sqrt{c_q} \left(b_q + \sqrt{K_{q-1}^{\max}} \|\hat{f}_{U_{q-1}}^t - f_{U_{q-1}}^t\| \right).$$

Observing that the error of layer $q = 0$, which consists of the labeled nodes, is 0 due to the assumption $\hat{f}_i^t = f_i^t$ for $i \in I_L^t$, we have

$$\|\hat{f}_{U_0}^t - f_{U_0}^t\| = 0.$$

This gives

$$\begin{aligned}
\|\hat{f}_{U_1}^t - f_{U_1}^t\| &\leq \sqrt{c_1} b_1 \\
\|\hat{f}_{U_2}^t - f_{U_2}^t\| &\leq \sqrt{c_2} b_2 + \sqrt{c_2} \sqrt{K_1^{\max}} \sqrt{c_1} b_1 \\
\|\hat{f}_{U_3}^t - f_{U_3}^t\| &\leq \sqrt{c_3} (b_3 + \sqrt{K_2^{\max}} (\sqrt{c_2} b_2 + \sqrt{c_2} \sqrt{K_1^{\max}} \sqrt{c_1} b_1)) \\
&= \sqrt{c_3} b_3 + \sqrt{c_3} \sqrt{K_2^{\max}} \sqrt{c_2} b_2 + \sqrt{c_3} \sqrt{K_2^{\max}} \sqrt{c_2} \sqrt{K_1^{\max}} \sqrt{c_1} b_1
\end{aligned} \tag{30}$$

Generalizing this, we get

$$\|\hat{f}_{U_q}^t - f_{U_q}^t\| \leq \sum_{l=1}^q \mu_l b_l$$

where $\mu_q = \sqrt{c_q}$ and

$$\mu_l = \sqrt{c_q} \prod_{m=l}^{q-1} \left(\sqrt{K_m^{\max}} \sqrt{c_m} \right)$$

for $l = 1, \dots, q-1$.

Let us define the vectors $\vec{\mu}_q = [\mu_1 \mu_2 \dots \mu_q]^T$ and $\vec{b}_q = [b_1 b_2 \dots b_q]^T$. Then we can bound the error of the q -th layer via the Cauchy-Schwartz inequality as

$$\|\hat{f}_{U_q}^t - f_{U_q}^t\|^2 \leq |\langle \vec{\mu}_q, \vec{b}_q \rangle|^2 \leq \|\vec{\mu}_q\|^2 \|\vec{b}_q\|^2. \tag{31}$$

Noticing that

$$b_q = \sqrt{\frac{B_q}{w_q^{\min}}} + \sqrt{\frac{\hat{B}_q}{w_q^{\min}}} \leq \frac{1}{\sqrt{w^{\min}}} (\sqrt{B_q} + \sqrt{\hat{B}_q})$$

and defining the vectors

$$\begin{aligned}
C_q &= [\sqrt{B_1} \sqrt{B_2} \dots \sqrt{B_q}]^T \\
\hat{C}_q &= [\sqrt{\hat{B}_1} \sqrt{\hat{B}_2} \dots \sqrt{\hat{B}_q}]^T
\end{aligned} \tag{32}$$

we can bound the norm of \vec{b}_q as

$$\begin{aligned}
\|\vec{b}_q\| &\leq \frac{1}{\sqrt{w^{\min}}} \|C_q + \hat{C}_q\| \leq \frac{1}{\sqrt{w^{\min}}} (\|C_q\| + \|\hat{C}_q\|) \\
&\leq \frac{1}{\sqrt{w^{\min}}} \left(\sqrt{B_1 + B_2 + \dots + B_q} + \sqrt{\hat{B}_1 + \hat{B}_2 + \dots + \hat{B}_q} \right) \\
&\leq \frac{1}{\sqrt{w^{\min}}} \left(\sqrt{(f^t)^T L^t f^t} + \sqrt{(\hat{f}^t)^T L^t \hat{f}^t} \right) \\
&\leq \frac{1}{\sqrt{w^{\min}}} \left(\sqrt{B} + \sqrt{\hat{B}} \right).
\end{aligned} \tag{33}$$

Using this in (31), the total target estimation error can be bounded as

$$\begin{aligned}
\|\hat{f}^t - f^t\|^2 &= \sum_{q=1}^Q \|\hat{f}_{U_q}^t - f_{U_q}^t\|^2 \leq \sum_{q=1}^Q \|\vec{\mu}_q\|^2 \|\vec{b}_q\|^2 \\
&\leq \frac{1}{w^{\min}} \left(\sqrt{B} + \sqrt{\hat{B}} \right)^2 \sum_{q=1}^Q \|\vec{\mu}_q\|^2.
\end{aligned} \tag{34}$$

Replacing $\|\vec{\mu}_q\|^2$ with its open expression as

$$\begin{aligned}
\|\vec{\mu}_q\|^2 &= \sum_{l=1}^q \mu_l^2 = \mu_q^2 + \sum_{l=1}^{q-1} \mu_l^2 \\
&= c_q + \sum_{l=1}^{q-1} c_q \prod_{m=l}^{q-1} K_m^{\max} c_m = c_q \left(1 + \sum_{l=1}^{q-1} \prod_{m=l}^{q-1} K_m^{\max} c_m \right) \\
&= \frac{|I_{U_q}^t|}{K_q^{\min}} \left(1 + \sum_{l=1}^{q-1} \prod_{m=l}^{q-1} K_m^{\max} \frac{|I_{U_m}^t|}{K_m^{\min}} \right)
\end{aligned} \tag{35}$$

we get the stated result.

□