Distributional Network of Networks for Modeling Data Heterogeneity

Jun Wu
University of Illinois at Urbana-Champaign
junwu3@illinois.edu

Jingrui He
University of Illinois at Urbana-Champaign
jingrui@illinois.edu

Hanghang Tong
University of Illinois at Urbana-Champaign
htong@illinois.edu

ABSTRACT

Heterogeneous data widely exists in various high-impact applications. Domain adaptation and out-of-distribution generalization paradigms have been formulated to handle the data heterogeneity across domains. However, most existing domain adaptation and out-of-distribution generalization algorithms do not explicitly explain how the label information can be adaptively propagated from the source domains to the target domain. Furthermore, little effort has been devoted to theoretically understanding the convergence of existing algorithms based on neural networks.

To address these problems, in this paper, we propose a generic distributional network of networks (TENON) framework, where each node of the main network represents an individual domain associated with a domain-specific network. In this case, the edges within the main network indicate the domain similarity, and the edges within each network indicate the sample similarity. The crucial idea of TENON is to characterize the within-domain label smoothness and cross-domain parameter smoothness in a unified framework. The convergence and optimality of TENON are theoretically analyzed. Furthermore, we show that based on the TENON framework, domain adaptation and out-of-distribution generalization can be naturally formulated as transductive and inductive distribution learning problems, respectively. This motivates us to develop two instantiated algorithms (TENON-DA and TENON-OOD) of the proposed TENON framework for domain adaptation and out-of-distribution generalization. The effectiveness and efficiency of TENON-DA and TENON-OOD are verified both theoretically and empirically.

CCS CONCEPTS

• **Computing methodologies** → *Transfer learning*.

KEYWORDS

network of networks, data heterogeneity, domain adaptation, out-of-distribution generalization

ACM Reference Format:
Jun Wu, Jingrui He, and Hanghang Tong. 2024. Distributional Network of Networks for Modeling Data Heterogeneity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671994

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08…$15.00
https://doi.org/10.1145/3637528.3671994

1 INTRODUCTION

Modern machine learning algorithms have demonstrated remarkable success across a wide range of high-impact applications, such as sentiment analysis [57], news tagging classification [31], etc. One common assumption behind these algorithms is that the training and test samples are independently and identically distributed (IID). However, this IID assumption is often violated in real scenarios where the samples are collected from heterogeneous domains under distribution shift [48], e.g., Amazon review collected from different products [8], news headlines collected from different time stamps [53]. Two learning paradigms have been developed to address the challenge of data heterogeneity across domains: domain adaptation [4, 57] and out-of-distribution generalization [6, 34]. As shown in Figure 1(a), domain adaptation[1] aims at learning a prediction function on a target domain with only unlabeled training samples, by exploiting knowledge from source domains. In contrast, out-of-distribution generalization optimizes a domain-agnostic model from source domains such that this model can be directly applied to any relevant unseen target domains. Different from domain adaptation, target domains are unseen during training for out-of-distribution generalization.

Most existing domain adaptation and out-of-distribution generalization algorithms [2, 28, 47, 57] build a single model to learn the domain-invariant representation from different domains. The invariant representation learned by a domain-agnostic model can be explained as the common knowledge shared by all domains. Nevertheless, it is a strong assumption that all domains share the same model parameters. This is because this assumption underestimates the domain-specific characteristics encoding class separability. Though recent works [7, 40, 51] propose to learn both within-domain specificity and cross-domain commonality, disentangling domain-invariant and domain-specific representations is a nontrivial task. This is because it is challenging to accurately differentiate the domain-invariant representation from the domain-specific representation within samples. The aforementioned frameworks might suffer from the following limitations. First, the connection between the domain relationship and the model (parameters) similarity is under-explored, e.g., similar domains may share similar model parameters [24, 49]. Second, it is not explained how the label information can be adaptively propagated from the source domain to the target domain. Third, little effort has been devoted to understanding

[1] It is also termed as "multi-source domain adaptation" to indicate the existence of multiple source domains in previous works [39, 57]. In this paper, we use the generic term "domain adaptation" by assuming that at least one source domain is available.

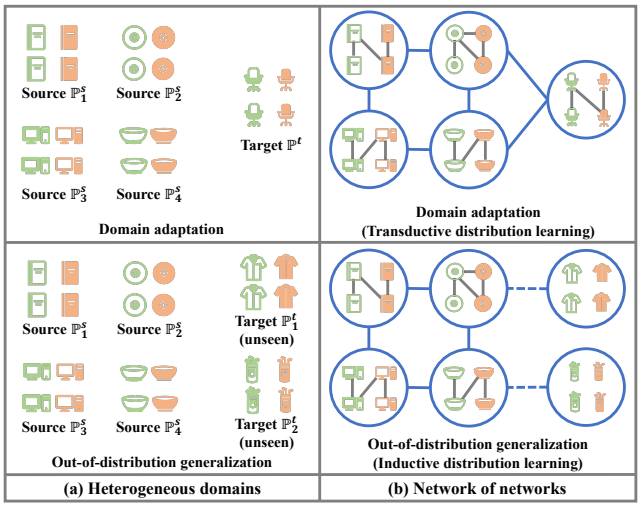


Figure 1: Illustration of the network of networks on handing heterogeneous domains (semantic classification on Amazon products [8] is used where green indicates negative review and orange indicates positive review). (a) Domain adaptation and out-of-distribution (OOD) generalization involve different domains. Target domains are unseen during training for out-of-distribution generalization. (b) In the network of networks, each node of the main network represents one domain, and it is formed by a network over domain-specific samples. For OOD generalization, the dotted lines indicate that the edges between source and target domains are accessible only during the testing phase.

the model convergence of previous algorithms [2, 45, 47, 57] based on deep neural networks.

To this end, in this paper, we propose a generic distributional network of networks (TENON) framework, which allows each domain to learn domain-specific model parameters. It is motivated by recent observation [6, 50] that both domain adaptation and out-of-distribution generalization can be explained as follows. Given a meta-distribution $\mathscr{P}$, the data distributions $\mathbb{P}_1, \cdots, \mathbb{P}_K$ of different domains can be considered as IID realizations of $\mathscr{P}$, and the samples $\{x_i^k, y_i^k\}_{i=1}^{n_k}$ within domain k are IID realizations from $\mathbb{P}_k$. Having this in mind, TENON reformulates the heterogeneous domains as a network of networks [36], which encodes both high-level domain relationships and low-level sample relationships. As shown in Figure 1(b), the main network characterizes the relationship among different domains, where each node (blue circles) is a domain, and the edges (solid or dotted blue lines) imply domain similarity. Each domain is further represented by a domain-specific network (e.g., a network within each blue circle), where each node (colored products) is a sample, and the edges (black lines) imply sample similarity. The intuition behind TENON is that (i) domains share similar model parameters [49, 56] if they have similar data distributions, and (ii) samples tend to have similar class labels if they are similar in the input space [58, 60]. To this end, the proposed TENON framework is composed of both within-domain label smoothness and cross-domain parameter smoothness regularizations. We theoretically show that the convergence and optimality of TENON can be guaranteed when using overparameterized neural networks [20, 25] to instantiate the learning functions of TENON.

More specifically, Figure 1(b) shows that domain adaptation [57] and out-of-distribution generalization [6, 34] can be naturally considered as transductive and inductive distribution learning problems, respectively. That is, domain adaptation can build a network of networks using source and target domains as a pre-processing step. Here the sample similarity within each domain-specific network and domain similarity within the main network can be empirically estimated using input samples. Then using the constructed network of networks, we propose an instantiated algorithm (TENON-DA) of TENON to propagate the knowledge from labeled source domains to the unlabeled target domain for domain adaptation. In contrast, out-of-distribution generalization can only access source (training) domains, and thus we build a network of networks over source domains during training. During the testing phase, target (testing) domains will be added to the main network as the new nodes. As a result, out-of-distribution generalization is formulated as an inductive learning [17] problem w.r.t. the network of networks. To solve this problem, we propose another instantiated algorithm (TENON-OOD) of TENON to generalize the relevant knowledge from source (training) domains to target (testing) domains. The effectiveness and efficiency of TENON-DA and TENON-OOD are demonstrated in a variety of data mining tasks. The major contributions of this paper are summarized as follows.

- **Framework:** We propose a generic distributional network of networks (TENON) framework for modeling data heterogeneity across domains. Notably, TENON provides a unified viewpoint of domain adaptation and out-of-distribution generalization. Furthermore, the convergence and optimality of TENON are theoretically analyzed.
- **Algorithms:** We provide two instantiated algorithms (i.e., TENON-DA and TENON-OOD) of TENON for domain adaptation and out-of-distribution generalization. It is revealed that both algorithms inherit the convergence properties of the TENON framework. Besides, in the context of domain adaptation, we show that TENON-DA minimizes the error upper bound of the target domain.
- **Experiments:** Extensive experiments on various data sets demonstrate the effectiveness and efficiency of the proposed algorithms for both domain adaptation and out-of-distribution generalization.

The rest of the paper is organized as follows. Section 2 summarizes the related work and Section 3 provides the problem settings. In Section 4, we propose a novel distributional network of networks (TENON) framework, followed by the instantiated algorithms for domain adaptation and out-of-distribution generalization in Section 5. Section 6 shows the experimental results, and finally, we conclude the paper in Section 7.

2 RELATED WORK

2.1 Domain Adaptation

Domain adaptation [4, 35] studies the transfer of knowledge or information from source domains to a relevant target domain. It is theoretically shown [1, 44, 47, 57] that the generalization error of

a learning algorithm within the target domain can be bounded by the source errors and domain discrepancy. This thus leads to the domain adaptation algorithms [9, 14, 30, 33, 39, 50, 55] by empirically minimizing the prediction errors within source domains and distribution discrepancy across domains. The most similar works to ours include [5, 52], where Xu et al. [52] build a domain graph to encode topological structures among different domains and Berthelot et al. [5] unify the semi-supervised learning and domain adaptation. However, our TENON framework is fundamentally different from previous works in the following aspects. First, previous works leverage a single model to learn domain-invariant representation, whereas TENON enables the domain-specific models to characterize the domain relationship. Second, the global convergence and optimality of TENON are analyzed theoretically. In contrast, little theoretical analysis regarding the convergence of domain adaptation algorithms is provided in previous works. Third, our TENON framework can be applied to both domain adaptation and out-of-distribution generalization, while previous works consider only the domain adaptation settings.

2.2 Out-of-Distribution Generalization

Out-of-distribution (OOD) generalization aims at learning a domain-agnostic model from an arbitrary number of training source domains [6, 21, 34]. In recent years, various OOD generalization algorithms have been proposed from the following aspects: domain-invariant representation learning [2, 28], meta regularization [3, 27], domain augmentation [46, 59], gradient operation [42, 45], etc. These algorithms directly apply the learned model to the new testing domains. Compared to previous works, the proposed TENON framework focuses on explicitly propagating model parameters from training to testing domains based on the distribution similarity among domains. This is in sharp contrast to previous works which learn a commonly shared model among all domains.

3 PROBLEM DEFINITIONS

We let $\mathcal{X}$ and $\mathcal{Y}$ be the input space and output space, respectively. Suppose there are K different domains drawn from a meta-distribution $\mathscr{P}$, i.e., $\mathbb{P}_1, \cdots, \mathbb{P}_K \sim \mathscr{P}$ where $\mathbb{P}_k$ denotes the data distribution[2] of the k^{th} domain over $\mathcal{X} \times \mathcal{Y}$. Each domain is associated with a model $f(\cdot;\theta_k) : \mathcal{X} \to \mathcal{Y}$ parameterized by θ_k. There are n_k labeled or unlabeled samples in domain k, where $x_i^k \in \mathcal{X}$ is the input sample and y_i^k is the output label if available. In addition, we let $\mathbb{I}$ denote the identity matrix, $||\cdot||_p$ and $||\cdot||_F$ denote L_p norm and Frobenius norm, respectively.

Following [4], we focus on the problem of learning from different domains, where data heterogeneity exists among domains. Specifically, in this paper, we focus on two research problems: domain adaptation [4, 57] and out-of-distribution generalization [6, 21]. Both research problems involve modeling the data heterogeneity across domains. Their goal is to learn a prediction function on the target domain without label information, by leveraging latent knowledge from relevant source domains.

[2]In this paper, we will use $\mathbb{P}_k$ to denote both the data distribution of domain k and the domain k itself.

Problem Definition 1 (Domain Adaptation). *Given a set of source domains $\{\mathbb{P}_k\}_{k=1}^{K-1}$ each with labeled samples $\{x_i^k, y_i^k\}_{i=1}^{n_k}$, and a target domain $\mathbb{P}_K$ with only unlabeled samples $\{x_i^K\}_{i=1}^{n_K}$, domain adaptation aims to learn a prediction function on the target domain using knowledge from source domains.*

Problem Definition 2 (Out-of-Distribution Generalization). *Given a set of source domains $\{\mathbb{P}_k\}_{k=1}^{K}$ each with samples $\{x_i^k, y_i^k\}_{i=1}^{n_k}$, out-of-distribution generalization aims to learn a prediction function from source domains such that this prediction function can be directly applied to unseen target domains.*

As illustrated in Figure 1, a group of distributions (or domains) $\{\mathbb{P}_k\}_{k=1}^{K}$ over a meta-distribution $\mathscr{P}$ can be formulated as a network of networks [36], where each node of the main network represents a domain and each network is formed by domain-specific samples. This motivates us to rethink the modeling of data heterogeneity by capturing both sample similarity within domains and distribution similarity across domains. First, in each domain, two samples tend to have similar output values if they are similar in the input space [19, 58, 60]. Second, given a learning algorithm $f(\cdot)$, two domains would be close in the parameter space if they are distributionally similar [49, 56].

4 PROPOSED FRAMEWORK

In this section, we propose a simple and generic distributional network of networks (TENON) framework for modeling heterogeneous data from multiple domains.

4.1 Distributional Network of Networks

It is shown [50] that knowledge transferability can be positively correlated with the distribution similarity across domains. This motivates us to model the heterogeneous domains by capturing the domain relationship in the parameter space (shown in Figure 2). To this end, we propose a simple yet generic distributional network of networks (TENON) framework with the following objective function.

$$\begin{aligned}\min_{\{\theta_k\}_{k=1}^K} \lambda \sum_{k=1}^{K} & \underbrace{\sum_{i=1}^{n_k} \left\| f(x_i^k;\theta_k) - y_i^k \right\|_2^2}_{\text{Label consistency within domain}} \\ & + \frac{1}{2} \underbrace{\sum_{k=1}^{K} \sum_{i,j=1}^{n_k} s_{ij}^k \left\| \frac{f(x_i^k;\theta_k)}{\sqrt{D_{ii}^k}} - \frac{f(x_j^k;\theta_k)}{\sqrt{D_{jj}^k}} \right\|_2^2}_{\text{Label smoothness within domain}} \\ & + \frac{1}{2} \underbrace{\sum_{k,k'=1}^{K} d_{kk'} \left\| \frac{\theta_k}{\sqrt{M_{kk}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}} \right\|_F^2}_{\text{Parameter smoothness across domains}}\end{aligned} \tag{1}$$

where s_{ij}^k indicates the sample similarity between x_i^k and x_j^k within the k-th domain, and $d_{kk'}$ denotes the domain similarity between the k-th domain and the k'-th domain. Here $D_{ii}^k = \sum_{j=1}^{n_k} s_{ij}^k$ and $M_{kk} = \sum_{k'=1}^{K} d_{kk'}$. θ_k denotes the model parameters within the k-th domain. $\lambda > 0$ is a hyper-parameter to balance different terms.

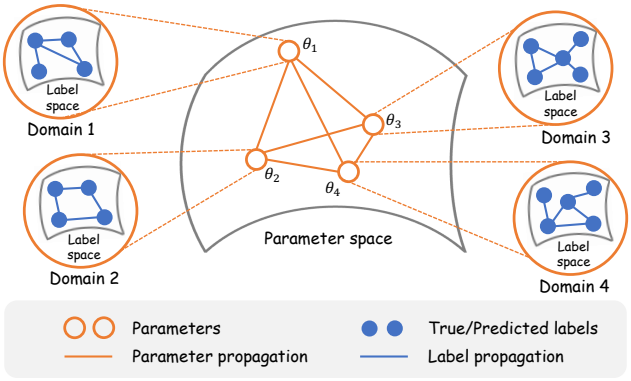


Figure 2: Illustration of TENON in information propagation. Label information is propagated in the labeling space within each domain, while parameter information is propagated in the parameter space across domains.

Following [19, 58], the sample similarity s_{ij}^k can be empirically estimated as follows.

$$s_{ij}^k = \exp\left(-\sigma \cdot \left\|x_i^k - x_j^k\right\|_1\right)$$

where $\sigma \in \mathbb{R}$ is a hyper-parameter. In addition, a variety of domain discrepancy measures have been proposed to model the heterogeneous domains, e.g., $\mathcal{H}\Delta\mathcal{H}$-divergence [4], Maximum Mean Discrepancy [15, 30], Wasserstein distance [44], f-divergence [1], etc. It is flexible in defining $d_{kk'}$ in Eq. (1) based on existing domain discrepancy measures. In this paper, under the covariate shift assumption [38] (i.e., $\mathbb{P}(y|x)$ is shared for all domains), we use Maximum Mean Discrepancy (MMD) [15] to define the domain similarity $d_{kk'}$ as follows.

$$\mathrm{MMD}(k,k') = \left\| \frac{1}{n_k}\sum_{i=1}^{n_k}\phi(x_i^k) - \frac{1}{n_{k'}}\sum_{j=1}^{n_{k'}}\phi(x_j^{k'}) \right\|_{\mathcal{H}_\mathcal{K}}^2$$

$$d_{kk'} = \exp\left(-\sigma \cdot \mathrm{MMD}(k,k')\right)$$

where $\phi(\cdot): \mathcal{X} \to \mathcal{H}_\mathcal{K}$ is a kernel mapping from an input space $\mathcal{X}$ to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\mathcal{K}$.

The intuition behind Eq. (1) is explained as follows. The first term captures the consistency of models $\{f(\cdot;\theta_k)\}_{k=1}^K$ with the prior label information. The second term measures the label smoothness within each domain. It implies that input samples have similar prediction values if they are similar in the input space. Furthermore, the third term measures the cross-domain model smoothness in the parameter space. Notably, graph-based parameter smoothness regularization [29, 56] has been studied in multi-task learning. However, compared to previous works, our framework of Eq. (1) explicitly reveals the connection between the domain distribution discrepancy and the model (parameters) similarity, i.e., domains have similar model parameters if they are distributionally similar. Furthermore, by incorporating the within-domain label smoothness regularization (i.e., the second term of Eq. (1)), TENON allows propagating label information from labeled source samples to unlabeled target samples, whereas previous works [29, 56] collaboratively update the model parameters over the labeled samples from all domains.

As shown in Figure 2, the label information encoded by a domain-specific model is propagated within each domain, while the model information is propagated across domains in the parameter space. We show in Subsection 4.3 that in the special case where $d_{kk'} = 0$ for all domains k, k', the objective of TENON in Eq. (1) exactly recovers the label propagation [58, 60] in every domain. On top of label propagation, the parameter propagation of TENON enables handling data heterogeneity when samples are collected from multiple domains [33, 49, 57].

4.2 Convergence Analysis

The convergence and optimality of TENON can be analyzed by considering different instantiations of learning models $\{f(\cdot;\theta_k)\}_{k=1}^K$. In the following, we start with the simple linear regression functions, i.e., $f(x;\theta_k) = \theta_k^T x$ for all $k \in \{1, 2, \cdots, K\}$. The following lemma shows the global convergence and optimality of the TENON framework.

Lemma 3. *Given linear models $f(x;\theta_k) = \theta_k^T x$ for $k \in \{1, \cdots, K\}$, the objective of Eq. (1) can be minimized at*

$$\Theta^* = \lambda\hat{\mathbf{X}}\left(\left(\hat{\mathbf{A}} + \lambda\mathbb{I}\right)\hat{\mathbf{X}}^T\hat{\mathbf{X}} + \left(\hat{\mathbf{X}}^T\hat{\mathbf{X}}\right)^{-1}\hat{\mathbf{X}}^T\hat{\mathbf{B}}\hat{\mathbf{X}}\right)^{-1}\mathbf{y}$$

where $\Theta = [\theta_1^T, \cdots, \theta_K^T]^T, \mathbf{X}_k = [x_1^k, x_2^k, \cdots, x_{n_k}^k], \mathbf{y} = [y_1^1, \cdots, y_{n_1}^1, \cdots, y_1^K, \cdots, y_{n_K}^K]^T$ *and*

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{X}_K \end{bmatrix}, \quad \hat{\mathbf{A}} = \begin{bmatrix} \bar{\mathbf{A}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{A}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{A}}_K \end{bmatrix}$$

$$\hat{\mathbf{B}} = \mathbb{I}_{Kd_{in}\times Kd_{in}} - \mathbf{B} \otimes \mathbb{I}_{d_{in}\times d_{in}}$$

where $\mathbf{A}_k = (\mathbf{D}^k)^{-1/2}\mathcal{S}^k(\mathbf{D}^k)^{-1/2}$ *is the normalized sample similarity matrix of domain k with* $\bar{\mathbf{A}}_k = \mathbb{I} - \mathbf{A}_k$, *and* $\mathbf{B} = \mathbf{M}^{-1/2}\mathcal{D}\mathbf{M}^{-1/2}$ *is the normalized domain similarity matrix.*[3] *$\otimes$ denotes the Kronecker product of two matrices. d_{in} is the dimensionality of input samples.*

Next, we instantiate the learning models $\{f(\cdot;\theta_k)\}_{k=1}^K$ with over-parameterized neural networks [20, 25]. This allows us to reveal the convergence of TENON in Eq. (1) with commonly used neural network architectures[4]. For notation simplicity, we will use $f(\cdot;\Theta)$ to denote the overall learning function with $f(x^k;\Theta) = f(x^k;\theta_k)$ for any sample x^k from domain k. It is observed [25] that neural network $f(\cdot;\Theta)$ can be approximated by its linearized version $f^{\mathrm{lin}}(\cdot;\Theta)$, i.e., $\sup_{t\geq 0}\left\|f_t(x;\Theta) - f_t^{\mathrm{lin}}(x;\Theta)\right\| = O(h^{-\frac{1}{2}})$ where h is the width of neural networks.[5] $f_t(\cdot;\Theta)$ denotes the model at time step t, and $f^{\mathrm{lin}}(\cdot;\Theta)$ is given by the first order Taylor expansion of $f(\cdot;\Theta)$: $f_t^{\mathrm{lin}}(x;\Theta) = f_0(x;\Theta) + \nabla f_0(x;\Theta)\,(\Theta_t - \Theta_0)$ Inspired by this observation, we generalize the results of Lemma 3 by instantiating $\{f(\cdot;\theta_k)\}_{k=1}^K$ with neural networks. The following theorem shows

[3] $\mathcal{S}^k$ is sample similarity matrix of domain k with the entry $[\mathcal{S}^k]_{ij} = s_{ij}^k$, and $\mathbf{D}^k$ is a diagnal matrix with the entry $[\mathbf{D}^k]_{ii} = D_{ii}^k$. $\mathcal{D}$ is domain similarity matrix with the entry $[\mathcal{D}]_{kk'} = d_{kk'}$, and $\mathbf{M}$ is a diagnal matrix with the entry $[\mathbf{M}]_{kk} = M_{kk}$.
[4] Following [25], θ_k denotes the vectorized parameters of the neural network model within domain k here.
[5] In this case, "width" can be the number of neurons in a fully-connected layer or the number of channels in a convolutional layer.

the global convergence of TENON under gradient descent when the layer width of $\{f(\cdot;\theta_k)\}_{k=1}^K$ goes to infinity.

Theorem 4 (Convergence and Optimality of TENON). *Let* $\mathbf{X}$ *denote all training samples. In the limit of layer width, the model parameters* $\mathbf{\Theta}$ *in the objective of Eq. (1) converges to*

$$\lim_{t\to\infty} \mathbf{\Theta}_t = -\nabla_{\mathbf{\Theta}} f_0(\mathbf{X})^T \mathbf{K}_{\text{NTK}}^{-1} \mathbf{\Gamma}^{-1} \left(\mathbf{\Omega} - \lambda \mathbf{y}\right) + \mathbf{\Theta}_0$$

where t *is the training time step,* $\mathbf{\Theta}_0$ *denotes the initialized parameters, and* $f_0(\mathbf{X}) = \text{vec}(f_0(x_i^k;\theta^k)|i \in \{1,2,\cdots,n_k\}, k \in \{1,2,\cdots,K\})$ *is model output with initialized parameters. Moreover, the prediction function* $f(\cdot;\theta_k)$ *of Eq. (1) for any testing sample* x^k *within domain* k *converges to*

$$\begin{aligned}\lim_{t\to\infty} f_t(x^k;\theta_k) = {} & \lambda \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}) \mathbf{K}_{\text{NTK}}^{-1} \mathbf{\Gamma}^{-1} \mathbf{y} \\ & + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}) \mathbf{K}_{\text{NTK}}^{-1} \mathbf{\Gamma}^{-1} \mathbf{\Omega}\end{aligned}$$

where

$$\begin{aligned}\mathbf{\Gamma} &= \hat{\mathbf{A}} + \lambda \mathbb{I} + \mathbf{K}_{\text{NTK}}^{-1} \nabla_{\mathbf{\Theta}} f_0(\mathbf{X}) \hat{\mathbf{B}} \nabla_{\mathbf{\Theta}} f_0(\mathbf{X})^T \mathbf{K}_{\text{NTK}}^{-1} \\ \mathbf{\Omega} &= \mathbf{K}_{\text{NTK}}^{-1} \nabla_{\mathbf{\Theta}} f_0(\mathbf{X}) \hat{\mathbf{B}} \mathbf{\Theta}_0 + (\hat{\mathbf{A}} + \lambda \mathbb{I}) f_0(\mathbf{X}) \\ \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}) &= [0,\cdots,0, \underbrace{\omega_1^k,\cdots,\omega_{n_k}^k}_{\text{Within domain } k}, 0,\cdots,0]\end{aligned}$$

and $\mathbf{K}_{\text{NTK}} = \text{diag}(\mathbf{K}_{11}, \mathbf{K}_{22}, \cdots, \mathbf{K}_{KK})$. $\mathbf{K}_{kk}$ *is a neural tangent kernel [20] matrix within domain* k*, i.e., its entry is given by* $[\mathbf{K}_{kk}]_{ij} = \left\langle \nabla_{\theta_k} f_0(x_i^k;\theta_k), \nabla_{\theta_k} f_0(x_j^k;\theta_k) \right\rangle$. $\omega_i^k = \left\langle \nabla_{\theta_k} f_0(x^k;\theta_k), \nabla_{\theta_k} f_0(x_i^k;\theta_k) \right\rangle$.

4.3 Discussion

In this section, we provide a more intuitive explanation regarding how the proposed TENON framework enables within-domain label propagation and cross-domain parameter propagation, respectively.

Corollary 5 (Individual Label Propagation). *In the special case where* $d_{kk'} = 0$ $(k \neq k')$*, with the same conditions as Theorem 4, for any* $k \in \{1,\cdots,K\}$*, the predicted values of* $f(\cdot;\theta_k)$ *in Eq. (1) over the training samples* $\mathbf{X}_k = [x_1^k,\cdots,x_{n_k}^k]$ *in domain* k *converge to*

$$\lim_{t\to\infty} f_t\left(\mathbf{X}_k;\theta_k\right) = (1-\alpha)\left(\mathbb{I} - \alpha \mathbf{A}_k\right)^{-1} \mathbf{y}_k \tag{2}$$

where $f_t(\mathbf{X}_k;\theta_k) = [f_t(x_1^k;\theta_k), f_t(x_2^k;\theta_k), \cdots, f_t(x_{n_k}^k;\theta_k)]^T$, $\mathbf{y}_k = [y_1^k, y_2^k, \cdots, y_{n_k}^k]^T$*, and* $\alpha = \frac{1}{\lambda+1}$*. Furthermore, for any testing sample* x^k*, it holds*

$$\begin{aligned}\lim_{t\to\infty} f_t(x^k;\theta_k) = {} & (1-\alpha)\mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}_k)\mathbf{K}_{kk}^{-1}\left(\mathbb{I} - \alpha\mathbf{A}_k\right)^{-1}\mathbf{y}_k \\ & + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}_k)\mathbf{K}_{kk}^{-1} f_0(\mathbf{X}_k)\end{aligned} \tag{3}$$

It can be seen from Eq. (2) in Corollary 5 that when all domains are irrelevant (i.e., $d_{kk'} = 0$ for any $k \neq k'$), the objective of TENON is equivalent to standard label propagation [19, 58, 60] on each individual domain and no knowledge is shared across domains. Furthermore, previous label propagation approaches [58, 60] focus on transductive semi-supervised learning, where labels are inferred for a set of unlabeled training samples (shown in Eq. (2)), whereas Corollary 5 provides a feasible solution for inductive semi-supervised learning, where the labels can be inferred for new unseen testing samples (shown in Eq. (3)).

Algorithm 1 TENON-DA

Input: $(K-1)$ source domains $\{\mathbb{P}_k\}_{k=1}^{K-1}$, a target domain $\mathbb{P}_K$;
Output: Predicted output values of target samples.

```
———————— Training Stage (Pre-computing) ————————
Calculate all sample similarity s_ij^k and domain similarity d_kk';
for k = 1, ··· , K do
    Calculate block neural tangent kernel K_kk;
    Calculate inverse matrix K_kk^{-1};
    for k' = k + 1, ··· , K do
        Calculate block neural tangent kernel K_kk';
    end for
end for
Calculate ΓK_NTK;
Calculate y* = λK_NTK^{-1} Γ^{-1} y = λ(ΓK_NTK)^{-1} y;
Obtain target propagated labels y_K^* = [y*]_{-n_K:};
———————— Inference Stage ————————————————
for testing sample x_K^test from target domain K do
    Calculate neural tangent kernel K_KK(x_K^test, X_K);
    Calculate y_K^test = K_KK(x_K^test, X_K) y_K^*;
end for
```

Corollary 6 (Global Parameter Propagation). *In the special case where* $s_{ij}^k = 0$ $(i \neq j, k = 1,\cdots,K)$*, with the same conditions as Theorem 4, for any* $k \in \{1,\cdots,K\}$*, the model parameters* θ_k *of* $f(\cdot;\theta_k)$ *in Eq. (1) is updated under gradient descent as follows.*

$$\begin{aligned}\theta_k(t+1) = {} & \left((1-\eta)\mathbb{I} - \lambda\nabla f(\mathbf{X}_k)^T \nabla f(\mathbf{X}_k)\right)\theta_k(t) \\ & + \eta \sum_{k'=1}^{K} \frac{d_{kk'}}{\sqrt{M_{kk} M_{k'k'}}} \theta_{k'}(t) + \eta\lambda \nabla f(\mathbf{X}_k)^T \mathbf{y}_k\end{aligned}$$

where η *is the learning rate and* $\theta_k(t)$ *denotes the model parameters* θ_k *at time step* t*.*

Corollary 6 reveals that if we do not consider the sample similarity, i.e., $s_{ij}^k = 0$ $(i \neq j, k = 1,\cdots,K)$, the model parameters θ_k of the domain k would recursively aggregate knowledge from all other domains. More specifically, if two domains have similar data distributions, i.e., $d_{kk'}$ is large, it is more likely to propagate parameter knowledge between these two domains. This observation is also consistent with previous works [24, 49].

5 PROPOSED ALGORITHMS

In this section, we provide two instantiated algorithms of TENON for domain adaptation (TENON-DA) and out-of-distribution generalization (TENON-OOD). The crucial idea is to formulate domain adaptation and out-of-distribution generalization as transductive distribution learning and inductive distribution learning w.r.t. network of networks [36], respectively.

5.1 Transductive Distribution Learning

We formulate domain adaptation [47] as a transductive distribution learning problem. As shown in Figure 1(b), each domain (source or target domain) is formulated as a node in the main network, and samples within each domain form a domain-specific network. Thus, domain adaptation aims to propagate the label information (1) from

source domains to the target domain (domain-level propagation) and (2) from labeled samples to unlabeled samples (sample-level propagation). To this end, we instantiate the proposed TENON framework (denoted as TENON-DA) for domain adaptation below.

Given $K-1$ source domains $\{\mathbb{P}_k\}_{k=1}^{K-1}$ each with labeled samples $\{x_i^k, y_i^k\}_{i=1}^{n_k}$, and a target domain $\mathbb{P}_K$ with only unlabeled samples $\{x_i^K\}_{i=1}^{n_K}$, the objective function of TENON-DA is directly given by Eq. (1). Here the class label y_i^k ($k = 1, \cdots, K-1$) of source training sample x_i^k is represented as a one-hot vector, and the class label y_i^K of unlabeled target training sample x_i^K is initialized as a zero vector. Following Theorem 4, we can obtain the closed-form solution of TENON-DA as follows. Suppose $f_0(\cdot;\mathbf{\Theta}) = 0, \mathbf{\Theta}_0 = \mathbf{0}$, the predicted class labels of target training samples are given by

$$\mathbf{y}_K^* = [\mathbf{y}^*]_{-n_K:} \qquad \text{where} \quad \mathbf{y}^* = \lambda \mathbf{K}_{\text{NTK}}^{-1} \mathbf{\Gamma}^{-1} \mathbf{y}$$

where $[\mathbf{y}^*]_{-n_K:}$ denotes the last n_K rows of predicted output values $\mathbf{y}^*$. Moreover, for any new target testing sample x_K^{test}, the predicted class label via TENON-DA is

$$y_K^{\text{test}} = \lambda \mathbf{K}_{\text{NTK}}(x_K^{\text{test}}, \mathbf{X}) \mathbf{K}_{\text{NTK}}^{-1} \mathbf{\Gamma}^{-1} \mathbf{y} = \mathbf{K}_{\text{KK}}(x_K^{\text{test}}, \mathbf{X}_K) \mathbf{y}_K^* \quad (4)$$

where $\mathbf{K}_{\text{KK}}(x_K^{\text{test}}, \mathbf{X}_K) = [\mathbf{K}_{\text{NTK}}(x_K^{\text{test}}, x_K^1), \cdots, \mathbf{K}_{\text{NTK}}(x_K^{\text{test}}, x_{n_K}^K)]$. We see that TENON-DA is a non-parametric domain adaptation approach. As shown in Algorithm 1, we can pre-compute the propagated labels $\mathbf{y}_K^*$ for unlabeled target training samples. Then, the class label of any testing target sample is inferred using the propagated labels $\mathbf{y}_K^*$ and the neural tangent kernel vector $\mathbf{K}_{\text{KK}}(x_K^{\text{test}}, \mathbf{X}_K)$ between this testing sample and the target training samples.

In the following, we theoretically analyze the generalization bound of TENON-DA for domain adaptation.

Theorem 7 (Generalization of TENON-DA). *Suppose that the learning models are instantiated with infinitely wide neural networks, given the hypothesis space $\mathcal{H}$, for any hypothesis $f(\cdot;\theta_k) \in \mathcal{H}$ and any $\delta \in (0,1)$, with probability at least $1-\delta$, the expected error of the target domain can be upper bounded by*

$$\begin{aligned}
\mathbb{E}_{x\sim\mathbb{P}_K}\left[\left\|f(x;\theta_K) - f(x;\theta_K^*)\right\|_2^2\right] \le \zeta\Bigg[&\lambda \sum_{k=1}^{K}\sum_{i=1}^{n_k}\left\|f(x_i^k;\theta_k) - y_i^k\right\|_2^2 \\
&+ \frac{1}{2}\sum_{k=1}^{K}\sum_{i,j=1}^{n_k} s_{ij}^k \left\|\frac{f(x_i^k;\theta_k)}{\sqrt{D_{ii}^k}} - \frac{f(x_j^k;\theta_k)}{\sqrt{D_{jj}^k}}\right\|_2^2 \\
&+ \frac{1}{2}\sum_{k,k'=1}^{K} d_{kk'}\left\|\frac{\theta_k}{\sqrt{M_{kk}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}}\right\|_2^2\Bigg] \\
&+ \frac{1}{n_K L_{\mathcal{R}}}\sum_{k=1}^{K}\Omega(\mathbf{X}_k, \theta_k^*) + \frac{1}{n_K L_{\mathcal{R}}}\Delta(\theta_1^*, \cdots, \theta_K^*) + O\left(\frac{\log(1/\delta)}{n_K}\right)
\end{aligned}$$

where $\Omega(\mathbf{X}_k, \theta_k^*) = \sum_{i,j=1}^{n_k} s_{ij}^k \left\|\frac{f(x_i^k;\theta_k^*)}{\sqrt{D_{ii}^k}} - \frac{f(x_j^k;\theta_k^*)}{\sqrt{D_{jj}^k}}\right\|_2^2$ denotes the label smoothness over $\theta_k^* = \arg\min_{\theta'} \mathbb{E}_{\mathbb{P}_k}[f(x^k;\theta'), y^k]$, $\Delta(\theta_1^*, \cdots, \theta_K^*) = \sum_{k,k'=1}^{K} d_{kk'}\left\|\frac{\theta_k^*}{\sqrt{M_{kk}}} - \frac{\theta_{k'}^*}{\sqrt{M_{k'k'}}}\right\|_2^2$ and $\zeta = \max\left\{\frac{U_{\mathcal{R}}}{\lambda n_K L_{\mathcal{R}}}, \frac{2}{n_K L_{\mathcal{R}}}\right\}$. $L_{\mathcal{R}}$ and $U_{\mathcal{R}}$ are constants depending on the maximum and minimum eigenvalues of $\mathbf{L}_K + \mathbf{K}_{KK}^{-1}$ respectively, where $\mathbf{L}_k$ is the symmetrically normalized Laplacian matrix of the target domain.

Algorithm 2 TENON-OOD

Input: K source (training) domains $\{\mathbb{P}_k\}_{k=1}^K$;
Output: Predicted output values of target samples.

```
———————— Training Stage (Pre-computing) ————————
Calculate ΓK_NTK (same procedures as Lines 2-10 in Alg. 1);
Calculate y* = λK_NTK^{-1}Γ^{-1}y = λ(ΓK_NTK)^{-1}y;
———————— Inference Stage ————————————————
for x_{K+1}^test from target (testing) domain K+1 do
    Calculate neural tangent kernel Φ(x_{K+1}^test, X);
    Calculate y_{K+1}^test = (λ/K) Φ(x_{K+1}^test, X) K_NTK^{-1} Γ^{-1} y;
end for
```

It can be seen from Theorem 7 that the generalization error of TENON-DA on the target domain is determined by the following crucial factors. One is the empirical prediction error given by TENON-DA (see Eq. (1)) over source and target training samples. The other one is the optimal label smoothness within each domain and the optimal parameter smoothness across domains. We would like to point out that previous works study the generalization performance of domain adaptation using either domain discrepancy [1, 55, 57] or label smoothness [35] across domains, by assuming that all domains share the same hypothesis. The learned prediction function in those works might lose domain-specific information, resulting in sub-optimal performance on the target domain. Though some recent works [40, 51] propose to learn both domain-invariant and domain-specific representations, their theoretical generalization performance is unclear. Instead, in this paper, we leverage the simple distributional network of networks framework to model data heterogeneity in domain adaptation with theoretical guarantees (e.g., the first three terms of the upper bound in Theorem 7 result in the optimization framework of Eq. (1) for domain adaptation).

5.2 Inductive Distribution Learning

We can formulate the out-of-distribution generalization [16, 23] as an inductive distribution learning problem. As illustrated in Figure 1(b), all source (training) domains can be used to construct a network of networks. Since the target (testing) domains are only available during the testing phase, they will be added to the main network as new nodes after model training. Therefore, out-of-distribution generalization can be considered as an inductive distributional learning problem, given the formulated network of networks. To solve this problem, we instantiate the proposed TENON framework (denoted as TENON-OOD) with the following training and inference stages (see Algorithm 2).

- *Training Stage*: Given K source (training) domains $\{\mathbb{P}_k\}_{k=1}^K$ each with labeled samples $\{x_i^k, y_i^k\}_{i=1}^{n_k}$, the objective function of TENON-OOD during training can be directly given by Eq. (1). Thus, based on Theorem 4, we can obtain the closed-form solution for model parameters $\{\theta_k\}_{k=1}^K$ over training domains.

$$\mathbf{\Theta}^* = \lambda \nabla_{\mathbf{\Theta}} f_0(\mathbf{X})^T \mathbf{K}_{\text{NTK}}^{-1} \mathbf{\Gamma}^{-1} \mathbf{y}$$

where $\mathbf{\Theta}^* = [\theta_1^{*T}, \cdots, \theta_K^{*T}]^T$. Here θ_k^* denotes the optimized model parameters within domain k.

- *Inference Stage*: In the inference stage, we can learn the model parameters θ^*_{K+1} for a new target (testing) domain $\mathbb{P}_{K+1}$ as follows. For standard out-of-distribution generalization, no prior knowledge regarding the target (testing) domain is available before model inference. In this case, we assume that the new target (testing) domain can be considered as a new (domain) node for the previously derived network of networks. The edge weight [6] between this new node and previous nodes within the main network is simply set as 1. Considering the objective function $\min_{\theta_{K+1}} \sum_{k,k'=1}^{K+1} d_{kk'} \left\| \frac{\theta_k}{\sqrt{M_{kk}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}} \right\|_F^2$, we obtain the closed-form solution $\theta^*_{K+1} = \frac{1}{K}\sum_{k=1}^{K} \theta^*_k$, and thus the predicted class label of any testing sample x^{test}_{K+1} is given by

$$y^{\text{test}}_{K+1} = \frac{\lambda}{K} \mathbf{\Phi}(x^{\text{test}}_{K+1}, \mathbf{X}) \mathbf{K}^{-1}_{\text{NTK}} \mathbf{\Gamma}^{-1} \mathbf{y}$$

where $\mathbf{\Phi}(x^{\text{test}}_{K+1}, \mathbf{X}) = [\mathbf{K}_{\text{NTK}}(x^{\text{test}}_{K+1}, x^1_1), \cdots, \mathbf{K}_{\text{NTK}}(x^{\text{test}}_{K+1}, x^1_{n_1}), \cdots, \mathbf{K}_{\text{NTK}}(x^{\text{test}}_{K+1}, x^1_K), \cdots, \mathbf{K}_{\text{NTK}}(x^{\text{test}}_{K+1}, x^K_{n_K})]$ denotes the neural tangent kernel between x^{test}_{K+1} and samples from training domains.

5.3 More Discussion Regarding Algorithms 1&2

It can be seen that the term $\mathbf{\Gamma}$ in Algorithms 1&2 involves the computationally expensive gradient terms $\nabla_{\mathbf{\Theta}} f_0(X)$.

$$\mathbf{\Gamma} = \hat{\mathbf{A}} + \lambda \mathbb{I} + \mathbf{K}^{-1}_{\text{NTK}} \nabla_{\mathbf{\Theta}} f_0(X) \hat{\mathbf{B}} \nabla_{\mathbf{\Theta}} f_0(X)^T \mathbf{K}^{-1}_{\text{NTK}}$$

However, we have the following observations.

$$\begin{aligned}
&\nabla_{\mathbf{\Theta}} f_0(X) \hat{\mathbf{B}} \nabla_{\mathbf{\Theta}} f_0(X)^T \\
&= \nabla_{\mathbf{\Theta}} f_0(X) \left(\mathbb{I}_{Kd_{in} \times Kd_{in}} - \mathbf{B} \otimes \mathbb{I}_{d_{in} \times d_{in}} \right) \nabla_{\mathbf{\Theta}} f_0(X)^T \\
&= \mathbf{K}_{\text{NTK}} - \begin{bmatrix}
\frac{d_{11}}{\sqrt{M_{11} \cdot M_{11}}} \mathbf{K}_{11} & \frac{d_{12}}{\sqrt{M_{11} \cdot M_{22}}} \mathbf{K}_{12} & \cdots & \frac{d_{1K}}{\sqrt{M_{11} \cdot M_{KK}}} \mathbf{K}_{1K} \\
\frac{d_{21}}{\sqrt{M_{22} \cdot M_{11}}} \mathbf{K}_{21} & \frac{d_{22}}{\sqrt{M_{22} \cdot M_{22}}} \mathbf{K}_{22} & \cdots & \frac{d_{2K}}{\sqrt{M_{22} \cdot M_{KK}}} \mathbf{K}_{2K} \\
\vdots & \vdots & \vdots & \vdots \\
\frac{d_{K1}}{\sqrt{M_{KK} \cdot M_{11}}} \mathbf{K}_{K1} & \frac{d_{K2}}{\sqrt{M_{KK} \cdot M_{22}}} \mathbf{K}_{K2} & \cdots & \frac{d_{KK}}{\sqrt{M_{KK} \cdot M_{KK}}} \mathbf{K}_{KK}
\end{bmatrix}
\end{aligned}$$

It shows that the term $\mathbf{\Gamma}$ in Algorithms 1&2 can be efficiently calculated using the domain similarity $d_{kk'}$ and neural tangent kernel $\mathbf{K}_{kk'}$ between domain k and domain k'.

5.4 Computational Complexity

Algorithms 1&2 show that the time complexity of TENON-DA and TENON-OOD is determined by the calculation of neural tangent kernel (NTK) of any pair of training samples and the inversion of the propagation matrix $\mathbf{\Gamma}\mathbf{K}_{\text{NTK}}$. The time complexity of calculating NTK over all domains is $O(\bar{n}^2)$ [37], where $\bar{n} = \sum_{k=1}^{K} n_k$ denotes the number of all training samples. The inversion of $\mathbf{\Gamma}\mathbf{K}_{\text{NTK}}$ requires $O(\bar{n}^3)$. Following [19], we can use the conjugate gradient method to solve the linear system $(\mathbf{\Gamma}\mathbf{K}_{\text{NTK}})\, \mathbf{y}^* = \lambda \mathbf{y}$, in order to estimate the propagated labels $\mathbf{y}^*$. This allows us to reduce the time complexity from $O(\bar{n}^3)$ to $O(b\bar{n}^2)$, where b is the number of iterations. In this case, we term the variants of TENON-DA and TENON-OOD algorithms with conjugate gradient as TENON-DA-Fast and TENON-OOD-Fast, respectively (see subsection 6.3.3 for more empirical analysis).

[6] Without prior knowledge regarding the unseen target domain, we assume that the unseen target domain is equally similar to all source domains. In this case, only the parameter smoothness regularization (i.e., the third term in Eq. (1)) will be available to optimize the model parameters of this unseen target domain.

6 EXPERIMENTS

In the experiment, we evaluate the proposed TENON algorithms on domain adaptation and out-of-distribution generalization data sets.

6.1 Experimental Setup

6.1.1 Data Sets. We use the following data sets.

- Amazon Review [8]: It contains positive and negative product reviews from four different domains: Books, DVD, Electronics, and Kitchen. Following [47, 57], we use top-5000 frequent unigrams/bigrams to extract the bag-of-words features for Amazon reviews. Each review is associated with a binary label indicating positive or negative sentiment.
- CityCam [54]: CityCam is a large-scale web camera data set. It contains images captured by several cameras in different city locations. Following [11], we use images from four cameras (with IDs: 253, 495, 511, and 572). Each image has a 2048-dimensional feature vector extracted from the pre-trained ResNet-50 [18]. Specifically, in this paper, we consider a binary classification task based on the number of vehicles within the camera images, i.e., whether there are at least 10 cars in an image.
- Huffpost [31]: Huffpost contains article headlines associated with 11 news categories collected from the Huffington Post from 2012 to 2018. Following [53], we use pre-trained DistilBERT [43] to extract a 768-dimensional feature vector for each new headline. The task is to identify the news tags of article headlines as one of the following 11 categories: Black Voices, Business, Comedy, Crime, Entertainment, Impact, Queer Voices, Science, Sports, Tech, Travel.
- ArXiv [10]: ArXiv provides metadata of arXiv preprints from 2007 to 2023. As illustrated in [53], each preprint consists of a paper title and its corresponding primary categories. The paper title can further be represented as a 768-dimensional feature vector using pre-trained DistilBERT [43]. The task of ArXiv is to predict the primary category of arXiv pre-prints from their paper titles.
- CivilComments [22]: CivilComments consists of comments scraped from the internet. It contains 8 demographic identities: male, female, LGBTQ, Christian, Muslim, other religions, Black, or White. Each identity is considered as a single domain. CivilComments involves a binary classification task to determine whether a comment is toxic.

6.1.2 Baselines. In the experiment, we consider the following domain adaptation baselines, including (1) semi-supervised learning: LabelProp [12, 58], and (2) domain adaptation: DANN [14], MDAN [57], M3SAD [39], DARN [47], and GRDA [52]. In addition, we use the following out-of-distribution generalization baselines: ERM, DANN [14], IRM [2], SD [41], Fish [45], and EQRM [13].

6.1.3 Configuration. Following [47], we use a 3-layer multi-layer perceptron (MLP) to instantiate the prediction function for all baselines. Then we implement our proposed algorithms using the

Model	Amazon Review				CityCam			
	Books	DVD	Electronics	Kitchen	253	495	511	572
LabelProp [58]	$0.7078_{\pm 0.0040}$	$0.7294_{\pm 0.0093}$	$0.7699_{\pm 0.0079}$	$0.7799_{\pm 0.0082}$	$0.6741_{\pm 0.1360}$	$0.6312_{\pm 0.0470}$	$0.6753_{\pm 0.0888}$	$0.7560_{\pm 0.0564}$
DANN [14]	$0.6958_{\pm 0.0157}$	$0.7229_{\pm 0.0031}$	$\mathbf{0.7818_{\pm 0.0053}}$	$0.7879_{\pm 0.0072}$	$0.7804_{\pm 0.0415}$	$0.6716_{\pm 0.0499}$	$\mathbf{0.8498_{\pm 0.0136}}$	$0.7563_{\pm 0.0344}$
MDAN [57]	$0.7196_{\pm 0.0095}$	$0.7432_{\pm 0.0205}$	$0.7744_{\pm 0.0121}$	$0.7869_{\pm 0.0156}$	$0.8007_{\pm 0.0579}$	$0.6685_{\pm 0.0339}$	$0.8222_{\pm 0.0227}$	$0.7280_{\pm 0.0261}$
M3SAD [39]	$0.7019_{\pm 0.0232}$	$0.7251_{\pm 0.0210}$	$0.7753_{\pm 0.0117}$	$0.7893_{\pm 0.0134}$	$0.8064_{\pm 0.0581}$	$0.6175_{\pm 0.0286}$	$0.7970_{\pm 0.0357}$	$0.7621_{\pm 0.0531}$
DARN [47]	$0.7175_{\pm 0.0126}$	$0.7412_{\pm 0.0180}$	$0.7703_{\pm 0.0119}$	$0.7888_{\pm 0.0145}$	$0.8243_{\pm 0.0392}$	$0.6795_{\pm 0.0321}$	$0.8271_{\pm 0.0161}$	$0.7547_{\pm 0.0385}$
GRDA [52]	$0.7110_{\pm 0.0099}$	$0.7294_{\pm 0.0110}$	$0.7714_{\pm 0.0091}$	$0.7884_{\pm 0.0056}$	$0.7949_{\pm 0.0751}$	$0.6698_{\pm 0.0340}$	$0.8254_{\pm 0.0196}$	$0.7495_{\pm 0.0389}$
TENON-DA-Fast	$\mathbf{0.7241_{\pm 0.0146}}$	$0.7499_{\pm 0.0101}$	$0.7713_{\pm 0.0093}$	$\mathbf{0.7898_{\pm 0.0101}}$	$\mathbf{0.8359_{\pm 0.0146}}$	$\mathbf{0.7145_{\pm 0.0122}}$	$0.7918_{\pm 0.0104}$	$0.7857_{\pm 0.0176}$
TENON-DA	$0.7238_{\pm 0.0135}$	$\mathbf{0.7503_{\pm 0.0094}}$	$0.7763_{\pm 0.0065}$	$0.7851_{\pm 0.0037}$	$0.8350_{\pm 0.0156}$	$0.7143_{\pm 0.0134}$	$0.7932_{\pm 0.0097}$	$\mathbf{0.7859_{\pm 0.0182}}$

Table 1: Domain adaptation on Amazon review and CityCam data sets

Model	2013	2014	2015	2016	2017	2018	Avg.
LabelProp [58]	$0.5312_{\pm 0.0109}$	$0.2942_{\pm 0.0095}$	$0.2641_{\pm 0.0202}$	$0.3535_{\pm 0.0237}$	$0.3900_{\pm 0.0145}$	$0.5055_{\pm 0.0165}$	0.3897
DANN [14]	$0.4171_{\pm 0.0293}$	$0.3510_{\pm 0.0234}$	$0.3847_{\pm 0.0423}$	$0.4468_{\pm 0.0205}$	$0.4568_{\pm 0.0352}$	$0.5490_{\pm 0.0254}$	0.4342
MDAN [57]	$0.4171_{\pm 0.0293}$	$0.3365_{\pm 0.0481}$	$0.3757_{\pm 0.0546}$	$0.4424_{\pm 0.0207}$	$0.4392_{\pm 0.0267}$	$0.5019_{\pm 0.0322}$	0.4188
M3SAD [39]	$0.4077_{\pm 0.0290}$	$0.3490_{\pm 0.0439}$	$0.4055_{\pm 0.0380}$	$0.4468_{\pm 0.0246}$	$0.4455_{\pm 0.0281}$	$0.4879_{\pm 0.0398}$	0.4237
DARN [47]	$0.4171_{\pm 0.0293}$	$0.3944_{\pm 0.0140}$	$0.3902_{\pm 0.0375}$	$0.4553_{\pm 0.0205}$	$\mathbf{0.4951_{\pm 0.0227}}$	$0.5601_{\pm 0.0106}$	0.4520
GRDA [52]	$0.4324_{\pm 0.0330}$	$0.3671_{\pm 0.0234}$	$0.3539_{\pm 0.0255}$	$0.4534_{\pm 0.0150}$	$0.4520_{\pm 0.0153}$	$0.4987_{\pm 0.0137}$	0.4262
TENON-DA-Fast	$0.5850_{\pm 0.0110}$	$\mathbf{0.5028_{\pm 0.0183}}$	$0.4573_{\pm 0.0275}$	$\mathbf{0.4995_{\pm 0.0129}}$	$0.4556_{\pm 0.0295}$	$\mathbf{0.5201_{\pm 0.0080}}$	0.5033
TENON-DA	$\mathbf{0.5851_{\pm 0.0110}}$	$\mathbf{0.5028_{\pm 0.0183}}$	$\mathbf{0.4575_{\pm 0.0273}}$	$0.4987_{\pm 0.0131}$	$0.4651_{\pm 0.0132}$	$0.5198_{\pm 0.0080}$	**0.5048**

Table 2: Domain adaptation on the Hoffpost data set ("Avg." indicates the average accuracy over all target domains)

Model	2009	2011	2013	2015	2017	2019	2021	Avg.
LabelProp [58]	$0.7006_{\pm 0.0267}$	$0.6938_{\pm 0.0060}$	$0.7186_{\pm 0.0063}$	$0.6780_{\pm 0.0056}$	$0.6717_{\pm 0.0130}$	$0.6857_{\pm 0.0085}$	$0.6741_{\pm 0.0099}$	0.6889
DANN [14]	$0.7483_{\pm 0.0122}$	$0.6943_{\pm 0.0387}$	$0.7187_{\pm 0.0304}$	$\mathbf{0.7283_{\pm 0.0092}}$	$0.7183_{\pm 0.0081}$	$0.7293_{\pm 0.0256}$	$0.7213_{\pm 0.0350}$	0.7226
MDAN [57]	$0.7483_{\pm 0.0122}$	$0.7161_{\pm 0.0075}$	$0.6251_{\pm 0.0819}$	$0.6593_{\pm 0.0535}$	$0.6748_{\pm 0.0254}$	$0.6944_{\pm 0.0410}$	$0.7139_{\pm 0.0223}$	0.6903
M3SAD [39]	$0.5533_{\pm 0.0099}$	$0.6026_{\pm 0.1134}$	$0.6845_{\pm 0.0637}$	$0.7151_{\pm 0.0200}$	$0.6980_{\pm 0.0176}$	$0.7384_{\pm 0.0117}$	$0.7473_{\pm 0.0174}$	0.6770
DARN [47]	$0.7483_{\pm 0.0122}$	$\mathbf{0.7228_{\pm 0.0110}}$	$0.7123_{\pm 0.0302}$	$0.7177_{\pm 0.0196}$	$0.7126_{\pm 0.0151}$	$\mathbf{0.7456_{\pm 0.0054}}$	$0.7471_{\pm 0.0140}$	0.7295
GRDA [52]	$0.7414_{\pm 0.0219}$	$0.7186_{\pm 0.0041}$	$0.6662_{\pm 0.0482}$	$0.6956_{\pm 0.0385}$	$0.6777_{\pm 0.0215}$	$0.7035_{\pm 0.0286}$	$0.7366_{\pm 0.0103}$	0.7056
TENON-DA-Fast	$0.7619_{\pm 0.0098}$	$0.7161_{\pm 0.0064}$	$0.7415_{\pm 0.0071}$	$0.7155_{\pm 0.0070}$	$0.7216_{\pm 0.0034}$	$0.7336_{\pm 0.0076}$	$0.7297_{\pm 0.0124}$	0.7314
TENON-DA	$\mathbf{0.7621_{\pm 0.0100}}$	$0.7157_{\pm 0.0063}$	$\mathbf{0.7420_{\pm 0.0067}}$	$0.7195_{\pm 0.0053}$	$\mathbf{0.7241_{\pm 0.0014}}$	$0.7403_{\pm 0.0078}$	$\mathbf{0.7477_{\pm 0.0096}}$	**0.7359**

Table 3: Domain adaptation on the ArXiv data set ("Avg." indicates the average accuracy over all target domains)

Model	Male	Female	LGBTQ	Christian	Muslim	Others	Black	White	Avg
ERM	$0.6859_{\pm 0.0091}$	$0.6428_{\pm 0.0186}$	$0.6796_{\pm 0.0226}$	$0.7058_{\pm 0.0248}$	$0.7396_{\pm 0.0258}$	$0.7024_{\pm 0.0074}$	$0.7118_{\pm 0.0169}$	$0.6796_{\pm 0.0137}$	0.6934
DANN [14]	$0.6850_{\pm 0.0096}$	$0.6453_{\pm 0.0249}$	$0.6794_{\pm 0.0202}$	$0.7160_{\pm 0.0156}$	$0.7340_{\pm 0.0207}$	$0.6984_{\pm 0.0058}$	$0.6872_{\pm 0.0280}$	$0.6776_{\pm 0.0145}$	0.6904
IRM [2]	$0.6848_{\pm 0.0087}$	$0.6432_{\pm 0.0199}$	$0.6862_{\pm 0.0186}$	$0.7071_{\pm 0.0240}$	$0.7416_{\pm 0.0195}$	$0.7028_{\pm 0.0083}$	$0.7121_{\pm 0.0192}$	$0.6837_{\pm 0.0121}$	0.6951
SD [41]	$0.6777_{\pm 0.0119}$	$0.6449_{\pm 0.0189}$	$0.6847_{\pm 0.0216}$	$0.7068_{\pm 0.0250}$	$0.7464_{\pm 0.0152}$	$\mathbf{0.7031_{\pm 0.0088}}$	$0.7092_{\pm 0.0192}$	$0.6784_{\pm 0.0105}$	0.6939
Fish [45]	$0.6882_{\pm 0.0055}$	$0.6650_{\pm 0.0059}$	$0.6793_{\pm 0.0079}$	$\mathbf{0.7363_{\pm 0.0195}}$	$0.7512_{\pm 0.0117}$	$0.6944_{\pm 0.0125}$	$0.7219_{\pm 0.0071}$	$\mathbf{0.6912_{\pm 0.0105}}$	0.7034
EQRM [13]	$0.6882_{\pm 0.0094}$	$0.6603_{\pm 0.0034}$	$0.6830_{\pm 0.0176}$	$0.7237_{\pm 0.0269}$	$0.7517_{\pm 0.0063}$	$0.6969_{\pm 0.0083}$	$0.7269_{\pm 0.0067}$	$0.6899_{\pm 0.0096}$	0.7025
TENON-OOD-Fast	$\mathbf{0.6922_{\pm 0.0062}}$	$0.6678_{\pm 0.0045}$	$0.6968_{\pm 0.0074}$	$0.7236_{\pm 0.0059}$	$0.7502_{\pm 0.0062}$	$0.7006_{\pm 0.0110}$	$0.7037_{\pm 0.0096}$	$0.6813_{\pm 0.0077}$	0.7020
TENON-OOD	$0.6909_{\pm 0.0082}$	$\mathbf{0.6702_{\pm 0.0030}}$	$\mathbf{0.7044_{\pm 0.0063}}$	$0.7217_{\pm 0.0086}$	$\mathbf{0.7620_{\pm 0.0066}}$	$0.7022_{\pm 0.0089}$	$\mathbf{0.7294_{\pm 0.0055}}$	$0.6909_{\pm 0.0048}$	**0.7089**

Table 4: Out-of-distribution generalization on CivilComments ("Avg." indicates the average accuracy over all testing domains)

NTK [26] induced by a 3-layer MLP with infinite width. The classification accuracy is used as the evaluation metric in the experiments. In addition, we set $\sigma = 2, \lambda = 1$ in our experiments.

6.2 Main Results

In the following, we discuss the evaluation results of TENON algorithms for domain adaptation and out-of-distribution generalization.

6.2.1 Domain Adaptation. Tables 1-3 provide the evaluation comparison between TENON-DA and baselines on various data sets (the best results are indicated in bold). All the experiments are repeated five times and then we report the mean and standard deviation of classification accuracies. For each run, we randomly select 200 samples from each domain as the training samples and others as the testing samples. Specifically, for Amazon Review and CityCam data sets, following [47], we take one domain (e.g., "Books") as the target domain, and others domains (e.g., "DVD", "Electronics" and "Kitchen") as source domains. In contrast, Hoffpost and ArXiv data sets [53] contain evolving domains where the data distribution is changing over time. In this case, we take one specific time stamp as the target domain and all historical time stamps as source domains.

We have the following observations from Tables 1-3. (1) LabelProp considers propagating the label information within a single graph. It does not capture the data heterogeneity among different domains, thus leading to sub-optimal performance in domain adaptation. (2) Compared to domain adaptation baselines, our proposed non-parametric TENON-DA algorithm can achieve superior performance in most cases. This observation verifies the effectiveness of TENON-DA in handling heterogeneous data across domains. (3) TENON-DA-Fast achieves comparable performance with TENON-DA. Furthermore, Figure 3(b) shows that TENON-DA-Fast significantly reduces the running time compared to TENON-DA.

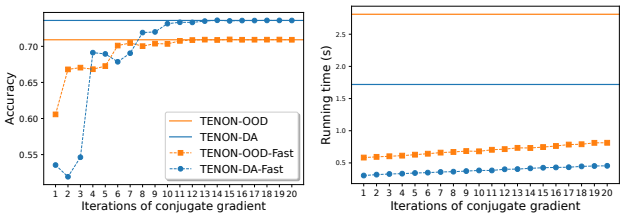
6.2.2 Out-of-Distribution Generalization. Table 4 shows the results of TENON-OOD on the CivilComments data set (the best results are indicated in bold). In this case, we take one domain (e.g., "Male") as the unseen testing target domain and others (e.g., "Female", "LGBTQ", "Christian", "Muslim", "Others", "Black", and "White") as source training domains. It is observed that TENON-OOD outperforms baselines for out-of-distribution generalization.

6.3 Analysis

6.3.1 Ablation Study. Here we study the impact of within-domain label smoothness regularization on the proposed TENON-DA/TENON-OOD algorithms. Table 5 reports the average accuracy of TENON-DA and TENON-OOD on ArXiv and CivilComments respectively. It indicates that the label smoothness regularization improves the model performance. Besides, Figure 3 compares the TENON-DA/TENON-OOD algorithms with their approximation introduced in Subsection 5.4. It can be seen that with only 10 iterations, TENON-DA-Fast/TENON-OOD-Fast based on conjugate gradient can efficiently achieve similar performance with their counterparts.

6.3.2 Hyperparameter Sensitivity. We investigate the impact of hyperparameter λ on the proposed TENON-DA and TENON-OOD algorithms. Figure 4 reports the results of TENON-DA and TENON-OOD on ArXiv and CivilComments respectively. It is observed that both algorithms are robust to the selection of λ.

6.3.3 Efficiency. Figure 5 shows the efficiency comparison between TENON-DA algorithm and baselines, where the overall training running time is reported. It is observed that the proposed TENON-DA algorithm is more computationally efficient than domain adaptation baselines involving gradient descent training. Due to the efficient approximation of matrix inversion, TENON-DA-Fast takes less time than TENON-DA on Amazon Review and ArXiv data sets.



(a) Accuracy (b) Running time

Figure 3: Analysis of conjugate gradient

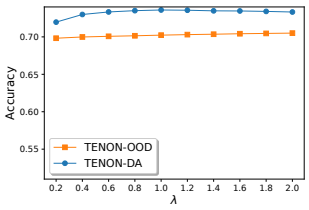


Figure 4: Impact of λ

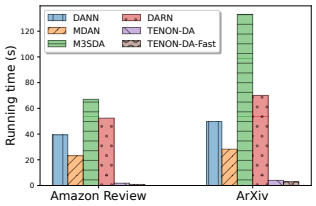


Figure 5: Efficiency analysis

Data	TENON-DA	TENON-DA w/o label smoothness
Amazon Review	0.7589	0.7573
CityCam	0.7821	0.7642
Hoffpost	0.5048	0.4987
ArXiv	0.7359	0.7297
Data	**TENON-OOD**	**TENON-OOD w/o label smoothness**
CivilComments	0.7089	0.7054

Table 5: Ablation study

7 CONCLUSION

In this paper, we propose a generic distributional network of networks (TENON) framework for modeling data heterogeneity, using within-domain label smoothness and cross-domain parameter smoothness. Then we provide two instantiated algorithms of TENON for domain adaptation and out-of-distribution generalization. The effectiveness and efficiency of our proposed algorithms are verified theoretically and empirically.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation (2117902 and 2134079), DARPA (HR001121C0165), and Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] David Acuna, Guojun Zhang, Marc T Law, and Sanja Fidler. 2021. f-Domain Adversarial Learning: Theory and Algorithms. In *ICML*. 66–75.

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

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. MetaReg: Towards domain generalization using meta-regularization. *NeurIPS* 31 (2018).

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning* 79 (2010), 151–175.

[5] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. 2022. AdaMatch: A unified approach to semi-supervised learning and domain adaptation. In *ICLR*.

[6] Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. *NeurIPS* 24 (2011).

[7] Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. 2021. Exploiting domain-specific features to enhance domain generalization. *NeurIPS* 34 (2021), 21189–21201.

[8] Minmin Chen, Zhixiang Xu, Kilian Q Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In *ICML*. 1627–1634.

[9] Qi Chen and Mario Marchand. 2023. Algorithm-Dependent Bounds for Representation Learning of Multi-Source Domain Adaptation. In *AISTATS*. 10368–10394.

[10] Colin B Clement, Matthew Bierbaum, Kevin P O'Keeffe, and Alexander A Alemi. 2019. On the use of arxiv as a dataset. *arXiv preprint arXiv:1905.00075* (2019).

[11] Antoine de Mathelin, Guillaume Richard, François Deheeger, Mathilde Mougeot, and Nicolas Vayatis. 2021. Adversarial weighting for domain adaptation in regression. In *ICTAI*. IEEE, 49–56.

[12] Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. 2005. Efficient non-parametric function induction in semi-supervised learning. In *AISTATS*. 96–103.

[13] Cian Eastwood, Alexander Robey, Shashank Singh, Julius Von Kügelgen, Hamed Hassani, George J Pappas, and Bernhard Schölkopf. 2022. Probable domain generalization via quantile risk minimization. *NeurIPS* 35 (2022), 17340–17358.

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* (2016).

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

[16] Ishaan Gulrajani and David Lopez-Paz. 2021. In search of lost domain generalization. In *ICLR*.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *NeurIPS* 30 (2017).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[19] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label propagation for deep semi-supervised learning. In *CVPR*. 5070–5079.

[20] Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *NeurIPS* 31 (2018).

[21] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. 2012. Undoing the damage of dataset bias. In *ECCV*. 158–171.

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran S. Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*. 5637–5664.

[23] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*.

[24] Ananya Kumar, Tengyu Ma, and Percy Liang. 2020. Understanding self-training for gradual domain adaptation. In *ICML*. 5468–5479.

[25] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. 2019. Wide neural networks of any depth evolve as linear models under gradient descent. *NeurIPS* 32 (2019).

[26] Ronaldas Paulius Lencevicius. 2022. *An Empirical Analysis of the Laplace and Neural Tangent Kernels.* Master's thesis. California State Polytechnic University, Pomona.

[27] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. 2018. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, Vol. 32.

[28] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018. Domain generalization with adversarial feature learning. In *CVPR*. 5400–5409.

[29] Liangyue Li and Hanghang Tong. 2015. The child is father of the man: Foresee the success at the early stage. In *KDD*. 655–664.

[30] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *ICML*. 97–105.

[31] Rishabh Misra. 2022. News Category Dataset. *arXiv preprint arXiv:2209.11429* (2022).

[32] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of machine learning*. MIT press.

[33] Eduardo Fernandes Montesuma and Fred Maurice Ngolè Mboula. 2021. Wasserstein Barycenter for Multi-Source Domain Adaptation. In *CVPR*. 16785–16793.

[34] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *ICML*. 10–18.

[35] Debarghya Mukherjee, Felix Petersen, Mikhail Yurochkin, and Yuekai Sun. 2022. Domain Adaptation meets Individual Fairness. And they get along. *NeurIPS* 35 (2022), 28902–28913.

[36] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. 2014. Inside the atoms: ranking on a network of networks. In *KDD*. 1356–1365.

[37] Roman Novak, Jascha Sohl-Dickstein, and Samuel S Schoenholz. 2022. Fast finite width neural tangent kernel. In *ICML*. 17018–17044.

[38] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* (2010).

[39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *ICCV*. 1406–1415.

[40] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. 2019. Domain agnostic learning with disentangled representations. In *ICML*. 5102–5112.

[41] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. 2021. Gradient starvation: A learning proclivity in neural networks. *NeurIPS* 34 (2021), 1256–1272.

[42] Alexandre Rame, Corentin Dancette, and Matthieu Cord. 2022. Fishr: Invariant gradient variances for out-of-distribution generalization. In *ICML*. 18347–18377.

[43] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[44] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, Vol. 32.

[45] Yuge Shi, Jeffrey Seely, Philip Torr, Siddharth N, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. 2022. Gradient matching for domain generalization. In *ICLR*.

[46] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. *NeurIPS* 31 (2018).

[47] Junfeng Wen, Russell Greiner, and Dale Schuurmans. 2020. Domain aggregation networks for multi-source domain adaptation. In *ICML*. 10214–10224.

[48] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. 2022. A fine-grained analysis on distribution shift. In *ICLR*.

[49] Jun Wu, Wenxuan Bao, Elizabeth Ainsworth, and Jingrui He. 2023. Personalized federated learning with parameter propagation. In *KDD*. 2594–2605.

[50] Jun Wu, Jingrui He, Sheng Wang, Kaiyu Guan, and Elizabeth Ainsworth. 2022. Distribution-informed neural networks for domain adaptation regression. *NeurIPS* 35 (2022), 10040–10054.

[51] Tongkun Xu, Weihua Chen, Pichao WANG, Fan Wang, Hao Li, and Rong Jin. 2022. CDTrans: Cross-domain Transformer for Unsupervised Domain Adaptation. In *ICLR*.

[52] Zihao Xu, Hao He, Guang-He Lee, Bernie Wang, and Hao Wang. 2022. Graph-relational domain adaptation. In *ICLR*.

[53] Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. 2022. Wild-time: A benchmark of in-the-wild distribution shift over time. *NeurIPS* 35 (2022), 10309–10324.

[54] Shanghang Zhang, Guanhang Wu, Joao P Costeira, and Jose MF Moura. 2017. Understanding traffic density from large-scale web camera data. In *CVPR*. 5898–5907.

[55] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019. Bridging theory and algorithm for domain adaptation. In *ICML*. 7404–7413.

[56] Yu Zhang and Dit-Yan Yeung. 2010. A convex formulation for learning task relationships in multi-task learning. In *UAI*. 733–742.

[57] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. 2018. Adversarial multiple source domain adaptation. *NeurIPS* 31 (2018), 8568–8579.

[58] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. *NeurIPS* 16 (2003).

[59] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2021. Domain Generalization with MixStyle. In *ICLR*.

[60] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*. 912–919.

A APPENDIX

In the appendix, we provide the proof of theoretical results presented in the paper.

A.1 Proof of Lemma 3

Proof. The objective of Eq. (1) can be rewritten as follows.

$$\mathcal{J}(\Theta) = \Theta^T \hat{\mathbf{X}}\hat{\mathbf{A}}\hat{\mathbf{X}}^T\Theta + \Theta^T\hat{\mathbf{B}}\Theta + \lambda \cdot \left\|\hat{\mathbf{X}}^T\Theta - \mathbf{y}\right\|_2^2$$

Then the derivative of $\mathcal{J}(\Theta)$ is given by

$$\begin{aligned}\frac{\partial \mathcal{J}(\Theta)}{\partial \Theta} &= \frac{\partial\left(\Theta^T \hat{\mathbf{X}}\hat{\mathbf{A}}\hat{\mathbf{X}}^T\Theta + \Theta^T\hat{\mathbf{B}}\Theta + \lambda \cdot \left\|\hat{\mathbf{X}}^T\Theta - \mathbf{y}\right\|_2^2\right)}{\partial \Theta} \\ &= 2\left(\hat{\mathbf{X}}\hat{\mathbf{A}}\hat{\mathbf{X}}^T + \hat{\mathbf{B}}\right)\Theta + 2\lambda\hat{\mathbf{X}}\hat{\mathbf{X}}^T\Theta - 2\lambda\hat{\mathbf{X}}\mathbf{y} \\ &= 2\left(\hat{\mathbf{X}}\hat{\mathbf{A}}\hat{\mathbf{X}}^T + \hat{\mathbf{B}} + \lambda\hat{\mathbf{X}}\hat{\mathbf{X}}^T\right)\Theta - 2\lambda\hat{\mathbf{X}}\mathbf{y}\end{aligned}$$

By setting $\frac{\partial \mathcal{J}(\Theta)}{\partial \Theta} = 0$, the minimizer of $\mathcal{J}(\Theta)$ is obtained at

$$\begin{aligned}\Theta^* &= \lambda\left(\hat{\mathbf{X}}\hat{\mathbf{A}}\hat{\mathbf{X}}^T + \hat{\mathbf{B}} + \lambda\hat{\mathbf{X}}\hat{\mathbf{X}}^T\right)^{-1}\hat{\mathbf{X}}\mathbf{y} \\ &= \lambda\hat{\mathbf{X}}\left(\hat{\mathbf{A}}\hat{\mathbf{X}}^T\hat{\mathbf{X}} + \left(\hat{\mathbf{X}}^T\hat{\mathbf{X}}\right)^{-1}\left(\hat{\mathbf{X}}^T\hat{\mathbf{B}}\hat{\mathbf{X}}\right) + \lambda\hat{\mathbf{X}}^T\hat{\mathbf{X}}\right)^{-1}\mathbf{y}\end{aligned}$$

which completes the proof. □

A.2 Proof of Theorem 4

Proof. Following [25], we consider the following linearized neural network

$$f_t^{\text{lin}}(x) = f_0(x) + \left(\nabla_{\theta_0} f_0(x)\right)^T (w_t)$$

where $w_t = \theta_t - \theta_0$ is the parameter change from the initial values.

Let $\mathbf{W}_t = \Theta_t - \Theta_0$ be the change of parameters from the initial values and $f_t(X) = \text{vec}(f_t(x_i^k; \theta^k) | i \in \{1, 2, \cdots, n_k\}, k \in \{1, 2, \cdots, K\})$ be the vectorized predicted values over all input samples. Based on continuous time gradient descent [25], the evolution of the parameters can be expressed as

$$\begin{aligned}\dot{\mathbf{W}}_t &= -\frac{\eta}{2}\nabla_\Theta \mathcal{J}(\Theta) = -\frac{\eta}{2}\nabla_\Theta f_0(X)^T\left(\hat{\mathbf{A}}\mathbf{f}_t + \lambda\left(\mathbf{f}_t - \mathbf{y}\right)\right) - \eta\hat{\mathbf{B}}\Theta_t \\ &= -\eta\nabla_\Theta f_0(X)^T\Bigg(\left(\hat{\mathbf{A}}\nabla_\Theta f_0(X) + \lambda\nabla_\Theta f_0(X)\right)\mathbf{W}_t \\ &\quad + \left(\nabla_\Theta f_0(X)\nabla_\Theta f_0(X)^T\right)^{-1}\nabla_\Theta f_0(X)\hat{\mathbf{B}}\mathbf{W}_t \\ &\quad + \left(\nabla_\Theta f_0(X)\nabla_\Theta f_0(X)^T\right)^{-1}\nabla_\Theta f_0(X)\hat{\mathbf{B}}\Theta_0 + \hat{\mathbf{A}}f_0(X) + \lambda f_0(X) - \lambda\mathbf{y}\Bigg)\end{aligned}$$

In this case, the ODE has a closed-form solution below.

$$\begin{aligned}&\Theta_t \\ &= -\nabla_\Theta f_0(X)^T\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\left(\nabla_\Theta f_0(X)\hat{\mathbf{B}}\nabla_\Theta f_0(X)^T\right)\right)^{-1} \\ &\quad \cdot\left(\mathbb{I} - \exp\left\{-\eta\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\left(\nabla_\Theta f_0(X)\hat{\mathbf{B}}\nabla_\Theta f_0(X)^T\right)\right)t\right\}\right) \\ &\quad \cdot\left(\mathbf{K}_{\text{NTK}}^{-1}\nabla_\Theta f_0(X)\hat{\mathbf{B}}\Theta_0 + \hat{\mathbf{A}}f_0(X) + \lambda f_0(X) - \lambda\mathbf{y}\right) + \Theta_0\end{aligned}$$

Thus,

$$\lim_{t\to\infty}\Theta_t = -\nabla_\Theta f_0(\mathbf{X})^T\mathbf{K}_{\text{NTK}}^{-1}\Gamma^{-1}\left(\Omega - \lambda\mathbf{y}\right) + \Theta_0$$

For an arbitrary point x^k, the predicted value is given by

$$\begin{aligned}&f_t^{\text{lin}}(x^k;\theta_k) \\ &= \lambda\cdot\mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\left(\nabla_\Theta f_0(\mathbf{X})\hat{\mathbf{B}}\nabla_\Theta f_0(\mathbf{X})^T\right)\right)^{-1} \\ &\quad\cdot\left(\mathbb{I} - \exp\left\{-\eta\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\left(\nabla_\Theta f_0(\mathbf{X})\hat{\mathbf{B}}\nabla_\Theta f_0(\mathbf{X})^T\right)\right)t\right\}\right)\mathbf{y} \\ &\quad + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\left(\nabla_\Theta f_0(X)\hat{\mathbf{B}}\nabla_\Theta f_0(X)^T\right)\right)^{-1} \\ &\quad\cdot\left(\mathbb{I} - \exp\left\{-\eta\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\left(\nabla_\Theta f_0(X)\hat{\mathbf{B}}\nabla_\Theta f_0(X)^T\right)\right)t\right\}\right) \\ &\quad\cdot\left(\mathbf{K}_{\text{NTK}}^{-1}\nabla_\Theta f_0(X)\hat{\mathbf{B}}\Theta_0 + \hat{\mathbf{A}}f_0(X) + \lambda f_0(X)\right)\end{aligned}$$

Thus, it holds that

$$\begin{aligned}\lim_{t\to\infty} f_t(x^k;\theta_k) = \lim_{t\to\infty} f_t^{\text{lin}}(x^k;\theta_k) &= \lambda\mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\mathbf{K}_{\text{NTK}}^{-1}\Gamma^{-1}\mathbf{y} \\ &\quad + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\mathbf{K}_{\text{NTK}}^{-1}\Gamma^{-1}\Omega\end{aligned}$$

which completes the proof. □

A.3 Proof of Corollary 5

Proof. In this case, it holds that $\hat{\mathbf{B}} = \mathbf{0}_{Kd_\theta\times Kd_\theta}$. Then for any sample x^k, it holds

$$\begin{aligned}&f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}} + \mathbf{K}_{\text{NTK}}^{-1}\nabla_\Theta f_0(X)\hat{\mathbf{B}}\nabla_\Theta f_0(X)^T\right)^{-1} \\ &\quad\cdot\left(\mathbf{K}_{\text{NTK}}^{-1}\nabla_\Theta f_0(\mathbf{X})\hat{\mathbf{B}}\Theta_0 + \hat{\mathbf{A}}f_0(\mathbf{X}) + \lambda f_0(\mathbf{X})\right) \\ &= f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}}\right)^{-1}\left(\hat{\mathbf{A}}f_0(\mathbf{X}) + \lambda f_0(\mathbf{X})\right) \\ &= f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\mathbf{K}_{\text{NTK}}^{-1}f_0(\mathbf{X}) \\ &= f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}_k)\mathbf{K}_{kk}^{-1}f_0(\mathbf{X}_k)\end{aligned}$$

Thus,

$$\begin{aligned}\lim_{t\to\infty} f_t(x^k;\theta_k) &= \lambda\mathbf{K}_{\text{NTK}}(x^k,\mathbf{X})\left(\hat{\mathbf{A}}\mathbf{K}_{\text{NTK}} + \lambda\mathbf{K}_{\text{NTK}}\right)^{-1}\mathbf{y} \\ &\quad + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}_k)\mathbf{K}_{kk}^{-1}f_0(\mathbf{X}_k) \\ &= \lambda\left[\mathbf{0},\cdots,\mathbf{0},\mathbf{K}(x^k,\mathbf{X}_k),\mathbf{0},\cdots,\mathbf{0}\right]\left(\left(\hat{\mathbf{A}} + \lambda\mathbb{I}\right)\mathbf{K}_{\text{NTK}}\right)^{-1}\mathbf{y} \\ &\quad + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}_k)\mathbf{K}_{kk}^{-1}f_0(\mathbf{X}_k) \\ &= \lambda\mathbf{K}\left(x^k,\mathbf{X}_k\right)\mathbf{K}_{kk}^{-1}\left(\lambda\mathbb{I}_{n_k\times n_k} + \mathbb{I}_{n_k\times n_k} - \mathbf{A}_k\right)^{-1}\mathbf{y}_k \\ &\quad + f_0(x^k;\theta_k) - \mathbf{K}_{\text{NTK}}(x^k,\mathbf{X}_k)\mathbf{K}_{kk}^{-1}f_0(\mathbf{X}_k)\end{aligned}$$

For training samples $\mathbf{X}_k = [x_1^k,\cdots,x_{n_k}^k]$, the following holds

$$\lim_{t\to\infty} f_t\left(\mathbf{X}_k;\theta_k\right) = \frac{\lambda}{\lambda+1}\left(\mathbb{I}_{n_k\times n_k} - \frac{1}{\lambda+1}\mathbf{A}_k\right)^{-1}\mathbf{y}_k$$

which completes the proof. □

A.4 Proof Corollary 6

Proof. With linearized model $f(x_i^k;\theta_k) = \nabla_\theta f_0(x_i^k)^T\theta_k$, the objective of Eq. (1) can be rewritten as follows.

$$\mathcal{J}(\Theta) = \frac{1}{2}\sum_{k,k'=1}^{K} d_{kk'}\left\|\frac{\theta_k}{\sqrt{M_{kk}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}}\right\|_2^2 + \lambda\sum_{k=1}^{K}\sum_{i=1}^{n_k}\left\|f(x_i^k;\theta_k) - y_i^k\right\|_2^2$$

Using gradient descent, for any k, it holds

$$\theta_k(t+1) = \theta_k(t) - \eta\frac{\partial\mathcal{J}(\Theta)}{\partial\theta_k}$$

$$
\begin{aligned}
&= \theta_k(t) - \eta\Bigg(\theta_k(t) - \sum_{k'=1}^{K} \frac{d_{kk'}}{\sqrt{M_{kk}M_{k'k'}}}\theta_{k'}(t) \\
&\quad + \lambda\left(\nabla f(\mathbf{X}_k)^T \nabla f(\mathbf{X}_k)\theta_k(t) - \nabla f(\mathbf{X}_k)^T \mathbf{y}_k\right)\Bigg) \\
&= \Big((1-\eta)\mathbb{I} - \lambda \nabla f(\mathbf{X}_k)^T \nabla f(\mathbf{X}_k)\Big)\theta_k(t) \\
&\quad + \eta \sum_{k'=1}^{K} \frac{d_{kk'}}{\sqrt{M_{kk}M_{k'k'}}}\theta_{k'}(t) + \eta\lambda \nabla f(\mathbf{X}_k)^T \mathbf{y}_k
\end{aligned}
$$

which completes the proof. □

A.5 Proof of Theorem 7

Proof. Following standard machine learning theory [32], given hypothesis space $\mathcal{H}$ and the loss function is bounded by B (for any x, $|f(x;\theta_k) - f(x;\theta_k^*)| \le B$), then for any $f(\cdot;\theta_K) \in \mathcal{H}$,

$$
\begin{aligned}
&\mathbb{E}_{x\sim\mathbb{P}_K}\left[\left(f\left(x;\theta_K\right) - f\left(x;\theta_K^*\right)\right)^2\right] \\
&\le \frac{1}{n_K}\sum_{i=1}^{n_K}\left(f\left(x_i^K;\theta_K\right) - f\left(x_i^K;\theta_K^*\right)\right)^2 + B\sqrt{\frac{\log|\mathcal{H}| + \log(2/\delta)}{2n_K}}
\end{aligned}
$$

where $|\mathcal{H}|$ is the size of the hypothesis space and can be further bounded by the VC dimension of hypothesis space $\mathcal{H}$. It holds

$$
\begin{aligned}
&\frac{1}{n_K}\sum_{i=1}^{n_K}\left(f\left(x_i^K;\theta_K\right) - f\left(x_i^K;\theta_K^*\right)\right)^2 \\
&\le \frac{1}{n_K}\left\|f(\mathbf{X}_K;\theta_K) - g(f(\mathbf{X}_S;\theta_S))\right\|^2 + \frac{1}{n_K}\left\|g(f(\mathbf{X}_S;\theta_S)) - g(f(\mathbf{X}_S;\theta_S^*))\right\|^2 \\
&\quad + \frac{1}{n_K}\left\|g(f(\mathbf{X}_S;\theta_S^*)) - f(\mathbf{X}_K;\theta_K^*)\right\|^2 \\
&\le \frac{1}{n_K}\frac{U_{\mathcal{R}}}{L_{\mathcal{R}}}\left\|f(\mathbf{X}_S;\theta_S) - f(\mathbf{X}_S;\theta_S^*)\right\|^2 + \frac{1}{n_K}\frac{2}{L_{\mathcal{R}}}\mathcal{R}\Big(f(\mathbf{X}_S;\theta_S), f(\mathbf{X}_K;\theta_K)\Big) \\
&\quad + \frac{1}{n_K}\frac{2}{L_{\mathcal{R}}}\mathcal{R}\Big(f(\mathbf{X}_S;\theta_S^*), f(\mathbf{X}_K;\theta_K^*)\Big) \\
&\le \zeta\Bigg(\lambda\sum_{k=1}^{K}\sum_{i=1}^{n_k}\left\|f(x_i^k;\theta_k) - y_i^k\right\|_2^2 + \frac{1}{2}\sum_{k=1}^{K}\sum_{i,j=1}^{n_k} s_{ij}^k\left\|\frac{f(x_i^k;\theta_k)}{\sqrt{D_{ii}^k}} - \frac{f(x_j^k;\theta_k)}{\sqrt{D_{jj}^k}}\right\|_2^2 \\
&\quad + \frac{1}{2}\sum_{k,k'=1}^{K} d_{kk'}\left\|\frac{\theta_k}{\sqrt{M_{kk}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}}\right\|_2^2\Bigg) \\
&\quad + \frac{1}{n_K L_{\mathcal{R}}}\sum_{k=1}^{K}\Omega(\mathbf{X}_k,\theta_k^*) + \frac{1}{n_K L_{\mathcal{R}}}\Delta(\theta_1^*,\cdots,\theta_K^*)
\end{aligned}
$$

where $\Delta(\theta_1^*,\cdots,\theta_K^*) = \sum_{k,k'=1}^{K} d_{kk'}\left\|\frac{\theta_k^*}{\sqrt{M_{kk}}} - \frac{\theta_{k'}^*}{\sqrt{M_{k'k'}}}\right\|_2^2$ and $\zeta = \max\left\{\frac{U_{\mathcal{R}}}{\lambda n_K L_{\mathcal{R}}}, \frac{2}{n_K L_{\mathcal{R}}}\right\}$.

Note that in previous steps, we let

$$
g(\hat{\mathbf{y}}_s) = \arg\min_{\hat{\mathbf{y}}_t}\mathcal{R}\Big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\Big) = \frac{1}{2}\sum_{k=1}^{K}\sum_{i,j=1}^{n_k} s_{ij}^k\left\|\frac{f(x_i^k;\theta_k)}{\sqrt{D_{ii}^k}} - \frac{f(x_j^k;\theta_k)}{\sqrt{D_{jj}^k}}\right\|_2^2
$$

$$
+ \frac{1}{2}\sum_{k,k'=1}^{K} d_{kk'}\left\|\frac{\theta_k}{\sqrt{M_{kk}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}}\right\|_2^2
$$

where $\hat{\mathbf{y}}_s = f(\mathbf{X}_s;\theta_s)$. We assume that $\mathcal{R}$ is strongly convex and smooth. For any $\theta_s,\theta_t,\theta_t' \in \mathbb{R}^{d_\theta}$, the following holds

$$
\begin{aligned}
\mathcal{R}\Big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\Big) &\ge \mathcal{R}\Big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t'\Big) + \Big\langle \hat{\mathbf{y}}_t - \hat{\mathbf{y}}_t', \partial_{\hat{\mathbf{y}}_t'}\mathcal{R}\big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t'\big)\Big\rangle + \frac{L_{\mathcal{R}}}{2}\left\|\hat{\mathbf{y}}_t - \hat{\mathbf{y}}_t'\right\|_2^2 \\
\mathcal{R}\Big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\Big) &\le \mathcal{R}\Big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t'\Big) + \Big\langle \hat{\mathbf{y}}_t - \hat{\mathbf{y}}_t', \partial_{\hat{\mathbf{y}}_t'}\mathcal{R}\big(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t'\big)\Big\rangle + \frac{U_{\mathcal{R}}}{2}\left\|\hat{\mathbf{y}}_t - \hat{\mathbf{y}}_t'\right\|_2^2
\end{aligned}
$$

Then

$$
\begin{aligned}
&\left\|g\left(\hat{\mathbf{y}}_s\right) - g\left(\hat{\mathbf{y}}_s^*\right)\right\|_2^2 \le \frac{U_{\mathcal{R}}}{L_{\mathcal{R}}}\left\|\hat{\mathbf{y}}_s - \hat{\mathbf{y}}_s^*\right\| = \frac{U_{\mathcal{R}}}{L_{\mathcal{R}}}\left\|f(\mathbf{X}_s;\theta_s) - f(\mathbf{X}_s;\theta_s^*)\right\| \\
&\left\|\hat{\mathbf{y}}_t - g\left(\hat{\mathbf{y}}_s\right)\right\|^2 \le \frac{2}{L_{\mathcal{R}}}\mathcal{R}\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)
\end{aligned}
$$

Next, following [35], we show the strong convexity and smoothness of $\mathcal{R}$.

$$
\begin{aligned}
\frac{\partial\mathcal{R}\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\theta_t} &= \frac{\partial\hat{\mathbf{y}}_t}{\theta_t}\cdot\frac{\partial\mathcal{R}\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\hat{\mathbf{y}}_t} \\
\frac{\partial\mathcal{R}\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\theta_t} &= \nabla f(\mathbf{X}_k)^T\left(\mathbf{I} - \mathbf{A}_K\right)\nabla f(\mathbf{X}_k)\theta_t \\
&\quad + \sum_{k'=1}^{K}\frac{d_{Kk'}}{\sqrt{M_{KK}}}\left(\frac{\theta_t}{\sqrt{M_{KK}}} - \frac{\theta_{k'}}{\sqrt{M_{k'k'}}}\right) \\
&= \nabla f(\mathbf{X}_k)^T\left(\mathbf{I} - \mathbf{A}_K\right)\nabla f(\mathbf{X}_k)\theta_t + \theta_t \\
&\quad - \sum_{k'=1}^{K}\frac{d_{Kk'}}{\sqrt{M_{k'k'}\sqrt{M_{KK}}}}\theta_{k'} \\
\frac{\partial\mathcal{R}^2\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\theta_t^2} &= \nabla f(\mathbf{X}_k)^T\left(\mathbf{I} - \mathbf{A}_K\right)\nabla f(\mathbf{X}_k) + \mathbf{I} \\
&= \nabla f(\mathbf{X}_k)^T\left(\mathbf{I} - \mathbf{A}_K + \left(\nabla f(\mathbf{X}_K)\nabla f(\mathbf{X}_K)^T\right)^{-1}\right)\nabla f(\mathbf{X}_k) \\
\frac{\partial\hat{\mathbf{y}}_t}{\theta_t} &= \nabla f(\mathbf{X}_k)^T \in \mathbb{R}^{d_\theta\times n_K} \\
\frac{\partial}{\partial\theta_t}\left(\frac{\partial\hat{\mathbf{y}}_t}{\theta_t}\cdot\frac{\partial\mathcal{R}\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\hat{\mathbf{y}}_t}\right) &= \nabla f(\mathbf{X}_k)^T\frac{\partial\mathcal{R}^2\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\hat{\mathbf{y}}_t\cdot\partial\theta_t} \\
&= \nabla f(\mathbf{X}_k)^T\frac{\partial\mathcal{R}^2\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\hat{\mathbf{y}}_t\cdot\partial\hat{\mathbf{y}}_t}f(\mathbf{X}_k)
\end{aligned}
$$

Thus,

$$
\frac{\partial\mathcal{R}^2\left(\hat{\mathbf{y}}_s,\hat{\mathbf{y}}_t\right)}{\partial\hat{\mathbf{y}}_t\cdot\partial\hat{\mathbf{y}}_t} = \mathbf{L}_K + \left(\nabla f(\mathbf{X}_K)\nabla f(\mathbf{X}_K)^T\right)^{-1} = \mathbf{L}_K + \mathbf{K}_{KK}^{-1}
$$

where $\mathbf{L}_K = \mathbf{I} - \mathbf{A}_K$ is a symmetrically normalized Laplacian matrix, and $\mathbf{K}_{KK}$ is the neural tangent kernel (NTK) matrix within the target domain. Thus, $L_{\mathcal{R}}$ and $U_{\mathcal{R}}$ are given by the maximum and minimum eigenvalues of $\mathbf{L}_K + \mathbf{K}_{KK}^{-1}$. □