HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Global-Local Regularization Via Distributional Robustness

Phan Viet Hoang
hoang.pv180086@sis.hust.edu.vn

Major: Information Technology
Specialization: Computer Science

Supervisor: MSc. Ngo Van Linh _____

Signature

Department: Computer Science

School: School of Information and Communication Technology

HANOI, 08/2022

ACKNOWLEDGMENT

First and foremost, I would like to express my sincere gratitude to subject teachers at Hanoi University of Science and Technology for their continuous guidance throughout my study, especially ones from School of Information and Communication Technology. During the past three and a half years, I have learned from them needed knowledge for pursuing later stages in my Computer Science journey.

I thankfully acknowledge the support and inspiration to pursue a career in research that I receive from my respected supervisor, MSc. Ngo Van Linh, and Prof. Than Quang Khoat, who accepted me into the Data Science Lab. Although in-depth research in Machine Learning did not suit me at first, the step-by-step guidance and valuable experience gained here helped me a lot in publishing quality papers. My appreciation also extends to my laboratory colleagues, I'm very happy to have had the opportunity to collaborate with all of you, especially Nguyen Van Son (UT Austin) who always provide me with insightful thinking into every aspect of research.

Secondly, transferring from Hanoi University of Science and Technology to VinAI Research, I am grateful to all the scientists and residents for maintaining this high-quality research environment. My first gratitude goes to Prof. Le Thanh Huong (HUST), who spent time writing the letter of recommendation and referred me to this wonderful program. I owe a great debt of gratitude to Dr. Le Minh Trung (Monash) and Prof. Ho Pham Minh Nhat (UT Austin) for mentoring me during the writing of the various papers composing this thesis. Over the last year, they have provided me the opportunity to work on many exciting projects. Without their support (both professionally and personally), it would not have been possible for this work to progress as far as it has.

Last but not least, this thesis is dedicated to my parents for the two decades of your love and support. I am also very grateful to my girlfriend, Thu Hang, for her unwavering moral support. I would not have come this far without your loving weight behind me.

ABSTRACT

Despite superior performance in many situations, deep neural networks are often vulnerable to adversarial examples and distribution shifts, limiting model generalization ability in real-world applications. To alleviate these problems, recent approaches leverage distributional robustness optimization (DRO) to find the most challenging distribution, and then minimize loss function over this most challenging distribution.

Regardless of having achieved some improvements, these DRO approaches have some obvious limitations. First, they purely focus on local regularization to strengthen model robustness, missing a global regularization effect that is useful in many real-world applications (e.g., domain adaptation, domain generalization, and adversarial machine learning). Second, the loss functions in the existing DRO approaches operate in only the most challenging distribution, hence decouple with the original distribution, leading to a restrictive modeling capability.

In this thesis, we propose a novel regularization technique, following the veins of Wasserstein-based DRO framework. Specifically, we define a particular joint distribution and Wasserstein-based uncertainty, allowing us to couple the original and most challenging distributions for enhancing modeling capability and applying both local and global regularizations. Empirical studies on different learning problems demonstrate that our proposed approach significantly outperforms the existing regularization approaches in various domains.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
AML	Adversarial machine learning
DA	Domain adaptation
DG	Domain generalization
DR	Distributional robustness
DRO	Distributional robustness optimization
OT	Optimal Transport
PGD	Projected gradient descent
RKHS	Reproducing kernel Hilbert space
SSL	Semi-supervised learning
SVGD	Stein Variational Gradient Descent
VAT	Virtual adversarial training
WS	Wasserstein

NOTATIONS

Notation	Meaning
x, X	If X is a random variable, then x is a realization of X
$\mathbb{E}(X)$	Mean or expectation of a random variable X
supp($\mathbb{P}$)	Support of a distribution $\mathbb{P}$
$T\#\mu$	Pushforward measure of μ by T
KL	Kullback–Leibler divergence
tr(X)	The trace of a square matrix X
X_{ij}	the entry (or element) on the i-th row and j-th column of X
$\sup A$	Supremum of a set A
$\inf A$	Infimum of a set A
pdf	Probability density function
cdf	Cumulative distribution function
w.r.t	With respect to
LHS	Left-hand side
RHS	Right-hand side

CHAPTER 1. INTRODUCTION

Despite achieving state-of-the-art performance on a broad range of applications, deep neural networks are often vulnerable to adversarial examples and distribution shifts, limiting model generalization ability in many real-world problems [1], [2]. To relieve these issues, many regularization techniques have been proposed, e.g. VAT [3], ADT [4], ME-ADA [5], Mixup [6], Cutout [7] and Cutmix [8]. In this thesis, we consider distributional robustness (DR), which is an emerging framework for improving model robustness and regularization ability, that seeks the worst-case expected loss among a ball of distributions, containing all distributions that are close to the empirical distribution [9].

As the Wasserstein distance is a powerful and convenient tool of measuring closeness between distributions, Wasserstein DR has been one of the most widely-used variants of DR. Here we consider a generic Polish space S endowed with a distribution $\mathbb{P}$. Let $r : S \to \mathbb{R}$ be a real-valued (risk) function and $c : S \times S \to \mathbb{R}_+$ be a cost function. Distributional robustness setting aims to find the distribution $\tilde{\mathbb{P}}$ in the vicinity of $\mathbb{P}$ and maximizes the risk in the expectation form [10], [11]:

$$\sup_{\tilde{\mathbb{P}}:\mathcal{W}_c\left(\mathbb{P},\tilde{\mathbb{P}}\right)<\epsilon} \mathbb{E}_{\tilde{Z}\sim\tilde{\mathbb{P}}}\left[r\left(\tilde{Z}\right)\right], \tag{1.1}$$

where $\epsilon > 0$ and $\mathcal{W}_c\left(\mathbb{P}, \tilde{\mathbb{P}}\right) := \inf_{\gamma\in\Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right)} \int c d\gamma$ denotes an optimal transport (OT) or a Wasserstein (WS) distance with the set of couplings $\Gamma\left(\mathbb{P}, \tilde{\mathbb{P}}\right)$ whose marginals are $\mathbb{P}$ and $\tilde{\mathbb{P}}$.

Direct optimization over the set of distributions $\tilde{\mathbb{P}}$ is often computationally intractable except in limited cases, we thus seek to cast this problem into its dual form instead. With the assumption that $r \in L^1(\mathbb{P})$ is upper semi-continuous and the cost c is a non-negative and continuous function satisfying $c(Z, \tilde{Z}) = 0$ iff $Z = \tilde{Z}$, [10], [11] showed the *dual* form for Eq. (1.1) is:

$$\inf_{\lambda\geq 0}\left\{\lambda\epsilon + \mathop{\mathbb{E}}_{Z\sim\mathbb{P}}[\sup_{\tilde{Z}}\left\{r\left(\tilde{Z}\right) - \lambda c\left(\tilde{Z}, Z\right)\right\}]\right\}. \tag{1.2}$$

When applying DR to the supervised learning setting, $\tilde{Z} = \left(\tilde{X}, \tilde{Y}\right)$ is a pair of data/label drawn from $\tilde{\mathbb{P}}$ and r is the loss function [10], [11]. The fact that r engages only $\tilde{Z} = \left(\tilde{X}, \tilde{Y}\right) \sim \tilde{\mathbb{P}}$ certainly restricts the modeling capacity of (1.2). The reasons are as follows. Firstly, for each anchor Z, the most challenging sample $\tilde{Z}$ is currently

defined as the one maximizing $\sup_{\tilde{Z}} \left(r(\tilde{Z}) - \lambda c\left(Z, \tilde{Z}\right)\right)$, where $r(\tilde{Z})$ is inherited from the primal form (1.1). Hence, it is not suitable to express the risk function r engaging both Z and $\tilde{Z}$ (e.g., Kullback-Leibler divergence $KL\left(p\left(\tilde{Z}\right) \| p(Z)\right)$ between the predictions for Z and $\tilde{Z}$ as in TRADES [12]). Secondly, it is also *impossible* to inject a *global regularization term* involving a batch of samples $\tilde{Z}$ and Z.

Contribution. To empower the formulation of DR for efficiently tackling various real-world problems, in this work, we propose a rich OT based DR framework, named ***G**lobal-**L**ocal **O**ptimal **T**ransport based **D**istributional **R**obustness* (GLOT-DR). Specifically, by designing special joint distributions $\mathbb{P}$ and $\tilde{\mathbb{P}}$ together with some constraints, our framework is applicable to a mixed variety of real-world applications, including domain generalization (DG), domain adaptation (DA), semi-supervised learning (SSL), and adversarial machine learning (AML). Additionally, our GLOT-DR makes it possible for us to equip not only a *local regularization term* for enforcing a local smoothness and robustness, but also a *global regularization term* for imposing a global effect targeting a downstream task. Moreover, by designing a specific WS distance, we successfully develop a closed-form solution for GLOT-DR without using the dual form in [10], [11] (i.e., Eq. (1.2)). Technically, our solution turns solving the inner maximization in the dual form (1.2) into sampling a set of challenging particles according to a local distribution, on which we can handle efficiently using Stein Variational Gradient Decent (SVGD) [13] approximate inference algorithm. Based on the general framework of GLOT-DR, we establish the settings for DG, DA, SSL, and AML and conduct experiments to compare our GLOT-DR to state-of-the-art baselines in these real-world applications. Overall, our contributions can be summarized as follows:

- We enrich the general framework of DR to make it possible for many real-world applications by enforcing both local and global regularization terms. Here we note that the global regularization term is crucial for many downstream tasks (see Section 4.0.1 for more details).
- We propose a closed-form solution for our GLOT-DR without involving the dual form in [10], [11] (i.e., equation (1.2)). Here we note that the dual form (1.2) is *not computationally convenient* to solve due to the minimization over λ.
- We conduct comprehensive experiments to compare our GLOT-DR to state-of-the-art baselines in DG, DA, SSL, and AML. The experimental results demonstrate the merits of our proposed approach and empirically prove that

both of the introduced local and global regularization terms advance existing methods across various scenarios, including DG, DA, SSL, and AML.

CHAPTER 2. RELATED WORK

Distributional robustness (DR). DR is an attractive framework for improving machine learning models in terms of robustness and generalization. The underlying idea of DR is to find the *most challenging distribution* around a given distribution and then challenge a model with this distribution. To characterize the closeness of a distribution to a center distribution, either a f-divergence [14]–[18] or Wasserstein distance [19]–[23] can be employed. Other works [10], [11] developed a dual form for DR, opening the door to incorporate DR into the training of deep learning models.

Adversarial Robustness (AR). Neural networks are generally vulnerable to adversarial attacks, notably FGSM [24], PGD [25], and Auto-Attack [26]. Among various kinds of defense approaches, Adversarial Training (AT), originating in [24], has drawn the most research attention. Given its effectiveness and efficiency, many variants of AT have been proposed with: (1) different types of adversarial examples (e.g., the worst-case examples [24] or most divergent examples [12]), (2) different searching strategies (e.g., non-iterative FGSM and Rand FGSM [25]), (3) additional regularization (e.g., adding constraints in the latent space [27], [28]). Inspired by the potential of DR, it has been applied to enhance model robustness in [3], [4], [11], [29]–[31].

Transfer Learning (TL). Domain adaptation (DA) and domain generalization (DG) are two typical settings in TL. As for domain adaptation, [32]–[35] aim at training a model based on a labeled source domain to adapt to an unlabeled target domain, while the works in DG [36]–[41] aim at training a model based on multiple labeled source domains to predict well on unseen target domains. Finally, in more recent work, it was leveraged with DG in [5] and DA in [42].

CHAPTER 3. BACKGROUND

This section will cover fundamental knowledge about optimal transport, Wasserstein distance, Stein Variational Gradient Descent . . . that lays the foundation of all of our work discussed in the subsequent chapters. They are essential for understanding this thesis.

3.1 Optimal transport and Wasserstein distance

Monge problem: Let Ω_S and Ω_T be two separate probability spaces and $\mu \in \mathcal{P}(\Omega_S), \nu \in \mathcal{P}(\Omega_T)$ be two probability measures. Formally, given a geodesic cost function $c : \Omega_S \times \Omega_T \rightarrow \mathbb{R}^+$, the Monge problem [43] seeks a transportation plan (i.e. a push-forward operator) $\mathcal{T} : \Omega_S \rightarrow \Omega_T$ between μ and ν that minimizes the total cost:

$$\inf_{\mathcal{T}\#\mu=\nu} \int_{\Omega_S} c(x, \mathcal{T}(x)) d\mu(x) \tag{3.1}$$

In the case where such a transportation plan exists, it is called optimal transport map, which associates each point from Ω_S from exactly one point in Ω_T. However, the existence of the optimal transport map is not always guaranteed, for instance, when μ is characterized by one Dirac while ν is defined by two. As such, Kantorovich proposed a relaxed version of the original Monge problem by expanding its feasible solution space [44].

Kantorovich problem: Rather than seeking an one-to-one mapping $\mathcal{T}$, Kantorovich problem is to search for a probabilistic coupling $\gamma \in \mathcal{P}(\Omega_S \times \Omega_T)$, where $\mathcal{P}(\Omega_S \times \Omega_T)$ denotes the space of all join distributions. By casting the problem in Equation 3.1 into an optimization problem over the set of couplings $\Pi(\mu, \nu) = \{\gamma \in \mathcal{P}(\Omega \times \Omega) \mid \pi_1 \# \gamma = \mu, \pi_2 \# \gamma = \nu\}$, the optimal solution γ_0 is guaranteed to be existed [45]:

$$\gamma_0 = \inf_{\gamma \in \Pi(\mu,\nu)} \int_{\Omega_S \times \Omega_T} c\left(\mathbf{x}^s, \mathbf{x}^t\right) d\gamma\left(\mathbf{x}^s, \mathbf{x}^t\right) \tag{3.2}$$

Wasserstein Distance: For any real number $p \geq 1$, the p -Wasserstein distance between μ and ν that is characterized by the $||\cdot||_p$ norm, is defined in the following way:

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Pi(\mu,\nu)} \int_{\Omega_S \times \Omega_T} ||\mathbf{x}^s - \mathbf{x}^t||^p d\gamma\left(\mathbf{x}^s, \mathbf{x}^t\right) \right)^{\frac{1}{p}} \tag{3.3}$$

The Wasserstein distance was proven to be a better way to quantify how distant two distributions are, address the limitations of existing divergence:

- **KL (Kullback–Leibler) divergence:**

$$D_{KL}(P\|Q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

- **f-divergence:**

$$D_f(P\|Q) = \int f\left(\frac{p(x)}{q(x)}\right) q(x) dx$$

- **JS (Jensen-Shannon) divergence:**

$$D_{JS}^{\alpha,\beta}(P,Q) = \alpha D_{KL}(P\|\alpha P + \beta Q) + \beta D_{KL}(Q\|\alpha P + \beta Q)$$

as: (i) they only divergence, not metric/distance since they do not follow metric properties (positive, symmetric and triangle inequality), (ii) they are not able to compute the distance between two distributions that are not in the same probability space.

We here provide a comparison of Wasserstein distance against these three measures, let's consider the example presented in [46], which is two uniform distributions at 0 and θ

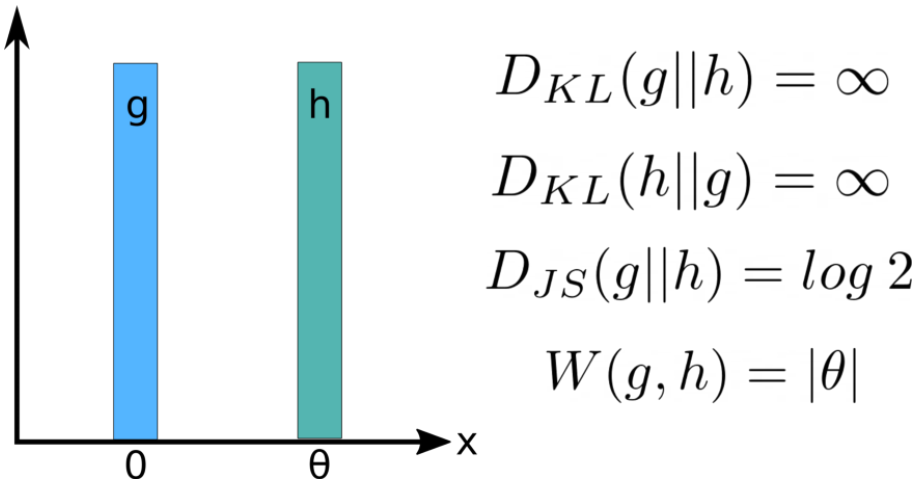

Figure 3.1: Wasserstein distance vs commonly-used divergences.

The above figure shows an example of a non-sharing probability space. Both distributions are not overlapping and have a distance of θ between them.

The KL-divergence between these two distributions will be infinity because of non-overlapping and the denominator part will be zero. Similarly, JS-divergence will be constant (log 2) irrespective of the value of θ as one of them will be zero at the given x.

In contrast, the Wasserstein metric depends on the distance between these two distributions i.e. which is desirable and considering the horizontal shifts irrespective of overlaps. Therefore, the Wasserstein metric is more practical and produces better results compared to JS-divergence.

c-transform: Given a cost function $c : \Omega_S \times \Omega_T \to \mathbb{R}^+$ the c-transform (or c-conjugate function) of a function $\phi : \Omega_S \to \mathbb{R}$ is defined on Ω_T as:

$$\phi^c(y) \triangleq \inf_{x \in \Omega_S} \{c(x, y) - \phi(x)\}$$

Similarly, the $\bar{c}$-transform of a function $\psi : \Omega_T \to \mathbb{R}$ is defined on as:

$$\psi^{\bar{c}}(x) \triangleq \inf_{y \in \Omega_T} \{c(x, y) - \psi(y)\}$$

Kantorovich problem - dual form: The dual form of the Kantorovich problem, whose primal form is given above:

$$\sup \left\{ \int_{\Omega_S} \phi(x) \mathrm{d}\mu(x) + \int_{\Omega_T} \psi(y) \mathrm{d}\mu(y) \right\}$$

where the supremum is taken over all functions $\phi : \Omega_S \to \mathbb{R}$ and $\psi : \Omega_T \to \mathbb{R}$ that satisfy $\phi(x) + \psi(y) \leq c(x, y)$ for all $(x, y) \in \Omega_S \times \Omega_T$. The solution of the dual problem above is called **Kantorovich potential**.

3.2 Stein's Method

The discussion in this section is based on [13], [47]. Stein's method [48] provides a general framework to determine the bound of the distance between any two distributions.

Stein variational gradient descent: SVGD is a general purpose method to approximate a target distribution P with p.d.f $p(\theta)$ by particles that are first randomly sampled from an initial simple distribution Q_0 with p.d.f $q_0(\theta)$ and then moved step by step towards P via a series of simple invertible transformations. An advantage of SVGD is that it requires only any unnormalised version of the target probability density.

We now consider the problem of finding an approximation of a target distribution $p(\theta)$. We denote the approximate distribution as $q(\theta)$. We denote the approximate distribution as $q(\theta)$ (with an additional index that will be made clear in the context). The target distribution is approximated via a set of samples $\theta_1, \ldots, \theta_M$.

Initially, we draw M particles $\theta_1, \ldots, \theta_M$ from an initial distribution $q_0(\theta)$ that is

usually simple and easy to sample from. e. In general, we want to learn a transformation T^* that transports the initial density q_0 to the target density p, but this is usually considered as a challenging task. According to SVGD framework, the task of learning T^* is decomposed into learning a sequence of simple invertible mappings $T^{(1)}, \ldots, T^{(L)}$

$$T^*(\theta) = T^{(L)}\left(T^{(L-1)}\left(\ldots T^{(1)}(\theta)\right)\right)$$

wherein at each step, the induced distribution $q_{[T]}$ is pushed closer to the target distribution. The authors of SVGD proposed to use the perturbation of identity mapping as $T(\theta) \triangleq \theta + \epsilon\Phi(\theta)$ where $\Phi(\theta)$ is a smooth function in the function class $\mathcal{F}$ and $|\epsilon|$ is a small real number.

In an arbitrary transformation step let q be the current approximate density and $T\#q$ be the pushforward measure of the distribution q via the mapping T, to find the optimal mapping T we need to solve the optimization problem:

$$\min_{\Phi \in \mathcal{F}} KL\left(T_\# q \| p\right)$$

where $KL\left(T_\# q \| p\right)$ is the KL divergence from the distribution $T\#q$ to distribution p.

It is worth noting that the transformation T defined above is injective when ϵ is sufficiently small [13]. We consider $KL\left(q^{[T]} \| p\right)$ as a function w.r.t. ϵ, by applying the first-order Taylor expansion at 0, we have:

$$KL\left(q^{[T]} \| p\right) = KL\left(q \| p\right) + \nabla_\epsilon KL\left(q^{[T]} \| p\right)\Big|_{\epsilon=0} \epsilon + O\left(\epsilon^2\right),$$

where $\lim_{\epsilon \to 0} O\left(\epsilon^2\right)/\epsilon^2 = const.$

Similar to [13], the gradient $\nabla_\epsilon KL\left(q^{[T]} \| p\right)|_{\epsilon=0}$ can be calculated as follows, with all proofs and derivations can be found in the appendix.

$$\nabla_\epsilon KL\left(q^{[T]} \| p\right)\Big|_{\epsilon=0} = -\left\langle \phi, \psi \right\rangle_{\mathcal{H}_k^d}, \tag{3.4}$$

where $\psi(\cdot) = \mathbb{E}_{\theta \sim q}\left[k(\theta, \cdot)\nabla_\theta \log p(\theta) + \nabla_\theta k(\theta, \cdot)\right]$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}_k^d}$ is the dot product in the RKHS.

3.3 Domain adaptation

Domain adaptation is the task of adapting models across domains. This is motivated by the challenge where the test and training datasets fall from different data distributions due to some factor. Domain adaptation aims to build machine learning models that can be generalized into a target domain and dealing with the discrepancy across domain distributions. Compared to conventional methods, which learn shared feature subspaces or reuse important source instances with shallow representations, deep domain adaptation methods leverage deep networks to learn more transferable representations by embedding domain adaptation in the pipeline of deep learning.

Formally, in domain adaptation, domains can be considered as three main parts: input or feature space $\mathcal{X}$, output or label space $\mathcal{Y}$ and an associated probability distribution $p(x, y)$, i.e., $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, p(x, y)\}$. Feature space $\mathcal{X}$ is a subset of a d-dimensional space, $\mathcal{X} \subset \mathbb{R}^d, \mathcal{Y}$ refers to either a space of binary $\{-1, +1\}$ or multi-class $\{1, \dots K\}$, where K is the number of classes, and $p(x, y)$ is a joint probability distribution over the feature-label space pair $\mathcal{X} \times \mathcal{Y}$. We can decompose the joint probability distribution as $p(x, y) = p(x)p(x \mid y)$ or $p(x, y) = p(y)p(y \mid x)$, where $p(.)$ is a marginal distribution and $p(. \mid .)$ is a conditional distribution.

Given a source domain $\mathcal{S}$ and a target domain $\mathcal{T}$, the source dataset samples drawn from the source domain consist of feature-label pairs $\{(x_i, y_i)\}_{i=1}^{n}$, where n is the number of samples in the source data set. Similarly, the target data set can be denoted as $\{z_i, u_i\}_{i=1}^{m}$, where (z_i, u_i) refers to the target samples and their associated labels. In unsupervised domain adaptation where the labels are not available in the target domain, u is unknown. When the source and target domains are related but from different distributions, naively extending the underlying knowledge contained in one domain into another might negatively affect the learner's performance in the target domain. Therefore, domain adaptation was proposed to tackle this problem by reducing the disparity across domains and further training a model that performs well on the target samples.

3.4 Semi-supervised learning

It has repeatedly been shown that deep neural networks can achieve human- or super-human-level performance on certain supervised learning problems by leveraging large collections of labeled data. However, these successes come at a cost: Creating these large datasets typically requires a great deal of human effort (to manually label examples), pain and/or risk (for medical datasets involving invasive tests) or financial expense (to hire annotators or build the infrastructure needed for domain-specific data collection). For many practical problems and applications, we lack the

resources to create a sufficiently large labeled dataset, which limits the wide-spread adoption of deep learning techniques.

An attractive approach towards addressing the lack of data is semi-supervised learning (SSL) [6]. In contrast with supervised learning algorithms, which require labels for all examples, SSL algorithms can improve their performance by also using unlabeled examples. SSL algorithms generally provide a way of learning about the structure of the data from the unlabeled examples, alleviating the need for labels. recent results [49], [50] have shown that in certain cases, SSL approaches the performance of purely supervised learning, even when a substantial portion of the labels in a given dataset has been discarded.

In supervised learning, we are given a training dataset of input-target pairs $(x, y) \in \mathcal{D}$ sampled from an unknown joint distribution $p(x, y)$. Our goal is to produce a prediction function $f_\theta(x)$ parametrized by θ which produces the correct target y for previously unseen samples from $p(x)$. For example, choosing θ might amount to optimizing a loss function which reflects the extent to which $f_\theta(x) = y$ for $(x, y) \in \mathcal{D}$. In SSL we are additionally given a collection of unlabeled input datapoints $x \in \mathcal{D}_{UL}$, sampled from $p(x)$. We hope to leverage the data from $\mathcal{D}_{UL}$ to produce a prediction function which is more accurate than what would have been obtained by using $\mathcal{D}$ on its own.

From a broad perspective, the goal of SSL is to use $\mathcal{D}_{UL}$ to augment $f_\theta(x)$ with information about the structure of $p(x)$. For example, $\mathcal{D}_{UL}$ can provide hints about the shape of the data "manifold" which can produce a better estimate of the decision boundary between different possible target values.

3.5 Domain generalization

Machine learning systems generally assume that the training and testing distributions are the same. To this end, a key requirement is to develop models that can generalize to unseen distributions. Domain generalization (DG), i.e., out-of-distribution generalization, has attracted increasing interests in recent years. Domain generalization deals with a challenging setting where one or several different but related domain(s) are given, and the goal is to learn a model that can generalize to an unseen test domain.

In many applications, target data is difficult to obtain or even unknown before deploying the model. For example, in biomedical applications where domain shift occurs between different patients' data, it is impractical to collect each new patient's data in advance; in traffic scene semantic segmentation it is infeasible to collect data capturing all different scenes and under all possible weather conditions ; when

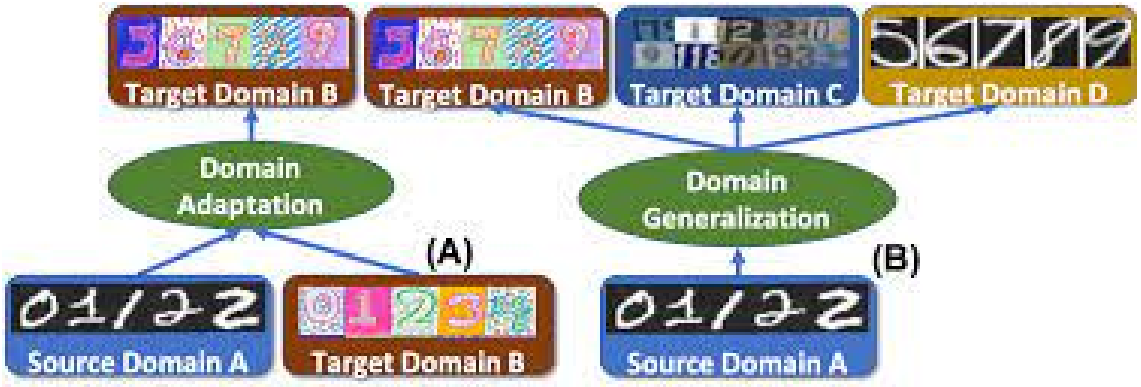

Figure 3.2: Comparison between domain adaptation and domain generalization

dealing with data stream, the model is also required to be intrinsically generalizable.

Let $\mathcal{X}$ be the input (feature) space and $\mathcal{Y}$ the target (label) space, a domain is defined as a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$.[1] For a specific domain P_{XY}, we refer to P_X as the marginal distribution on X, $P_{Y|X}$ the posterior distribution of Y given X, and $P_{X|Y}$ the class-conditional distribution of X given Y.

In the context of DG, we have access to K similar but distinct source domains $\mathcal{S} = \left\{S_k = \left\{\left(x^{(k)}, y^{(k)}\right)\right\}\right\}_{k=1}^{K}$, each associated with a joint distribution $P_{XY}^{(k)}$. Note that $P_{XY}^{(k)} \neq P_{XY}^{(k')}$ with $k \neq k'$ and $k, k' \in \{1, \ldots, K\}$. The goal of DG is to learn a predictive model $f : \mathcal{X} \to \mathcal{Y}$ using only source domain data, such that the prediction error on an unseen target domain $\mathcal{T} = \left\{x^{\mathcal{T}}\right\}$ is minimized. The corresponding joint distribution of the target domain $\mathcal{T}$ is denoted by $P_{XY}^{\mathcal{T}}$. Also, $P_{XY}^{\mathcal{T}} \neq P_{XY}^{(k)}, \forall k \in \{1, \ldots, K\}$.

DG has typically been studied under two different settings, namely multi-source DG ($K \geq 2$) and single source DG ($K = 1$).

3.6 Adversarial Machine Learning

Machine learning models are complicated things and, often, we can have a poor understanding of how they make predictions. This can leave hidden weaknesses that could be exploited by attackers. They could trick the model into making incorrect predictions or give away sensitive information. Fake data could even be used to corrupt models without us knowing. The field of adversarial machine learning aims to address these weaknesses.

To set the stage for our discussion, let us briefly introduce adversarial examples. By now, most researchers in ML have probably seen a picture like the following:

On the left, we have an image of a pig that is correctly classified as such by a state-of-the-art convolutional neural network. After perturbing the image slightly (every

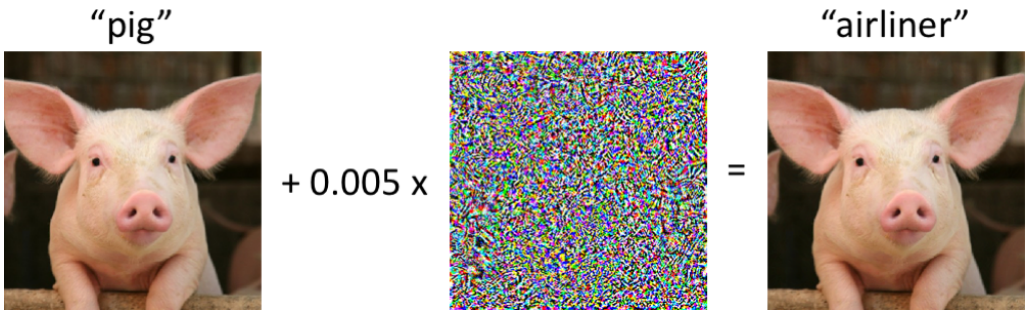

Figure 3.3: An adversarial input, overlaid on a typical image, can cause a classifier to miscategorize a pig as an airliner.

pixel is in the range [0, 1] and changed by at most 0.005), the network now returns class “airliner” with high confidence. Such attacks on trained classifiers have been studied since at least 2004 , and there has already been work on adversarial examples for image classification in 2006 . This phenomenon has then received considerably more attention starting in 2013 when it was pointed out that neural networks are also vulnerable to such attacks (see here and here). Since then, many researchers have proposed ways to construct adversarial examples, as well as methods to make classifiers more robust against adversarial perturbations. It is important to keep in mind though that we do not need to go to neural networks to observe adversarial examples.

Seeing the airliner-pig above might be a bit disturbing at first. However, one should note that the underlying classifier (an Inception-v3 network) is not as fragile as it might seem. While the network misclassifies the perturbed pig with high confidence, this occurs only for specifically crafted perturbations - the network is significantly more robust to random perturbations of similar magnitude. So a natural question is whether it is actually the adversarial perturbations that are fragile. If they crucially rely on precise control over all input pixels, adversarial examples become less of a concern for classifying images in real-world settings.

Recent work shows that this is not the case: the perturbations can be made robust to various channel effects in concrete physical scenarios. For instance, you can print adversarial examples with a standard office printer so that pictures of them taken with a smartphone camera are still misclassified. It is also possible to create stickers that cause neural networks to misclassify various real-world scenes.

As we all know, training a classifier is often formulated as finding model parameters θ that minimize an empirical loss function for a given set of samples $x_1, \ldots, x_n$:

$$\min_{\theta} \sum_{x} \text{loss}(x, \theta)$$

So to cause a misclassification for a fixed model θ and "benign" input x, a natural approach is to find a bounded perturbation δ such that the loss on $x + \delta$ is as large as possible:

$$\max_{\delta} \text{loss}(x + \delta, \theta)$$

In this section, we first introduce the GLOT-DR framework and provide the theoretical guarantees in Section 4.0.1. Then Section 4.0.2 presents the general training procedure of our proposed approach, and the detailed formulations of scenarios are described in the remainder of this section.

4.0.1 Our Framework

We propose a regularization technique based on optimal transport DR, widely applied to many settings including i) *semi-supervised learning*, ii) *domain adaptation*, iii) *domain generalization*, and iv) *adversarial machine learning* . In what follows, we present the general setting along with the notations used throughout this thesis and technical details of our framework.

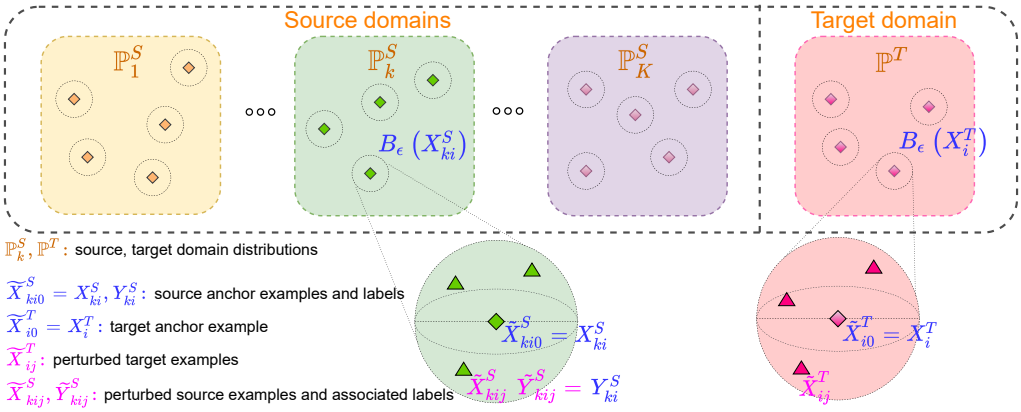

Figure 4.1: Overview of GLOT-DR. We sample $\left[X_{ki}^S, Y_{ki}^S\right]_{i=1}^{B_k^S}$ for each source domain, $\left[X_i^T\right]_{i=1}^{B^T}$ for the target domain, and define $Z, \tilde{Z}$ as in Eqs. (4.1,4.2). For $\left(Z, \tilde{Z}\right) \sim \gamma$ satisfying $\mathbb{E}_\gamma\left[\rho\left(Z, \tilde{Z}\right)\right]^{1/q} \leq \epsilon$, we have $\tilde{X}_{ki0}^S = X_{ki0}^S = X_{ki}^S$, $\tilde{X}_{i0}^T = X_{i0}^T = X_i^T$. Besides, $\tilde{X}_{kij}^S$ with $j \geq 1$ can be viewed as the *perturbed* examples in the ball $B_\epsilon\left(X_{ki}^S\right)$, which have the same label Y_{ki}^S. Similarly, $\tilde{X}_{ij}^T$ with $j \geq 1$ can be viewed as the *perturbed* examples in the ball $B_\epsilon\left(X_i^T\right)$.

Assume that we have *multiple labeled source domains* with the *data/label* distributions $\left\{\mathbb{P}_k^S\right\}_{k=1}^K$ and a *single unlabeled target domain* with the *data* distribution $\mathbb{P}^T$. For the k-th source domain, we draw a batch of B_k^S examples as $\left(X_{ki}^S, Y_{ki}^S\right) \overset{\text{iid}}{\sim} \mathbb{P}_k^S$, where $i = 1, \ldots, B_k^S$ is the sample index. Meanwhile, for the target domain, we sample a batch of B^T examples as $X_i^T \overset{\text{iid}}{\sim} \mathbb{P}^T$, $i = 1, \ldots, B^T$. It is worth noting that for the DG setting, we set $B^T = 0$ (i.e., not use any target data in training).

Furthermore, we examine the multi-class classification problem with the label set $\mathcal{Y} := \{1, ..., M\}$. Hence, the prediction of a classifier is a prediction probability belonging to the *label simplex* $\Delta_M := \left\{\pi \in \mathbb{R}^M : \|\pi\|_1 = 1 \text{ and } \pi \geq \mathbf{0}\right\}$. Finally, let $f_\psi = h_\theta \circ g_\phi$ with $\psi = (\phi, \theta)$ be parameters of our deep net, wherein g_ϕ is the feature extractor and h_θ is the classifier on top of feature representations.

Constructing Challenging Samples: As explained below, our method involves the construction of a random variable Z with distribution $\mathbb{P}$ and another random variable $\tilde{Z}$ with distribution $\tilde{\mathbb{P}}$, "containing" anchor samples $\left(X^S_{ki}, Y^S_{ki}\right), X^T_i$ and their perturbed counterparts $\left(\tilde{X}^S_{kij}, \tilde{Y}^S_{kij}\right), \tilde{X}^T_{ij}$ (see Figure 4.1 for the illustration). The inclusion of both anchor samples and perturbed samples allows us to define a unifying cost function containing local regularization, global regularization, and classification loss.

Concretely, we first start with the construction of Z, containing repeated anchor samples as follows:

$$Z := \left[\left[\left[X^S_{kij}, Y^S_{kij}\right]_{k=1}^{K}\right]_{i=1}^{B^S_k}\right]_{j=0}^{n^S}, \left[\left[X^T_{ij}\right]_{i=1}^{B^T}\right]_{j=0}^{n^T}. \tag{4.1}$$

Here, each source sample is repeated $n^S + 1$ times $\left(X^S_{kij}, Y^S_{kij}\right) = \left(X^S_{ki}, Y^S_{ki}\right), \forall j$, while each target sample is repeated $n^T + 1$ times $X^T_{ij} = X^T_i, \forall j$. The corresponding distribution of this random variable is denoted as $\mathbb{P}$. In contrast to Z, we next define random variable $\tilde{Z} \sim \tilde{\mathbb{P}}$, whose form is

$$\tilde{Z} := \left[\left[\left[\tilde{X}^S_{kij}, \tilde{Y}^S_{kij}\right]_{k=1}^{K}\right]_{i=1}^{B^S_k}\right]_{j=0}^{n^S}, \left[\left[\tilde{X}^T_{ij}\right]_{i=1}^{B^T}\right]_{j=0}^{n^T}. \tag{4.2}$$

Here we note that for $\tilde{X}^S_{kij}$, the index k specifies the k-th source domain, the index i specifies an example in the k-th source batch, while the index j specifies the j-th perturbed example to the source example X^S_{ki}. Similarly, for $\tilde{X}^T_{ij}$, the index i specifies an example in the target batch, while the index j specifies the j-the perturbed example to the target example X^T_i.

We would like $\tilde{Z}$ to contain both: i) anchor examples, i.e., $\left(\tilde{X}^S_{ki0}, \tilde{Y}^S_{ki0}\right) = \left(X^S_{ki}, Y^S_{ki}\right)$ and $\tilde{X}^T_{i0} = X^T_i$; ii) n^S perturbed source samples $\left\{\left(\tilde{X}^S_{kij}, \tilde{Y}^S_{kij}\right)\right\}_{j=1}^{n^S}$ to $\left(X^S_{ki}, Y^S_{ki}\right)$ and n^T perturbed target samples $\left\{\tilde{X}^T_{ij}\right\}_{i=1}^{n^T}$ to X^T_i. In order to impose this requirement, we only consider sampling $\tilde{Z}$ from distribution $\tilde{\mathbb{P}}$ inside the Wasserstein-ball of $\mathbb{P}$,

i.e., satisfying $\mathcal{W}_\rho\left(\mathbb{P}, \tilde{\mathbb{P}}\right) := \inf_{\gamma\in\Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right)} \mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{\frac{1}{q}} \leq \epsilon$, where the cost metric ρ is defined as

$$\begin{aligned}\rho\left(Z,\tilde{Z}\right) := \infty \sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\left\|X_{ki0}^S - \tilde{X}_{ki0}^S\right\|_p^q + \infty\sum_{i=1}^{B^T}\left\|X_{i0}^T - \tilde{X}_{i0}^T\right\|_p^q \\ + \sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=1}^{n^S}\left\|X_{kij}^S - \tilde{X}_{kij}^S\right\|_p^q + \sum_{i=1}^{B^T}\sum_{j=1}^{n^T}\left\|X_{ij}^T - \tilde{X}_{ij}^T\right\|_p^q \\ + \infty\sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=0}^{n^S}\rho_l\left(Y_{kij}^S, \tilde{Y}_{kij}^S\right),\end{aligned}$$

where ρ_l is a metric on the *label simplex* Δ_M. Here we slightly abuse the notion by using $Y \in \mathcal{Y}$ to represent its corresponding one-hot vector. By definition, this cost metric almost surely: i) enforces all 0-th (i.e., $j = 0$) samples in $\tilde{Z}$ to be anchor samples, i.e., $\tilde{X}_{ki0}^S = X_{ki0} = X_{ki}$; ii) allows perturbations on the input data, i.e., $\tilde{X}_{kij}^S \neq X_{ki}^S$ and $\tilde{X}_{ij}^T \neq X_i^T$, for $\forall j \neq 0$; iii) restricts perturbations on labels, i.e., $Y_{kij}^S = \tilde{Y}_{kij}^S$ for $\forall j$ (see Figure 4.1 for the illustration).

Learning Robust Classifier: Upon clear definitions of $\tilde{Z}$ and $\tilde{\mathbb{P}}$, we wish to learn good representations and regularize the classifier f_ψ, via the following distributional robustness problem:

$$\min_{\theta,\phi}\max_{\tilde{\mathbb{P}}:\mathcal{W}_\rho\left(\mathbb{P},\tilde{\mathbb{P}}\right)\leq\epsilon} \mathbb{E}_{\tilde{Z}\sim\tilde{\mathbb{P}}}\left[r\left(\tilde{Z};\phi,\theta\right)\right]. \tag{4.3}$$

The cost function $r\left(\tilde{Z};\phi,\theta\right) := \alpha r^l\left(\tilde{Z};\phi,\theta\right) + \beta r^g\left(\tilde{Z};\phi,\theta\right) + \mathcal{L}\left(\tilde{Z};\phi,\theta\right)$ with $\alpha, \beta > 0$ is defined as the weighted sum of a *local-regularization function* $r^l\left(\tilde{Z};\phi,\theta\right)$, a *global-regularization function* $r^g\left(\tilde{Z};\phi,\theta\right)$, and the *loss function* $\mathcal{L}\left(\tilde{Z};\phi,\theta\right)$, whose explicit forms are dependent on the task (DA, SSL, DG, and AML). Intuitively, the optimization in Eq. (4.3) iteratively searches for the worst-case $\tilde{\mathbb{P}}$ w.r.t. the cost $r\left(\cdot;\phi,\theta\right)$, then changes the network f_ψ to minimize the worst-case cost.

Let us define $\Gamma_\epsilon := \left\{\gamma : \gamma \in \cup_{\tilde{\mathbb{P}}}\Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right) \text{ and } \mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} \leq \epsilon\right\}$, and show that the inner max problem in Eq. (4.3) is equivalent to searching in Γ_ϵ.

Lemma 1. *The optimization problem in Eq. (4.3) is equivalent to the following optimization problem:*

$$\min_{\theta,\phi}\max_{\gamma:\in\Gamma_\epsilon} \mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[r\left(\tilde{Z};\phi,\theta\right)\right]. \tag{4.4}$$

To tackle the optimization problem (OP) in Eq. (4.4), we add the entropic regularization and arrive at the following OP:

$$\min_{\theta,\phi} \max_{\gamma:\in\Gamma_\epsilon} \left\{ \mathbb{E}_{(Z,\tilde{Z})\sim\gamma} \left[r\left(\tilde{Z};\phi,\theta\right)\right] + \frac{1}{\lambda}\mathbb{H}\left(\gamma\right) \right\}, \tag{4.5}$$

where $\lambda > 0$ is the entropic regularization parameter and $\mathbb{H}$ returns the entropy of a given distribution.

The following theorem indicates the optimal solution of the inner max in the OP in Eq. (4.5).

Theorem 2. *Assuming* $r\left(\tilde{Z};\psi\right) = \alpha r^l\left(\tilde{Z};\psi\right) + \beta r^g\left(\tilde{Z};\psi\right) + \mathcal{L}\left(\tilde{Z};\psi\right)$ *with* $\psi = (\phi,\theta)$. *In addition,* Z *and* $\tilde{Z}$ *are constructed as in Eq.(4.1) and Eq.(4.2), respectively. Let* ℓ *denote the loss function, so the expected classification loss becomes*

$$\mathcal{L}\left(\tilde{Z};\psi\right) := \sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=0}^{n_k^S} \ell\left(\tilde{X}_{kij}^S, \tilde{Y}_{kij}^S;\psi\right).$$

Moreover, let the global-regulazation $r^g\left(\tilde{Z};\psi\right) := r^g\left(\left[\tilde{X}_{ki0}^S\right]_{k,i}, \left[\tilde{X}_{i0}^T\right]_i;\psi\right)$ *depend only on anchor samples, while the local-regularization depend on the differences between anchor samples and perturbed samples,*

$$r^l\left(\tilde{Z};\psi\right) := \sum_{i=1}^{B^T}\sum_{j=1}^{n^T} s\left(\tilde{X}_{i0}^T, \tilde{X}_{ij}^T;\psi\right) + \sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=1}^{n_k^S} s\left(\tilde{X}_{ki0}^S, \tilde{X}_{kij}^S;\psi\right),$$

where $s\left(\tilde{X}_0, \tilde{X}_j;\psi\right)$ *measures the difference between 2 input samples, and* $s\left(X, X;\psi\right) = 0, \forall X$. *To this end, the inner max in the OP when* $q = \infty$ *has the following solution*

$$\begin{aligned}\gamma^*\left(Z,\tilde{Z}\right) = &\prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S} p_k^S\left(X_{ki}^S, Y_{ki}^S\right)\prod_{i=1}^{B^T}\prod_{j=0}^{n^T} p^T\left(X_i^T\right)\\ &\prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S} q_{ki}^S\left(\tilde{X}_{kij}^S \mid X_{ki}^S Y_{ki}^S;\psi\right)\prod_{i=1}^{B^T}\prod_{j=1}^{n^T} q_i^T\left(\tilde{X}_{ij}^T \mid X_i^T;\psi\right),\end{aligned} \tag{4.6}$$

where $B_\epsilon\left(X\right) := \left\{X' : \left\|X' - X\right\|_p \le \epsilon\right\}$ *is the* ϵ*-ball around* X, $\left(X_{ki}^S, Y_{ki}^S\right)_{i=1}^{B_k^S} \overset{iid}{\sim} \mathbb{P}_k^S, \forall k$, $X_{1:B^T}^T \overset{iid}{\sim} \mathbb{P}^T$, p_k^S *is the density function of* $\mathbb{P}_k^S$, p^T *is the density function of* $\mathbb{P}^\mathbb{T}$, $q_{ki}^S\left(\tilde{X}_{kij}^S \mid X_{ki}^S Y_{ki}^S;\psi\right) \propto \exp\left\{\lambda[\alpha s(X_{ki}^S, \tilde{X}_{kij}^S;\psi) + \ell(\tilde{X}_{kij}^S, Y_{ki}^S;\psi)]\right\}$ ***is the local distribution over*** $B_\epsilon\left(X_{ki}^S\right)$ *around the anchor example* X_{ki}^S, *and* $q_i^T\left(\tilde{X}_{ij}^T \mid X_i^T;\psi\right) \propto \exp\left\{\lambda\alpha s\left(X_i^T, \tilde{X}_{ij}^T;\psi\right)\right\}$ ***is the local distribution over*** $B_\epsilon\left(X_i^T\right)$ *around the anchor example* X_i^T.

The optimal γ^* in Eq. (4.6) involves the local distributions q_{ki}^S around the anchor example X_{ki}^S and q_i^T around the anchor example X_i^T. By substituting the optimal solution in Eq. (4.6) back to Eq. (4.4), we reach the following OP with $\psi = (\phi, \theta)$:

$$\min_{\psi} \mathbb{E}_{\forall k:(X_{ki}^S, Y_{ki}^S)_{i=1}^{B_k^S} \overset{\text{iid}}{\sim} \mathbb{P}_k^S, X_{1:B^T}^T \overset{\text{iid}}{\sim} \mathbb{P}^T} \left[r\left(\tilde{Z}; \psi\right) \right], \tag{4.7}$$

where $r\left(\tilde{Z}; \psi\right)$ is defined as

$$\begin{aligned} &\mathbb{E}_{\left[\tilde{X}_{kij}^S\right]_j \sim q_{ki}^S} \left[\alpha s(X_{ki}^S, \tilde{X}_{kij}^S; \psi) + \ell(\tilde{X}_{kij}^S, Y_{ki}^S; \psi) \right] \\ &\qquad + \mathbb{E}_{\left[\tilde{X}_{ij}^T\right]_j \sim q_i^T} \left[\alpha s\left(X_i^T, \tilde{X}_{ij}^T; \psi\right) \right] + \beta r^g \left(\left[X_{ki}^S\right]_{k,i}, \left[X_i^T\right]_i; \psi \right), \end{aligned} \tag{4.8}$$

with the *local distribution* q_{ki}^S over $B_\epsilon\left(X_{ki}^S\right)$ and the *local distribution* q_i^T over $B_\epsilon\left(X_i^T\right)$.

It is worth noting how flexible the global-regularization function $r^g \left(\left[X_{ki}^S\right]_{k,i}, \left[X_i^T\right]_i; \psi \right)$ is in enforcing various characteristics suitable for the task, e.g., bridging the distribution shift between source and target domains in DA, between labeled and unlabeled portions in SSL, and between benign and adversarial data examples in AML, as well as learning domain invariant features in DG. Moreover, our global and local regularization terms can be naturally applied to the latent space induced by the feature extractor g_ϕ. Additionally, the theory development for this case is similar to that for the data space except replacing X in the data space by $g_\phi(X)$ in the latent space.

4.0.2 Training Procedure of Our Approach

In what follows, we present how to solve the OP in Eq. (4.7) efficiently. Accordingly, we first need to sample $\left(X_{ki}^S, Y_{ki}^S\right)_{i=1}^{B_k^S} \overset{\text{iid}}{\sim} \mathbb{P}_k^S, \forall k$ and $X_{1:B^T}^T \overset{\text{iid}}{\sim} \mathbb{P}^T$. For each source anchor $\left(X_{ki}^S, Y_{ki}^S\right)$, we sample $\left[\tilde{X}_{kij}^S\right]_{j=1}^{n^S} \overset{\text{iid}}{\sim} q_{ki}^S$ in the ball $B_\epsilon\left(X_{ki}^S\right)$ with the *density function proportional* to $\exp\left\{\lambda[\alpha s(X_{ki}^S, \bullet; \psi) + \ell(\bullet, Y_{ki}^S; \psi)]\right\}$. Furthermore, for each target anchor X_i^T, we sample $\left[\tilde{X}_{ij}^T\right]_{j=1}^{n^T} \overset{\text{iid}}{\sim} q_i^T$ in the ball $B_\epsilon\left(X_i^T\right)$ with the d*ensity function proportional* to $\exp\left\{\lambda \alpha s\left(X_i^T, \bullet; \psi\right)\right\}$.

To sample the particles from their local distributions, we use Stein Variational Gradient Decent (SVGD) [13] with a RBF kernel with kernel width σ. obtained particles $\tilde{X}_{kij}^S$ and $\tilde{X}_{ij}^T$ are then utilized to minimize the objective function in Eq. (4.7) for updating $\psi = (\phi, \theta)$. Specifically, we utilize cross-entropy for the classification loss term ℓ and the symmetric Kullback-Leibler (KL) divergence for the local regular-

ization term $s\left(X, \tilde{X}; \psi\right)$ as $\frac{1}{2}KL\left(f_\psi\left(X\right) \| f_\psi\left(\tilde{X}\right)\right) + \frac{1}{2}KL\left(f_\psi\left(\tilde{X}\right) \| f_\psi\left(X\right)\right)$.

Finally, the global-regularization function of interest $r^g\left(\left[X_{ki}^S\right]_{k,i}, \left[X_i^T\right]_i; \psi\right)$ is defined accordingly depending on the task and explicitly presented in the sequel.

4.0.3 Setting for Domain Adaptation and Semi-supervised Learning

By considering the single source domain as the labeled portion and the target domain as the unlabeled portion, the same setting can be employed for DA and SSL. Particularly, we denote the data/label distribution of the source domain or labeled portion by $\mathbb{P}_1^{S|l}$ and the data distribution of target domain or unlabeled portion by $\mathbb{P}^{T|u}$. Notice that for SSL, $\mathbb{P}^{T|u}$ could be the marginal of $\mathbb{P}^{S|l}$ by marginalizing out the label dimension. Evidently, with this consideration, DA and SSL are special cases of our general framework in Section 4.0.1, where the global-regularization function of interest $r^g\left(\left[X_i^S\right]_i, \left[X_j^T\right]_j; \psi\right)$ is defined as

$$\mathcal{W}_d\left(\frac{1}{B^S}\sum_{i=1}^{B^S}\delta_{U_i^S}, \frac{1}{B^T}\sum_{j=1}^{B^T}\delta_{U_j^T}\right), \tag{4.9}$$

where $U_i^S = \left[g_\phi\left(X_i^S\right), h_\theta\left(g_\phi\left(X_i^S\right)\right)\right]$, $U_j^T = \left[g_\phi\left(X_j^T\right), h_\theta\left(g_\phi\left(X_j^T\right)\right)\right]$, and δ is the Dirac delta distribution. The cost metric d is defined as

$$d\left(U_i^S, U_j^T\right) := \rho_d\left(g_\phi\left(X_i^S\right), g_\phi\left(X_j^T\right)\right) + \gamma\rho_l\left(h_\theta\left(g_\phi\left(X_i^S\right)\right), h_\theta\left(g_\phi\left(X_j^T\right)\right)\right),$$

where ρ_d is a metric on the latent space and $\gamma > 0$.

With the global term in Eq. (4.9), we aim to reduce the discrepancy gap between the *source (labeled)* domain and the *target (unlabeled)* domain for learning domain-invariant representations. It is worth noting that this global term in Eq. (4.9) was inspected in DeepJDOT [32] for DA setting. Our approach is different from that approach in the local regularization term.

4.0.4 Setting for Domain Generalization

By setting $B^T = 0$ (i.e., not use any target data in training), our general framework in Section 4.0.1 is applicable to DG, wherein the global-regularization function of interest $r^g\left(\left[X_{ki}^S\right]_{k,i}, \left[X_i^T\right]_i; \psi\right)$ is

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\frac{1}{K}\mathcal{W}_d\left(\tilde{\mathbb{P}}_{km}, \tilde{\mathbb{P}}_m\right), \tag{4.10}$$

where the cost metric $d = \rho_d$ is a metric on the latent space, $\tilde{\mathbb{P}}_{km}$ is the empirical distribution over $g_\phi\left(X_{ki}^S\right)$ with $Y_{ki}^S = m$, and $\tilde{\mathbb{P}}_m = \frac{1}{K}\sum_{k=1}^{K}\tilde{\mathbb{P}}_{km}$.

4.0.5 Setting for Adversarial Machine Learning

For AML, we have only *single source domain* and need to train a deep model which is robust to adversarial examples. We denote the data/label distribution of the source domain by $\mathbb{P}_1^S$ and propose using a dynamic and pseudo target domain of the *on-the-fly adversarial examples* $\left[\left[X_{1ij}^S\right]_{i=1}^{B^S}\right]_{j=1}^{n^S}$. In addition to the local and loss terms as in equation (4.7), to strengthen model robustness, we propose the following global term to move adversarial examples ($\sim \mathbb{P}^T$) to benign examples ($\sim \mathbb{P}_1^S$):

$$\mathcal{W}_d\left(\frac{1}{B_1^S}\sum_{i=1}^{B_1^S}\delta_{U_i^S}, \frac{1}{B_1^S n^S}\sum_{i=1}^{B_1^S}\sum_{j=1}^{n^S}\delta_{U_{ij}^S}\right), \tag{4.11}$$

where $U_i^S = \left[g_\phi\left(X_{1i}^S\right), h_\theta\left(g_\phi\left(X_{1i}^S\right)\right)\right]$, $U_{ij}^S = \left[g_\phi\left(X_{1ij}^S\right), h_\theta\left(g_\phi\left(X_{1ij}^S\right)\right)\right]$, and the metric d is

$$d\left(U_i^S, U_{\bar{i}j}^S\right) =$$

$$= \mathbb{I}_{Y_{1i}^S = Y_{1\bar{i}}^S}\left[\rho_d\left(g_\phi\left(X_{1i}^S\right), g_\phi\left(X_{1\bar{i}j}^S\right)\right) + \gamma\rho_l\left(h_\theta\left(g_\phi\left(X_{1i}^S\right)\right), h_\theta\left(g_\phi\left(X_{1\bar{i}j}^S\right)\right)\right)\right], \tag{4.12}$$

where $\mathbb{I}$ is the indicator function. Here we note that $X_{1\bar{i}j}^S$ is an adversarial example of $X_{1\bar{i}}^S$ which has the ground-truth label $Y_{1\bar{i}}^S$, hence by using the cost metric as in equation (4.12), we encourage the adversarial example $X_{1\bar{i}j}^S$ to move to a group of the benign examples with the same label.

Finally, to tackle the WS-related terms in Eqs. (4.9,4.10, and 4.11), we employ the entropic regularization dual form of WS, which was demonstrated to have favorable computational complexities [51]–[53].

CHAPTER 5. EXPERIMENTS

To demonstrate the effectiveness of our proposed method, we evaluate its performance on various experiment protocols, including DG, DA, SSL, and AML. Due to the space limitation, the detailed setup regarding the architectures and hyper-parameters are presented in the supplementary material. We tried to use the exact configuration of optimizers and hyper-parameters for all experiments and report the original results in prior work, if possible.

5.0.1 Experiments for DG

Table 5.1: Single domain generalization accuracy (%) on CIFAR-10-C and CIFAR-100-C datasets with different backbone architectures. We use the **bold** font to highlight the best results.

Datasets	Backbone	Standard	Cutout	CutMix	AutoDA	Mixup	AdvTrain	ADA	ME-ADA	GLOT-DR
CIFAR-10-C	AllConvNet	69.2	67.1	68.7	70.8	75.4	71.9	73	78.2	**82.5**
	DenseNet	69.3	67.9	66.5	73.4	75.4	72.4	69.8	76.9	**83.6**
	WideResNet	73.1	73.2	72.9	76.1	77.7	73.8	79.7	83.3	**84.4**
	ResNeXt	72.5	71.1	70.5	75.8	77.4	73	78	83.4	**84.5**
	Average	71	69.8	69.7	74	76.5	72.8	75.1	80.5	**83.7**
CIFAR-100-C	AllConvNet	43.6	43.2	44	44.9	46.6	44	45.3	51.2	**54.8**
	DenseNet	40.7	40.4	40.8	46.1	44.6	44.8	45.2	47.8	**53.2**
	WideResNet	46.7	46.5	47.1	50.4	49.6	44.9	50.4	52.8	**56.5**
	ResNeXt	46.6	45.4	45.9	48.7	48.6	45.6	53.4	57.3	**58.4**
	Average	44.4	43.9	44.5	47.5	47.4	44.8	48.6	52.3	**55.7**

In DG experiments, our setup closely follows [5]. In particular, we validate our method on the CIFAR-C single domain generalization benchmark: train the model on either CIFAR-10 or CIFAR-100 dataset [54], then evaluate it on CIFAR-10-C or CIFAR-100-C [55], correspondingly. In terms of network architectures, we use the exact backbones from [5] to examine the versatility of our method that can be adopted in any type of classifier. GLOT-DR is compared with other state-of-the-art methods in image corruption robustness: Mixup [6], Cutout [7] and Cutmix [8], AutoDA [56], ADA [57], and ME-ADA [5]. Table 5.1 shows the average accuracy when we alternatively train the model on one category and evaluate on the rest. In every setting, GLOT-DR outperforms other methods by large margins. Specifically, our method exceeds the second-best method ME-ADA [5] by 3.2% on CIFAR-10-C and 3.4% on CIFAR-100-C. The substantial gain in terms of the accuracy on various backbone architectures demonstrates the high applicability of the proposed techniques.

Furthermore, we examine multi-source DG where the classifier needs to generalize from multiple source domains to an unseen target domain, using the popular PACS

Table 5.2: Multi-source domain generalization accuracy (%) on PACS datasets. Each column title indicates the target domain used for evaluation, while the rest are for training.

	DSN	L-CNN	MLDG	Fusion	MetaReg	Epi-FCR	AGG	HEX	PAR	ADA	ME-ADA	GLOT-DR
Art	61.1	62.9	66.2	64.1	69.8	64.7	63.4	66.8	66.9	64.3	**67.1**	66.1
Cartoon	66.5	67.0	66.9	66.8	70.4	**72.3**	66.1	69.7	67.1	69.8	69.9	**72.3**
Photo	83.3	89.5	88.0	90.2	91.1	86.1	88.5	87.9	88.6	85.1	88.6	**90.4**
Sketch	58.6	57.5	59.0	60.1	59.2	65.0	56.6	56.3	62.6	60.4	63.0	**65.4**
Average	67.4	69.2	70.0	70.3	72.6	72.0	68.7	70.2	71.3	69.9	72.2	**73.5**

dataset [38]. Our proposed method is applicable in this scenario since it is designed to better learn domain invariant features as well as leverage the diversity from generated data. We compare GLOT-DR against DSN [37], L-CNN [38], MLDG [39], Fusion [41], MetaReg [36], Epi-FCR, AGG [40], HEX [58], and PAR [59]. Table 5.2 shows that our GLOT-DR outperforms the baselines for three cases and averagely surpasses the second-best baseline by 0.9%. The most noticeable improvement is on the Sketch domain ($\approx 2.4\%$), which is the most challenging due to the fact that the styles of the images are colorless and far different from the ones from Art Painting, Cartoon or Photos (i.e. larger domain shift).

5.0.2 Experiments for DA

Table 5.3: Accuracy (%) on Office-31 [60] of ResNet50 model [61] in unsupervised DA methods.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DAN	80.5	97.1	99.6	78.6	63.6	62.8	80.4
RTN	70.2	96.6	95.5	66.3	54.9	53.1	72.8
DANN	84.5	96.8	99.4	77.5	66.2	64.8	81.6
JAN	82	96.9	99.1	79.7	68.2	67.4	82.2
GTA	89.5	97.9	99.8	87.7	72.8	**71.4**	86.5
CDAN	93.1	98.2	**100**	**89.8**	70.1	68	86.6
DeepJDOT	88.9	98.5	99.6	88.2	**72.1**	70.1	86.2
ETD	92.1	**100**	**100**	88	71	67.8	86.2
GLOT-DR	**96.2**	98.9	**100**	90.6	69.9	69.6	**87.8**

In this section, we conduct experiments on the commonly used dataset for real-world unsupervised DA - Office-31 [60], comprising images from three domains: Amazon (A), Webcam (W) and DSLR (D). Our proposed GLOT-DR is compared against baselines: ResNet-50 [61], DAN [62], RTN [63], DANN [33], JAN [64], GTA [65], CDAN [35], DeepJDOT [32] and ETD [34]. For a fair comparison, we follow the training setups of CDAN and compare with other works using this configuration. As can be seen from Table 5.3, GLOT-DR achieves the best overall performance among baselines with 87.8% accuracy. Compared with ETD, which is another OT-

based domain adaptation method, our performance significantly increase by 4.1% on A$\rightarrow$W task, 2.1% on W$\rightarrow$A and 1.6% on average.

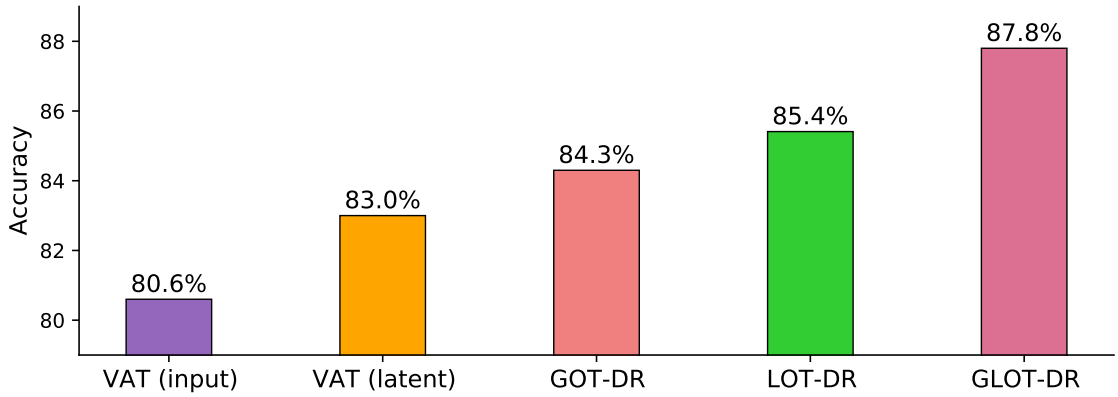

Figure 5.1: Average accuracy of ResNet50 [61] on Office-31: Comparision between GLOT-DR's variants and VAT [3] on the input and latent spaces.

We further extensively investigate the role of different components in GLOT-DR. Specifically, the elimination of the global-regularization term in equation (4.9) downgrades our method to Local Optimal Transport based Distributional Robustness (LOT-DR). Similarly, when discarding the local distribution robustness term, the attained method is denoted by GOT-DR. We then compare these 2 variants of GLOT-DR to the well-known adversarial machine learning method VAT [3]. To be more specific, in the adversarial samples generation, we apply VAT by perturbing on the: (i) input space, (ii) latent space. Figure 5.1 shows that the employment of VAT on latent space (orange) is more effective than on the input space (purple), 83% and 80.6%. However, using GOT-DR or LOT-DR is even more effective: performance is boosted to 84.3% and 85.4%, respectively. Lastly, using the full method GLOT-DR yields the highest average accuracy score among all.

5.0.3 Experiments for SSL

Sharing a similar objective with DA, which utilizes the unlabeled samples for improving the model performance, SSL methods can also benefit from our proposed technique. We present the empirical results on CIFAR-10 benchmark with ConvLarge architecture, following VAT's protocol [3], which serves as a strong baseline in this experiment. We refer readers to the supplementary material for more details on the architecture of ConvLarge. Results in Figure 5.2 (when training with 1,000 and 4,000 labeled examples) demonstrate that, with only $n^S = n^T = 1$ perturbed sample per anchor, the performance of LOT-DR slightly outperforms VAT with $\sim 0.5\%$. With more perturbed samples per anchor, this gap increases: approximately 1% when $n^S = n^T = 2$ and 1.5% when $n^S = n^T = 4$. Similar to the previous DA experiment, adding the global regularization term helps increase accuracy by $\sim 1\%$

in this setup.

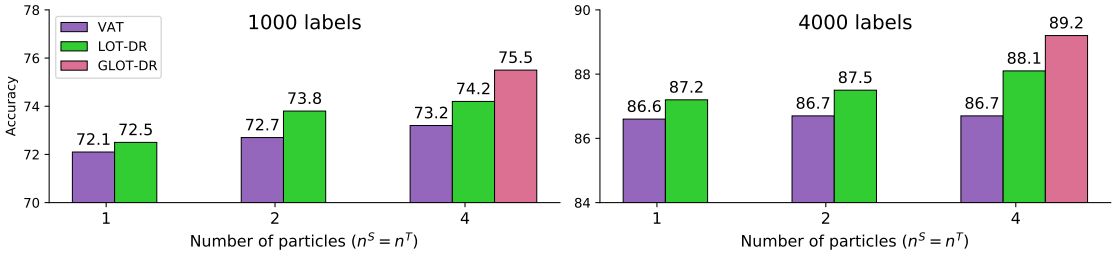

Figure 5.2: Accuracy (%) on CIFAR-10 of ConvLarge model in SSL settings when using 1,000 and 4,000 labeled examples (i.e. 100 and 400 labeled samples each class). Best viewed in color.

5.0.4 Experiments for AML

Table 5.4 shows the evaluation against adversarial examples. We compare our method with PGD-AT [25] and TRADES [12], two well-known defense methods in AML. For the sake of fair comparison, we use the same adversarial training setting for all methods, which is carefully investigated in [66]. We also compare with adversarial distributional training methods [4] (ADT-EXP and ADT-EXPAM), which assume that the adversarial distribution explicitly follows normal distribution. It can be seen from Table 5.4 that our GLOT-DR method outperforms all these baselines in both natural and robustness performance. Specifically, compared to PGD-AT, our method has an improvement of 0.8% in natural accuracy and around 1% robust accuracies against PGD200 and AA attacks. Compared to TRADES, while achieving the same level of robustness, our method has a better performance with benign examples with a gap of 2.5%. Especially, our method significantly outperforms ADT by around 7% under the PGD200 attack.

Table 5.4: Adversarial robustness evaluation on CIFAR10 of ResNet18 model. PGD, AA and B&B represent the robust accuracy against the PGD attack (with 10/200 iterations) [25], Auto-Attack [26] and B&B attack [67], respectively, while NAT denotes the natural accuracy.

Method	NAT	PGD10	PGD200	AA	B&B
PGD-AT*	82.52	53.58	-	48.51	-
TRADES*	81.45	53.51	-	49.06	-
PGD-AT$^\diamond$	83.36	53.52	52.21	49.00	48.50
TRADES$^\diamond$	81.64	53.73	53.11	49.77	49.02
ADT-EXP	83.02	-	45.80	45.80	46.50
ADT-EXPAM	84.11	-	46.10	44.50	45.83
GLOT-DR	**84.13**	**54.13**	**53.1**8	**49.94**	**49.40**

* Results are taken from Pang et al. [66]

$^\diamond$ Our reproduced results.

CHAPTER 6. CONCLUSION

Although DR is a promising framework to improve neural network robustness and generalization capability, its current formulation shows some limitations, circumventing its application to real-world problems. First, its formulation is not sufficiently rich to express a global regularization effect targeting many applications. Second, the dual form is not readily trainable to incorporate into deep learning models. In this work, we propose a rich OT based DR framework, named ***G**lobal-**L**ocal **O**ptimal **T**ransport based **D**istributional **R**obustness* (GLOT-DR) which is sufficiently rich for many real-world applications including DG, DA, SSL, and AML and has a closed-form solution. We conduct comprehensive experiments to compare our GLOT-DR with state-of-the-art baselines accordingly. Empirical results have demonstrated the merits of our GLOT-DR.

Publications: Parts of this thesis are the extended/modified versions of the ones from the following papers:

- **Hoang Phan**, Ngoc Tran, Trung Le, Toan Tran, Nhat Ho, Dinh Phung. "Stochastic multiple target sampling gradient descent". Under review. 2022
- **Hoang Phan**, Trung Le, Trung Phung, Anh Bui, Nhat Ho, Dinh Phung. "Global-Local Regularization Via Distributional Robustness". Under review, 2022.

Implementations: Our code employs Pytorch framework is available on Github: `https://github.com/VietHoang1512/thesis`.

Supplement to "Global-Local Regularization Via Distributional Robustness"

These appendices provide supplementary details and results of GLOT, including our theory development and additional experiments. This consists of the following sections:

- Appendix 7.1 contains the proofs of our theory development.
- Appendix 7.2 contains the network architectures, experiment settings of our experiments and additional ablation studies.

7.1 Proofs of Our Theory Development

We here provide derivations for the Taylor expansion formulation in SVGD formulations in 3.4 and the proof for the equivalence in optimizing two equations (4.3) and (4.4) in Section 7.1.2. Then, we detail how to derive the optimization formulations (2) and (4.7) for solving the problems discussed in Section 4.0.1.

7.1.1 Derivations for the Taylor expansion formulation

We have

$$\nabla_\epsilon KL\left(q^{[T]}\|p\right)\Big|_{\epsilon=0} = -\langle\phi,\psi\rangle_{\mathcal{H}_k^d}. \tag{7.1}$$

Proof of Equation (7.1): Since T is assumed to be an invertible mapping, we have the following equations:

$$KL\left(q^{[T]}\|p\right) = KL\left(T\#q\|p\right) = KL(q||T^{-1}\#p)$$

and

$$KL(q||T^{-1}\#p) = KL(q||T^{-1}\#p)\big|_{\epsilon=0} + \epsilon\nabla_\epsilon KL(q||T^{-1}\#p)\big|_{\epsilon=0} + O(\epsilon^2). \tag{7.2}$$

According to the change of variables formula, we have $T^{-1}\#p(\theta) = p(T(\theta))|\det\nabla_\theta T(\theta)|$, then:

$$KL(q||T^{-1}\#p) = \mathbb{E}_{\theta\sim q}[\log q(\theta) - \log p(T(\theta)) - \log|\det\nabla_\theta T(\theta)|].$$

Using this, the first term in Equation (7.2) is rewritten as:

$$\begin{aligned} KL(q||p) =& KL(T\#q||p)\big|_{\epsilon=0} = KL(q||T^{-1}\#p)\big|_{\epsilon=0} = \\ &= \mathbb{E}_{\theta\sim q}[\log q(\theta) - \log p(\theta) - \log|\det \nabla_\theta \theta|] \\ &= \mathbb{E}_{\theta\sim q}[\log q(\theta) - \log p(\theta)]. \end{aligned} \tag{7.3}$$

Similarly, the second term in Equation (7.2) could be expressed as:

$$\nabla_\epsilon KL(q||T^{-1}\#p_k)\big|_{\epsilon=0} \tag{7.4}$$

$$\begin{aligned} &= \mathbb{E}_{\theta\sim q}[\nabla_\epsilon \log q(\theta) - \nabla_\epsilon \log p(T(\theta)) - \nabla_\epsilon \log|\det \nabla_\theta T(\theta)|\Big]\Big|_{\epsilon=0} \\ &= -\mathbb{E}_{\theta\sim q}[\nabla_\epsilon \log p(T(\theta)) + \nabla_\epsilon \log|\det \nabla_\theta T(\theta)|\Big]\Big|_{\epsilon=0} \\ &= -\mathbb{E}_{\theta\sim q}[\nabla_T \log p(T(\theta))\nabla_\epsilon T(\theta)\Big]\Big|_{\epsilon=0} \\ &\qquad - \mathbb{E}_{\theta\sim q}\Big[\frac{1}{|\det \nabla_\theta T(\theta)|}\frac{|\det \nabla_\theta T(\theta)|}{\det \nabla_\theta T(\theta)}\nabla_\epsilon \det \nabla_\theta T(\theta)\Big]\Big|_{\epsilon=0} \end{aligned} \tag{7.5}$$

$$\begin{aligned} &= -\mathbb{E}_{\theta\sim q}[\nabla_T \log p(T(\theta))\phi(\theta)]\big|_{\epsilon=0} \\ &\qquad - \mathbb{E}_{\theta\sim q}p\Big[\frac{\det \nabla_\theta T(\theta)\,\mathrm{tr}((\nabla_\theta T(\theta)^{-1}\nabla_\epsilon \nabla_\theta T(\theta))}{\det \nabla_\theta T(\theta\theta)}\Big]\Big|_{\epsilon=0} \\ &= -\mathbb{E}_{\theta\sim q}[\nabla_\theta \log p(\theta)\phi(\theta) + \mathrm{tr}(\nabla_\theta \phi(\theta))]. \end{aligned}$$

It could be shown from the reproducing property of the RKHS that $\phi(\theta) = \langle \phi(\cdot), k(\theta,\cdot)\rangle_{\mathcal{H}_k}$, then we find that

$$\frac{\partial \phi(\theta)}{\partial \hat{\theta}} = \left\langle \phi(\cdot), \frac{\partial k(\theta,\cdot)}{\partial \hat{\theta}} \right\rangle_{\mathcal{H}_k}. \tag{7.6}$$

Let $U_{d\times d} = \nabla_\theta \phi(\theta)$ whose u^T denotes the i^{th} row vector and the particle $\theta \in \mathbb{R}^d$ is represented by $\{\hat{\theta}\}_{i=1}^d$, the row vector u^T is given by:

$$u^T \frac{\partial \phi(\theta)}{\partial \theta} = \frac{\partial \phi(\theta)}{\partial(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_d)} = \Big[\frac{\partial \phi(\theta)}{\partial \hat{\theta}_1}; \frac{\partial \phi(\theta)}{\partial \hat{\theta}_2}; \ldots; \frac{\partial \phi(\theta)}{\partial \hat{\theta}_d}\Big]. \tag{7.7}$$

Combining Property (7.6) and Equation (7.7), we have:

$$\begin{aligned} &u^T\Big[\frac{\partial \phi(\theta)}{\partial \hat{\theta}_1}; \frac{\partial \phi(\theta)}{\partial \hat{\theta}_2}; \ldots; \frac{\partial \phi(\theta)}{\partial \hat{\theta}_d}\Big] \\ &\quad = \Big[\left\langle \phi(\cdot), \frac{\partial k(\theta,\cdot)}{\partial \hat{\theta}_1} \right\rangle_{\mathcal{H}_k}; \left\langle \phi(\cdot), \frac{\partial k(\theta,\cdot)}{\partial \hat{\theta}_2} \right\rangle_{\mathcal{H}_k}; \ldots; \left\langle \phi(\cdot), \frac{\partial k(\theta,\cdot)}{\partial \hat{\theta}_d} \right\rangle_{\mathcal{H}_k}\Big]. \end{aligned} \tag{7.8}$$

Substituting Equation (7.8) to Equation (7.5), the linear term of the Taylor expansion could be derived as:

$$
\begin{aligned}
&\nabla_\epsilon KL(q||T^{-1}\#p)\big|_{\epsilon=0} \\
&= -\mathbb{E}_{\theta\sim q}[\nabla_\theta \log p(\theta)\phi(\theta) + \mathrm{tr}(\nabla_\theta \phi(\theta))] \\
&= -\mathbb{E}_{\theta\sim q}\Big[\sum_{j=1}^{d} \langle \phi_j(\cdot), k(\theta,\cdot)\rangle_{\mathcal{H}_k} (\nabla_\theta \log p(\theta))_j + \frac{\partial \phi_j(\theta)}{\partial \hat{\theta}_j})\Big] \\
&= -\sum_{j=1}^{d} \mathbb{E}_{\theta\sim q}\Big[\, \langle \phi_j(\cdot), k(\theta,\cdot)(\nabla_\theta \log p(\theta))_j\rangle_{\mathcal{H}_k} + \Big\langle \phi_j(\cdot), \Big(\frac{\partial k(\theta,\cdot)}{\partial \theta}\Big)_j \Big\rangle_{\mathcal{H}_k}\Big] \\
&= -\sum_{j=1}^{d} \Big\langle \phi_j(\cdot), \mathbb{E}_{\theta\sim q}\Big[k(\theta,\cdot)(\nabla_\theta \log p(\theta))_j + \Big(\frac{\partial k(\theta,\cdot)}{\partial \theta}\Big)_j\Big]\Big\rangle_{\mathcal{H}_k} \\
&= -\left\langle \phi(\cdot), \psi(\cdot)\right\rangle_{\mathcal{H}_k^d},
\end{aligned}
$$

where $(v)_j$ denotes the j-th element of v and $\psi(\cdot) \in \mathcal{H}_k^d$ is a matrix whose j^{th} column vector is given by

$$\mathbb{E}_{\theta\sim q}\Big[k(\theta,\cdot)(\nabla_\theta \log p(\theta))_j + \Big(\frac{\partial k(\theta,\cdot)}{\partial \theta}\Big)_j\Big].$$

In other word, the formula of $\psi(\cdot)$ becomes

$$\mathbb{E}_{\theta\sim q}\Big[k(\theta,\cdot)\nabla_\theta \log p(\theta) + \frac{\partial k(\theta,\cdot)}{\partial \theta}\Big].$$

As a consequence, we obtain the conclusion of Equation (7.1).

7.1.2 Proof of Lemma 1

Let

$$\gamma^* = \underset{\gamma:\in\Gamma_\epsilon}{\mathrm{argmax}} \mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[r\left(\tilde{Z};\phi,\theta\right)\right]$$

be the optimal solution of the inner max in equation (4.4). Denote $\tilde{\mathbb{P}}^*$ as the distribution obtained from γ^* by maginalizing the dimensions of Z. We prove that $\tilde{\mathbb{P}}^*$ is the optimal solution of the inner max in equation (4.3). Let $\tilde{\mathbb{P}}$ be a feasible solution of the inner max in equation (4.3), meaning that $\mathcal{W}_\rho\left(\mathbb{P},\tilde{\mathbb{P}}\right) \leq \epsilon$. Therefore, there exists $\gamma \in \Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right)$ such that $\mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} \leq \epsilon$ or $\gamma \in \Gamma_\epsilon$. We have

$$
\begin{aligned}
\mathbb{E}_{\tilde{\mathbb{P}}}\left[r\left(\tilde{Z};\phi,\theta\right)\right] &= \mathbb{E}_{\gamma}\left[r\left(\tilde{Z};\phi,\theta\right)\right] \\
&\leq \mathbb{E}_{\gamma^*}\left[r\left(\tilde{Z};\phi,\theta\right)\right] = \mathbb{E}_{\mathbb{P}^*}\left[r\left(\tilde{Z};\phi,\theta\right)\right].
\end{aligned}
$$

We reach the conclusion that $\tilde{\mathbb{P}}^*$ is the optimal solution of the inner max in equation (4.3). That concludes our proof.

7.1.3 Proof of Theorem 2

Given $\gamma \in \Gamma_\epsilon$, we first prove that if $\mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]$ is finite $\forall q > 1$ then

$$\begin{aligned} M_\gamma &:= \lim_{q\to\infty} \mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} \\ &= \sup_{(Z,\tilde{Z})\in \text{supp}(\gamma)} \max\left\{\max_{k,i,j}\left\|X^S_{kij} - \tilde{X}^S_{kij}\right\|_p, \max_{i,j}\left\|X^T_{ij} - \tilde{X}^T_{ij}\right\|_p\right\}. \end{aligned}$$

Let denote A_γ as the set of $\left(Z,\tilde{Z}\right) \in \text{supp}\left(\gamma\right)$ such that

$$\max\left\{\max_{k,i,j}\left\|X^S_{kij} - \tilde{X}^S_{kij}\right\|_p, \max_{i,j}\left\|X^T_{ij} - \tilde{X}^T_{ij}\right\|_p\right\} = M_\gamma.$$

We have

$$\mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} = \left[\int_{A_\gamma} \rho\left(Z,\tilde{Z}\right) d\gamma\left(Z,\tilde{Z}\right) + \int_{A^c_\gamma} \rho\left(Z,\tilde{Z}\right) d\gamma\left(Z,\tilde{Z}\right)\right]^{1/q}.$$

It is obvious that if $\left(Z,\tilde{Z}\right) \sim \gamma$ then

$$\rho\left(Z,\tilde{Z}\right) := \sum_{i=1}^{B^T}\sum_{j=1}^{n^T}\left\|X^T_{ij} - \tilde{X}^T_{ij}\right\|^q_p + \sum_{k=1}^{K}\sum_{i=1}^{B^S_k}\sum_{j=1}^{n^S}\left\|X^S_{kij} - \tilde{X}^S_{kij}\right\|^q_p.$$

Therefore, for $\left(Z,\tilde{Z}\right) \in A^c_\gamma$, we have

$$\lim_{q\to\infty}\frac{\rho\left(Z,\tilde{Z}\right)}{M^q_\gamma} = 0,$$

while for $\left(Z,\tilde{Z}\right) \in A_\gamma$, we have

$$\lim_{q\to\infty}\frac{\rho\left(Z,\tilde{Z}\right)}{M^q_\gamma} = 1.$$

We derive as

$$\begin{aligned}
&\lim_{q\to\infty} \mathbb{E}_{(Z,\tilde{Z})\sim\gamma} \left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} \\
&= M_\gamma \lim_{q\to\infty} \left[\int_{A_\gamma} \frac{\rho\left(Z,\tilde{Z}\right)}{M^q} d\gamma\left(Z,\tilde{Z}\right) + \int_{A_\gamma^c} \frac{\rho\left(Z,\tilde{Z}\right)}{M^q} d\gamma\left(Z,\tilde{Z}\right)\right]^{1/q} \\
&= M_\gamma \lim_{q\to\infty} \gamma\left(A_\gamma\right)^{1/q} = M_\gamma.
\end{aligned}$$

Therefore, $\gamma \in \Gamma_\epsilon$ with $q = \infty$ is equivalent to the fact that the support set $\text{supp}\left(\gamma\right)$ is the union of B_Z with $Z \in \text{supp}\left(\mathbb{P}\right)$, where B_Z is defined as follows:

$$\begin{aligned}
B_Z &:= \prod_{k=1}^{K} \prod_{i=1}^{B_k^S} \prod_{j=0}^{n_k^S} B_\epsilon\left(X_{kij}^S\right) \prod_{i=1}^{B^T} \prod_{j=1}^{n^T} B_\epsilon\left(X_{ij}^T\right) \\
&= \prod_{k=1}^{K} \prod_{i=1}^{B_k^S} \prod_{j=0}^{n_k^S} B_\epsilon\left(X_{ki}^S\right) \prod_{i=1}^{B^T} \prod_{j=1}^{n^T} B_\epsilon\left(X^T\right).
\end{aligned}$$

We can equivalently turn the optimization problem in equation (4.5) as follows:

$$\begin{aligned}
&\max_{\gamma\in\Gamma} \mathbb{E}_{(Z,\tilde{Z})\sim\gamma}\left[r\left(\tilde{Z};\phi,\theta\right)\right] + \frac{1}{\lambda}\mathbb{H}\left(\gamma\right) \\
&\text{s.t. : } \text{supp}\left(\gamma\right) = \cup_{Z\in\,\text{supp}(\mathbb{P})} B_Z,
\end{aligned} \tag{7.9}$$

where $\Gamma = \cup_{\tilde{\mathbb{P}}} \Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right)$.

Because $\gamma \in \Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right)$ for some $\tilde{\mathbb{P}}$, we can parameterize its density function as:

$$\gamma\left(Z,\tilde{Z}\right) = p\left(Z\right)\tilde{p}\left(\tilde{Z} \mid Z\right),$$

where $p\left(Z\right)$ is the density function of $\mathbb{P}$ and $\tilde{p}\left(\tilde{Z} \mid Z\right)$ has the support set B_Z. Please note that the constraint for $\tilde{p}\left(\tilde{Z} \mid Z\right)$ is $\int_{B_Z} \tilde{p}\left(\tilde{Z} \mid Z\right) d\tilde{Z} = 1$.

The Lagrange function for the optimization problem in equation (7.9) is as follows:

$$\begin{aligned}
\mathcal{L} = &\int r\left(\tilde{Z};\phi,\theta\right) p\left(Z\right)\tilde{p}\left(\tilde{Z}|Z\right) dZd\tilde{Z} \\
&- \frac{1}{\lambda}\int p\left(Z\right)\tilde{p}\left(\tilde{Z}|Z\right)\log\left[p\left(Z\right)\tilde{p}\left(\tilde{Z}|Z\right)\right] dZd\tilde{Z} \\
&+ \int \alpha\left(Z\right)\left[\tilde{p}\left(\tilde{Z} \mid Z\right) d\tilde{Z} - 1\right] d\tilde{Z}dZ,
\end{aligned}$$

where the integral w.r.t Z over on $\text{supp}\left(\mathbb{P}\right)$ and the one w.r.t. $\tilde{Z}$ over B_Z.

Taking the derivative of $\mathcal{L}$ w.r.t. $\tilde{p}\left(\tilde{Z} \mid Z\right)$ and setting it to 0, we obtain

$$0 = r\left(\tilde{Z};\phi,\theta\right)p\left(Z\right) + \alpha\left(Z\right) - \frac{p\left(Z\right)}{\lambda}\left[\log p\left(Z\right) + \log\tilde{p}\left(\tilde{Z}|Z\right) + 1\right],$$

$$\tilde{p}\left(\tilde{Z}|Z\right) = \frac{\exp\left\{\lambda\left[r\left(\tilde{Z};\phi,\theta\right) + \frac{\alpha(Z)}{p(Z)}\right] - 1\right\}}{p\left(Z\right)}.$$

Taking into account $\int_{B_Z}\tilde{p}\left(\tilde{Z} \mid Z\right)d\tilde{Z} = 1$, we achieve

$$\int_{B_Z}\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}d\tilde{Z} = \frac{p\left(Z\right)}{\exp\left\{\lambda\frac{\alpha(Z)}{p(Z)} - 1\right\}}.$$

Therefore, we arrive at

$$\tilde{p}^{*}\left(\tilde{Z}|Z\right) = \frac{\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}}{\int_{B_Z}\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}d\tilde{Z}},$$

$$\gamma^{*}\left(Z,\tilde{Z}\right) = p\left(Z\right)\frac{\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}}{\int_{B_Z}\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}d\tilde{Z}}. \tag{7.10}$$

Finally, by noting that

$$p\left(Z\right) = \prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S}p_k^S\left(X_{ki}^S, Y_{ki}^S\right)\prod_{i=1}^{B^T}\prod_{j=0}^{n^T}p^T\left(X^T\right)\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}$$

$$= \exp\left\{\lambda\beta r^g\left(\tilde{Z};\psi\right)\right\}\prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S}\exp\left\{\lambda[\alpha s(X_{ki}^S, \tilde{X}_{kij}^S;\psi) + \ell(\tilde{X}_{kij}^S, Y_{ki}^S;\psi)]\right\}$$

$$\prod_{i=1}^{B^T}\prod_{j=1}^{n^T}\exp\left\{\lambda\alpha s\left(X^T, \tilde{X}_{ij}^T;\psi\right)\right\}.$$

And

$$\int_{B_Z} \exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\} d\tilde{Z} = \exp\left\{\lambda\beta r^g\left(\tilde{Z};\psi\right)\right\}$$

$$\prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S}\int_{B_\epsilon(X_{ki}^S)} \exp\left\{\lambda[\alpha s(X_{ki}^S,\tilde{X}_{kij}^S;\psi)+\ell(\tilde{X}_{kij}^S,Y_{ki}^S;\psi)]\right\} d\tilde{X}_{kij}^S$$

$$\prod_{i=1}^{B^T}\prod_{j=1}^{n^T}\int_{B_\epsilon(X^T)} \exp\left\{\lambda\alpha s\left(X^T,\tilde{X}_{ij}^T;\psi\right)\right\} d\tilde{X}_{ij}^T,$$

we reach the conclusion.

7.1.4 Proof of the optimization problem in equation (4.7)

By substituting $\gamma^*\left(Z,\tilde{Z}\right)$ in equation (7.10) back to the objective function in (4.4), we obtain

$$\min_{\psi}\min_{\theta,\phi}\max_{\gamma:\in\Gamma_\epsilon}\mathbb{E}_{(Z,\tilde{Z})\sim\gamma^*}\left[r\left(\tilde{Z};\phi,\theta\right)\right].$$

By referring to the construction of Z and $\tilde{Z}$ and noting that for $\left(Z,\tilde{Z}\right)\sim\gamma^*$

$$\begin{aligned} r^l\left(\tilde{Z};\psi\right) &:= \sum_{i=1}^{B^T}\sum_{j=1}^{n^T} s\left(\tilde{X}_{i0}^T,\tilde{X}_{ij}^T;\psi\right)+\sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=1}^{n_k^S} s\left(\tilde{X}_{ki0}^S,\tilde{X}_{kij}^S;\psi\right)\\ &= \sum_{i=1}^{B^T}\sum_{j=1}^{n^T} s\left(X^T,\tilde{X}_{ij}^T;\psi\right)+\sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=1}^{n_k^S} s\left(X_{ki}^S,\tilde{X}_{kij}^S;\psi\right),\\ \mathcal{L}\left(\tilde{Z};\psi\right) &:= \sum_{k=1}^{K}\sum_{i=1}^{B_k^S}\sum_{j=0}^{n_k^S}\ell\left(\tilde{X}_{kij}^S,Y_{ki}^S;\psi\right). \end{aligned}$$

As a consequence, we gain the final optimization problem.

7.2 Implementation Details

In this section, we provide the detailed implementation for all of our experiments along with some additional experimental results. We begin with presenting the pseudo code used to sample from local distributions of our method.

Algorithm 1 Projected SVGD.

Input: A local distribution around X with an unnormalized density function $\tilde{p}(\cdot)$ and a set of initial particles $\{X^0\}_{i=1}^n$.
Output: A set of particles $\{X\}_{i=1}^n$ that approximates the local distribution corresponding to $\tilde{p}(\cdot)$.
for $l = 1$ to L **do**
$X^{l+1} = \prod_{B_\epsilon(X)} \left[X^l + \eta_l \hat{\phi}^*(X^l)\right]$
where $\hat{\phi}^*(X) = \frac{1}{n}\sum_{j=1}^n [k(X_j^l, X)\nabla_{X_j^l} \log \tilde{p}(X_j^l) + \nabla_{X_j^l} k(X_j^l, X)]$ and η_l is the step size at the l^{th} iteration.
end for

7.2.1 Entropic Regularized Duality for WS

To enable the application of optimal transport in machine learning and deep learning, Genevay et al. developed an entropic regularized dual form in [68]. First, they proposed to add an entropic regularization term to the primal form:

$$\mathcal{W}_d^\epsilon(\mathbb{P}, \mathbb{Q}) := \min_{\gamma \in \Gamma(\mathbb{Q}, \mathbb{P})} \left\{ \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \left[d(\mathbf{x}, \mathbf{y}) \right] + \epsilon KL(\gamma \| \mathbb{P} \otimes \mathbb{Q}) \right\}$$

where ϵ is the regularization rate, $KL(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence, and $\mathbb{P} \otimes \mathbb{Q}$ represents the specific coupling in which $\mathbb{Q}$ and $\mathbb{P}$ are independent. Note that when $\epsilon \to 0$, $\mathcal{W}_d^\epsilon(\mathbb{P}, \mathbb{Q})$ approaches $\mathcal{W}_d(\mathbb{P}, \mathbb{Q})$ and the optimal transport plan γ_ϵ^* of equation (7.11) also weakly converges to the optimal transport plan γ^* of the primal form. In practice, we set ϵ to be a small positive number, hence γ_ϵ^* is very close to γ^*. Second, using the Fenchel-Rockafellar theorem, they obtained the following dual form w.r.t. the potential ϕ

$$\begin{aligned} \mathcal{W}_d^\epsilon(\mathbb{P}, \mathbb{Q}) &= \max_\phi \left\{ \int \phi_\epsilon^c(\mathbf{x}) \, \mathrm{d}\mathbb{Q}(\mathbf{x}) + \int \phi(\mathbf{y}) \, \mathrm{d}\mathbb{P}(\mathbf{y}) \right\} \\ &= \max_\phi \left\{ \mathbb{E}_\mathbb{Q}\left[\phi_\epsilon^c(\mathbf{x})\right] + \mathbb{E}_\mathbb{P}\left[\phi(\mathbf{y})\right] \right\}, \end{aligned} \tag{7.11}$$

where $\phi_\epsilon^c(\mathbf{x}) := -\epsilon \log\left(\mathbb{E}_\mathbb{P}\left[\exp\left\{\frac{-d(\mathbf{x},\mathbf{y})+\phi(\mathbf{y})}{\epsilon}\right\}\right]\right)$.

In order to calculate the global WS-related regularization terms in equations (4.9),

(4.10), and (4.11), we apply the above entropic regularized dual form. The Kantorovich potential network ϕ is a simple network with two fully connected layers with ReLU activation in the middle: $\mathrm{FC}_{\mathrm{latent\_dim}\times 512} \rightarrow \mathrm{ReLU} \rightarrow \mathrm{FC}_{512\times 1}$ is used throughout experiments. Note that the $\mathrm{latent\_dim}$ depends on the main network.

Additionally, the distance ρ_d in equation (4.10) used in all experiments is the Euclidean distance $\mathrm{d}(\mathrm{x}_1, \mathrm{x}_2) = ||\mathrm{x}_1 - \mathrm{x}_2||_2^2$, the prediction discrepancy trade-off γ is set equal to 0.5, and the entropic regularization parameter λ in equation (4.5) is 0.1.

7.2.2 Projected SVGD Setting

For Projected SVGD in Algorithm 1, we employ an RBF kernel

$$k\left(X, \tilde{X}\right) = \exp\left\{\frac{-\left\|X - \tilde{X}\right\|_2^2}{2\sigma^2}\right\},$$

where the kernel width is set according to the main text [13].

7.2.3 Experiments for DG

a, Network Architecture and Hyperparamters

As mentioned in the main text, we incorporate well-studied backbones for our experiments, following the implementation of for single domain generalization tasks in [5]. In particular:

- LeNet5 [69] is employed in the MNIST experiment. We first pre-train the network on the MNIST dataset without applying any DG method for 100 iterations, then on each iteration 100, 200, 300 we generate particles with $n^S = n^T = n \in \{1, 2, 4\}$ by running the Projected SVGD sampling1 in $L = 15$ iterations, step size $\eta = 0.002$. We use Adam optimizer [70] with learning rate 10^{-5} and train for 15000 iteration in total with batch size of 32.
- CIFAR-C [1] experiment uses 4 different backbone architectures, namely: All Convolutional Network (AllConvNet) [71], DenseNet [72] , WideResNet [73], and ResNeXt [74]. We set $n^S = n^T = n = 2$ particles, $L = 15$ iterations, step size $\eta = 0.001$ and minimize the loss with SGD optimizer with initial learning rate of 0.1 and batch size 128. Similar to MNIST experiment, we first pretrain the network for 10 epochs then generate augmented images on epoch 10 and 20, number of total epochs required for training are 150 in the case of AllConvNet and WideResNet, 250 epochs for DenseNet and ResNeXt.
- We used an AlexNet [75] pretrained on ImageNet [76] in the PACS experiment.

[1] Note that in both CIFAR-C and MNIST experiments, we are provided with only a single source domain, thus GLOT-DR downgrades exactly to LOT-DR.

Table 7.1: Details on the domain generalization benchmark datasets

Dataset	# classes	Shape
MNIST [77]	10	32×32
SVHN [78]	10	32×32
MNIST-M [79]	10	32×32
SYN [79]	10	32×32
USPS [80]	10	32×32
CIFAR-10-C [55]	15	$3 \times 32 \times 32$
CIFAR-100-C [55]	15	$3 \times 32 \times 32$
PACS [38]	7	$3 \times 224 \times 224$

Different from the two above experiments, which generate augmented images and append them directly to the training set, we generate the augmented images in each mini-batch and calculate the local/global regularization terms. $n^S = n^T$ are set qual to 2, $L = 15$ iterations, step size $\eta = 0.007$. The initial global and local trade-off are 3.10^{-5} and 50, these parameters are is adjusted by $\frac{\text{iter}}{\#\text{num\_iter}}$ in iter-th iteration. We train AlexNet for 45.000 iterations with SGD optimizer and 10^{-3} learning rate.

b, Datasets and Baselines

We present the details on each dataset used in domain generalization experiments in Table. 7.1. Digits datasets: MNIST [77], SVHN [78], MNIST-M [79], SYN [79], USPS [80] - each contains 10 classes from $0-9$, which are resized to 32×32 images in our experiment. CIFAR-10-C [55], and CIFAR-100-C [55] consist of corrupted images from the original CIFAR [54] datasets with 15 corruptions types applied. In terms of multi-source domain generalization, we test our proposed model on PACS dataset [38], which includes $3 \times 224 \times 224$ images from four different datasets, including Photo, Art painting, Cartoon, and Sketch.

In the digits experiment, 10000 images are sellected from MNIST dataset as the training set for the source domain and the other four data sets as the target domains: SVHN , MNIST-M, SYN , USPS. We compare our method with the following baselines: (i) Empirical Risk Minimization (ERM), (ii) PAR [59], (iii) ADA [57] and (iv) ME-ADA [5]. For a fair comparison, we did not use any data augmentation in this digits experiment, all the samples are considered as RGB images (we duplicate the channels if they are grayscale images).

c, Experimental Results

Table. 7.2 shows that our model achieves the highest average accuracy compared to the other baselines for all values of $n^S = n^T = n \in \{1, 2, 4\}$, with the highest

Table 7.2: Average classification accuracy (%) on MNIST benchmark, we first train the LeNet5 [69] architecture on MNIST then evaluate on SVHN, MNIST-M, SYN, USPS. We repeat experiment for 10 times and report the mean value and standard deviation.

	SVHN	MNIST-M	SYN	USPS	Average
Standard (ERM)	31.95± 1.91	55.96± 1.39	43.85± 1.27	79.92± 0.98	52.92± 0.98
PAR	36.08± 1.27	61.16± 0.21	45.48± 0.35	79.95± 1.18	55.67 ± 0.33
ADA	35.70 ± 2.00	58.65± 1.72	47.18± 0.61	80.40± 1.70	55.48± 0.74
ME-ADA	42.00± 1.74	63.98± 1.82	49.80± 1.74	79.10± 1.03	58.72± 1.12
GLOT-DR n=1	42.70 ± 1.03	67.72 ± 0.63	**50.53 ± 0.88**	82.32 ± 0.63	60.82 ± 0.79
GLOT-DR n=2	42.35 ± 1.44	67.95 ± 0.56	**50.53 ± 0.99**	82.33 ± 0.61	60.81± 0.90
GLOT-DR n=4	**43.10 ± 1.16**	**68.44 ± 0.46**	50.49 ± 1.04	**82.48 ± 0.51**	**61.13 ± 0.79**

overall score when $n = 4$. In particular, we observe the highest improvement in MNIST-M target domain of $\approx 5\%$, and $\approx 2.5\%$ overall. Our GLOT-DR also exhibits more consistent with smaller variation in terms of accuracy between runs compared to the second-best method, $(0.79\% - 1.12\%)$.

7.2.4 Experiments for DA

a, Network architectures and Hyperparameters

The ResNet50 [61] architecture pretrained on ImageNet, followed by a two fully connected layers classifier. is the same as that of the previous work. We evaluate GLOT-DR on the standard object image classification benchmarks in domain adaptation: Office-31 and ImageCLEF-DA. The proposed method is employed on the latent space, trade-off parameters for global and local terms are set equal to 0.02 and 5 throughout all the DA experiments. We train the ResNet50 model for 20000 steps with batch size of 36, following the standard protocols in [35], with data augmentation techniques like random flipping and cropping.

b, Dataset

The Office-31 [60] dataset consists of $4,110$ images, divide into 31 classes from three domains as presented in the main text, we conduct one more experiment on another dataset: ImageCLEF-DA, containing 12 categories from three public datasets: Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P). Each of these domains includes 50 images per class and 600 in total, which were resized to $3 \times 224 \times 224$ in our experiment. We evaluate all baselines in 6 adaptation scenarios as in previous studies: DAN [62], DANN [33], JAN [64], CDAN [35], and ETD [34].

Table 7.3: Accuracy (%) on ImageCLEF-DA of ResNet50 model [61] in unsupervised domain adaptation methods with results of related work are from original papers.

	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet	74.8	83.9	91.5	78.0	65.5	91.3	80.7
DAN	74.8	83.9	91.5	78.0	65.5	91.3	80.7
DANN	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN	76.8	88.4	94.8	89.5	74.2	91.7	85.8
CDAN	76.7	90.6	97.0	90.5	74.5	93.5	87.1
ETD	**81.0**	91.7	97.9	**93.3**	79.5	95.0	89.7
GLOT-DR	**81.0**	**93.8**	**98.0**	**93.3**	**79.7**	**96.3**	**90.4**

c, Experimental Results

As reported in Table. 7.3, the GLOT-DR approach outperforms the comparison methods on nearly all settings, except the pairs of I→P and C→I, where our scores are equal to ETD [34]. Our proposed method achieves 90.4% average accuracy overall, which is the highest compared to all baselines.

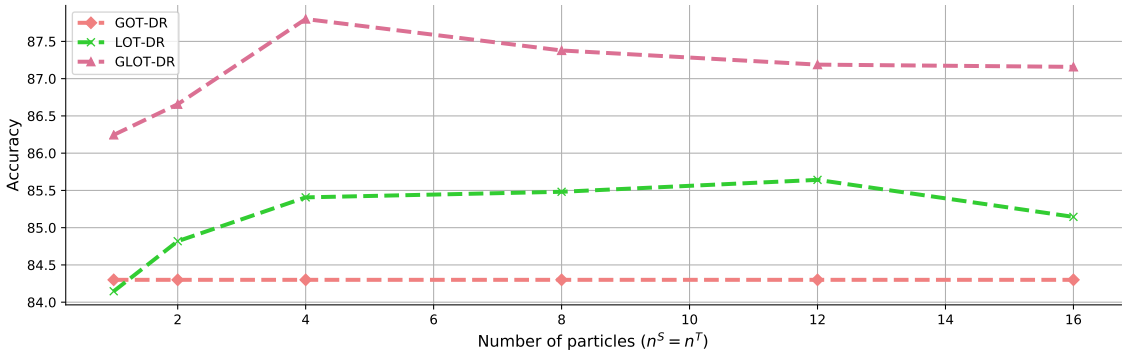

Figure 7.1: Classification accuracy (%) on Office-31 [60] of ResNet50 [61] model when varying the number of generated examples sampled from Project SVGD Algorithm.1.

Up till now, we have almost finished the needed experiments to examine the effectiveness of our method on domain adaptation. In this ultimate experiment, we illustrate the strength of our proposed regularization technique by varying the number of generated adversarial examples (i.e. $n^S$and n^T) from 1 to 16. Results are presented in Figure 7.1, where we perform extensive experiment via comparing GLOT-DR against its variants on different values of n^S, n^T. It can be easily seen that, increasing the number of generated samples can consistently improves the performance in both LOT-DR and GLOT-DR (note that in GOT-DR there is no local regularization term involved, thus there is no difference between different number of samples). Setting $n^S = n^T \geq 2$ helps LOT-DR surpass the performance of GOT-DR.

7.2.5 Experiments for SSL

a, Network architectures and Hyperparameters

In the semi supervised learning experiment, our main competitor is Virtual Adversarial Training (VAT) [3], we thus replicate their Conv-Large[2] architecture as:

$$
\begin{aligned}
&32 \times 32\,\text{RGB image} \rightarrow 3 \times 3\,\text{conv.}128\,\text{LReLU} \\
&\rightarrow 3 \times 3\,\text{conv.}128\,\text{LReLU} \rightarrow 3 \times 3\,\text{conv.}128\,\text{LReLU} \\
&\rightarrow 2 \times 2\,\text{MaxPool, stride}\,2 \rightarrow \text{Dropout}(0.5) \\
&\rightarrow 3 \times 3\,\text{conv.}256\,\text{LReLU} \rightarrow 3 \times 3\,\text{conv.}256\,\text{LReLU} \\
&\rightarrow 3 \times 3\,\text{conv.}256\,\text{LReLU} \rightarrow 2 \times 2\,\text{MaxPool, stride}\,2 \\
&\rightarrow \text{Dropout}(0.5) \rightarrow 3 \times 3\ \text{conv.}512\,\text{LReLU} \\
&\rightarrow 1 \times 1\,\text{conv.}256\,\text{LReLU} \rightarrow 1 \times 1\,\text{conv.}128\,\text{LReLU} \\
&\rightarrow \text{Global Average Pool}, 6 \times 6 \rightarrow 1 \times 1 \rightarrow \text{FC}_{128\times 10}
\end{aligned}
$$

We train the Conv-Large network in 600 epochs with batch size of 128 using SGD optimizer and cosine annealing learning rate scheduler [82]. The global and local trade-off parameters are ajusted by exponential rampup from [83]:

$$
\tau = \begin{cases} \exp^{-5(1-\frac{\text{epoch}}{\text{ramup length}})^2} & \text{epoch} < \text{rampup length} \\ 1 & \text{otherwise} \end{cases}
$$

with $\text{rampup length} = 30$ and initial trade-off for global and local terms are 0.1 and 10, respectively.

b, Experimental Results

In this section, we compare the training time in section 5.0.3 of LOT-DR and GLOT-DR against VAT in a single epoch. We repeat this process several times to get the average results, which are plotted in Figure 7.2. While VAT and LOT-DR run in almost equivalent time for all values of generated examples, GLOT-DR requires approximately 25% extra running time. Note that this is worthy because of the superior performance and great flexibility it brings on different scenarios.

7.2.6 Experiments for AML

a, General setting

We follow the setting in [66] for the experiment on adversarial machine learning domain. Specifically, the experiment has been conducted on CIFAR-10 dataset with

[2] LReLU indicates the Leaky ReLU [81] activation function with the negative slope equal to 0.1.

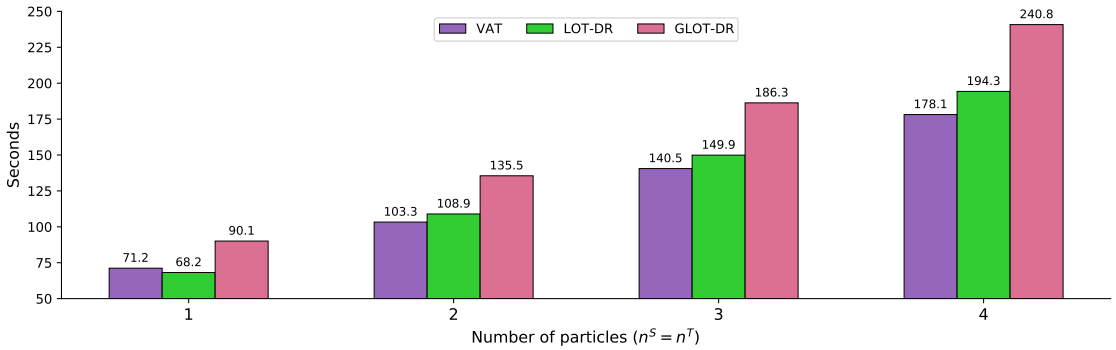

Figure 7.2: Running time of our proposed approach on: Intel(R) Xeon(R) CPU @ 2.00GHz CPU and Tesla P100 16GB VRAM GPU. Results are averaged over 3 runs.

ResNet18 architecture. All models have been trained with 110 epochs with SGD optimizer with momentum 0.9, weight decay 5×10^{-4}. The initial learning rate is 0.1 and reduce at epoch 100-th and 105-th with rate 0.1 as mentioned in [66].

b, Attack setting

We use different SOTA attacks to evaluate the defense methods including: (1) PGD attack [25] which is a gradient based attack with parameter $\{k = 200, \epsilon = 8/255, \eta = 2/255\}$ where k is the number of attack iterations, ϵ is the perturbation boundary and η is the step size of each iteration. (2) Auto-Attack (AA) [26] which is an ensemble methods of four different attacks. We use standard version with $\epsilon = 8/255$. (3) B&B attack [67] which is a decision based attack. Following [84], we initialized with the PGD attack with $k = 20, \epsilon = 8/255, \eta = 2/255$ then apply B&B attack with 200 steps. We use L_∞ for measuring the perturbation size and we use the full test set of 10k samples of the CIFAR-10 dataset in all experiments.

c, Baseline setting

We compare our method with PGD-AT [25] and TRADES [12] which are two well-known defense methods in AML. PGD-AT seeks the most violating examples that maximize the loss w.r.t. the true hard-label $\mathcal{L}_{CE}(h_\theta(x_a), y)$ while TRADES seeks the most divergent examples by maximizing the KL-divergence w.r.t. the current prediction (as consider as a soft-label) $\mathcal{L}_{KL}\left(h_\theta\left(x_a\right) \| h_\theta\left(x\right)\right)$. To be fair comparison, we use the same training setting for all methods, and succesfully reproduce performance of PGD-AT and TRADES as reported in [66]. We also compare with adversarial distributional training [4] (ADT-EXP and ADT-EXPAM) which assume that the adversarial distribution explicitly follows normal distribution.

REFERENCE

[1] I. Evtimov, K. Eykholt, E. Fernandes, *et al.*, "Robust physical-world attacks on machine learning models," *arXiv preprint arXiv:1707.08945*, vol. 2, no. 3, p. 4, 2017.

[2] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[3] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE TPAMI*, vol. 41, no. 8, pp. 1979–1993, 2018.

[4] Y. Dong, Z. Deng, T. Pang, J. Zhu, and H. Su, "Adversarial distributional training for robust deep learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8270–8283, 2020.

[5] L. Zhao, T. Liu, X. Peng, and D. Metaxas, "Maximum-entropy adversarial data augmentation for improved generalization and robustness," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 14 435–14 447.

[6] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018.

[7] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[8] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6023–6032.

[9] R. Gao, X. Chen, and A. J. Kleywegt, "Wasserstein distributional robustness and regularization in statistical learning," *arXiv e-prints*, arXiv–1712, 2017.

[10] J. Blanchet and K. Murthy, "Quantifying distributional model risk via optimal transport," *Mathematics of Operations Research*, vol. 44, no. 2, pp. 565–600, 2019.

[11] A. Sinha, H. Namkoong, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," in *International Conference on Learning Representations*, 2018.

[12] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proceedings of ICML*, PMLR, 2019, pp. 7472–7482.

[13] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," in *Proceedings of NeurIPS*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, 2016.

[14] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.

[15] J. C. Duchi, P. W. Glynn, and H. Namkoong, "Statistics of robust optimization: A generalized empirical likelihood approach," *Mathematics of Operations Research*, 2021.

[16] J. C. Duchi, T. Hashimoto, and H. Namkoong, "Distributionally robust losses against mixture covariate shifts," *Under review*, 2019.

[17] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," *arXiv preprint arXiv:1507.00677*, 2015.

[18] H. Namkoong and J. C. Duchi, "Stochastic gradient methods for distributionally robust optimization with f-divergences.," in *NIPS*, vol. 29, 2016, pp. 2208–2216.

[19] J. Blanchet, Y. Kang, and K. Murthy, "Robust wasserstein profile inference and applications to machine learning," *Journal of Applied Probability*, vol. 56, no. 3, pp. 830–857, 2019.

[20] R. Gao and A. J. Kleywegt, "Distributionally robust stochastic optimization with wasserstein distance," *arXiv preprint arXiv:1604.02199*, 2016.

[21] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, "Wasserstein distributionally robust optimization: Theory and applications in machine learning," in *Operations Research & Management Science in the Age of Analytics*, INFORMS, 2019, pp. 130–166.

[22] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations," *arXiv e-prints*, arXiv–1505, 2015.

[23] S. Shafieezadeh-Abadeh, P. M. Esfahani, and D. Kuhn, "Distributionally robust logistic regression," *arXiv preprint arXiv:1509.09259*, 2015.

[24] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[26] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proceedings of ICML*, 2020.

[27] A. Bui, T. Le, H. Zhao, *et al.*, "Improving adversarial robustness by enforcing local and global compactness," in *Proceedings of ECCV*, Springer, 2020, pp. 209–223.

[28] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," in *Advances in Neural Information Processing Systems*, 2019, pp. 1829–1839.

[29] A. Levine and S. Feizi, "Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 3938–3947.

[30] A. Najafi, S.-i. Maeda, M. Koyama, and T. Miyato, "Robustness to adversarial perturbations in learning from incomplete data," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5541–5551, 2019.

[31] M. Staib and S. Jegelka, "Distributionally robust deep learning as a generalization of adversarial training," *NIPS workshop on Machine Learning and Computer Security*, 2017.

[32] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty, "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 447–463.

[33] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[34] M. Li, Y.-M. Zhai, Y.-W. Luo, P.-F. Ge, and C.-X. Ren, "Enhanced transport distance for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 936–13 944.

[35] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *arXiv preprint arXiv:1705.10667*, 2017.

[36] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," *Advances in Neural Information Processing Systems*, vol. 31, pp. 998–1008, 2018.

[37] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, pp. 343–351, 2016.

[38] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5542–5550.

[39] ——, "Learning to generalize: Meta-learning for domain generalization," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[40] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1446–1455.

[41] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci, "Best sources forward: Domain generalization through source-specific nets," in *2018 25th IEEE international conference on image processing (ICIP)*, IEEE, 2018, pp. 1353–1357.

[42] H. Wang, A. Liu, Z. Yu, Y. Yue, and A. Anandkumar, *Distributionally robust learning for unsupervised domain adaptation*, 2021.

[43] G. Monge, "Histoire de l'académie royale des sciences de paris," *De L'imprimerie Royale*, 1781.

[44] L. V. Kantorovich, "On the translocation of masses," *Journal of mathematical sciences*, vol. 133, no. 4, pp. 1381–1382, 2006.

[45] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[46] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[47] Q. Liu, J. Lee, and M. Jordan, "A kernelized stein discrepancy for goodness-of-fit tests," in *International conference on machine learning*, PMLR, 2016, pp. 276–284.

[48] C. Stein, "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables," in *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, University of California Press, vol. 6, 1972, pp. 583–603.

[49] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[50] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[51] T. Lin, N. Ho, X. Chen, M. Cuturi, and M. I. Jordan, "Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm," in *NeurIPS*, 2020, pp. 5368–5380.

[52] T. Lin, N. Ho, and M. Jordan, "On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms," in *ICML*, 2019, pp. 3982–3991.

[53] T. Lin, N. Ho, and M. I. Jordan, "On the efficiency of the Sinkhorn and Greenkhorn algorithms and their acceleration for optimal transport," *ArXiv Preprint: 1906.01437*, 2019.

[54] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[55] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.

[56] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[57] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *arXiv preprint arXiv:1805.12018*, 2018.

[58] H. Wang, Z. He, Z. L. Lipton, and E. P. Xing, "Learning robust representations by projecting superficial statistics out," in *International Conference on Learning Representations*, 2019.

[59] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[60] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*, Springer, 2010, pp. 213–226.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.

[62] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings

of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 97–105.

[63] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016.

[64] ——, "Deep transfer learning with joint adaptation networks," in *International conference on machine learning*, PMLR, 2017, pp. 2208–2217.

[65] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.

[66] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, "Bag of tricks for adversarial training," in *International Conference on Learning Representations*, 2020.

[67] W. Brendel, J. Rauber, M. Kümmerer, I. Ustyuzhaninov, and M. Bethge, "Accurate, reliable and fast robustness evaluation," *arXiv preprint arXiv:1907.01003*, 2019.

[68] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, "Stochastic optimization for large-scale optimal transport," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016.

[69] Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the International Conference on Learning Representations*, 2014.

[71] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *Proceedings of the International Conference on Learning Representations Workshops*, 2014.

[72] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[73] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference*, 2016.

[74] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[76] O. Russakovsky, J. Deng, H. Su, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[77] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[78] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[79] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of ICML*, PMLR, 2015, pp. 1180–1189.

[80] J. S. Denker, W. Gardner, H. P. Graf, *et al.*, "Neural network recognizer for hand-written zip code digits," in *Advances in neural information processing systems*, Citeseer, 1989, pp. 323–331.

[81] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, Citeseer, vol. 30, 2013, p. 3.

[82] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[83] L. Samuli and A. Timo, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations (ICLR)*, vol. 4, 2017, p. 6.

[84] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *Advances in Neural Information Processing Systems*, vol. 33, 2020.