Global-Local Regularization Via Distributional Robustness

Hoang Phan[◇] **Trung Le**[◇†] **Trung Phung**[+] **Anh Bui**[†] **Nhat Ho**[‡] **Dinh Phung**[◇†]
VinAI Research[◇] Monash University, Australia[†] Johns Hopkins University[+] University of Texas, Austin[‡]

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 paper, 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 enabling 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.

1 Introduction

As the Wasserstein (WS) distance is a powerful and convenient tool of measuring closeness between distributions, Wasserstein Distributional Robustness (WDR) 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 (Blanchet and Murthy, 2019; Sinha et al., 2018):

$$\sup_{\tilde{\mathbb{P}}:\mathcal{W}_c\left(\mathbb{P},\tilde{\mathbb{P}}\right)<\epsilon} \mathop{\mathbb{E}}_{\tilde{Z}\sim\tilde{\mathbb{P}}}\left[r\left(\tilde{Z}\right)\right], \tag{1}$$

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 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. 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 $Z = \tilde{Z}$, (Blanchet and Murthy, 2019; Sinha et al., 2018) showed the *dual* form for Eq. (1) is:

$$\inf_{\lambda\geq 0}\left\{\lambda\epsilon + \mathop{\mathbb{E}}_{Z\sim\mathbb{P}}\left[\sup_{\tilde{z}}\left\{r\left(\tilde{Z}\right) - \lambda c\left(\tilde{Z},Z\right)\right\}\right]\right\}. \tag{2}$$

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 (Blanchet and Murthy, 2019; Sinha et al., 2018). The fact that r engages only $\tilde{Z} = \left(\tilde{X},\tilde{Y}\right) \sim \tilde{\mathbb{P}}$ certainly restricts the modeling capacity of (2). The reasons are as follows. Firstly, for each anchor Z, the most challenging sample $\tilde{Z}$ is currently defined as the one maximizing $\sup_{\tilde{Z}}\left\{r(\tilde{Z}) - \lambda c(Z,\tilde{Z})\right\}$, where $r(\tilde{Z})$ is inherited from the primal form (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 (Zhang et al., 2019)). 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,

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

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* to impose 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 (Blanchet and Murthy, 2019; Sinha et al., 2018) (i.e., Eq. (2)).

Technically, our solution turns solving the inner maximization in the dual form (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) (Liu and Wang, 2016) 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. We note that the global regularization term is crucial for many downstream tasks (see Section 3.1 for more details).
- We propose a closed-form solution for our GLOT-DR without involving the dual form in (Blanchet and Murthy, 2019; Sinha et al., 2018) (i.e., Eq. (2)). We note that the dual form (2) is *not computationally tractable* 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.

2 Related Work

Distributional robustness (DR). DR is an attractive framework for improving machine learning models in terms of robustness and generalization. Its underlying idea 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 (Ben-Tal et al., 2013; Duchi et al., 2021, 2019; Miyato et al., 2015; Namkoong and Duchi, 2016) or Wasserstein distance (Blanchet et al., 2019; Gao and Kleywegt, 2016; Kuhn et al., 2019; Mohajerin Esfahani and Kuhn, 2015; Shafieezadeh-Abadeh et al., 2015) can be employed. Other works (Blanchet and Murthy, 2019; Sinha et al., 2018) 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 (Goodfellow et al., 2014), PGD (Madry et al., 2018), and Auto-Attack (Croce and Hein, 2020). Among various kinds of defense approaches, Adversarial Training (AT), originating in (Goodfellow et al., 2014), 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 (Goodfellow et al., 2014) or most divergent examples (Zhang et al., 2019)), (2) different searching strategies (e.g., non-iterative FGSM and Rand FGSM (Madry et al., 2018)), (3) additional regularization (e.g., adding constraints in the latent space (Bui et al., 2020; Zhang and Wang, 2019)). Inspired by the potential of DR, it has been applied to enhance model robustness in (Dong et al., 2020; Levine and Feizi, 2020; Miyato et al., 2018; Sinha et al., 2018; Nguyen-Duc et al., 2022; Bui et al., 2022; Le et al., 2022; Hoang et al., 2020).

Transfer Learning (TL). Domain adaptation (DA) and domain generalization (DG) are two typical settings in TL. As for domain adaptation, (Ganin et al., 2016; Li et al., 2020; Long et al., 2017a; Nguyen et al., 2022; Le et al., 2021; Nguyen et al., 2021b,c,a) aim at training a model based on a labeled source domain to adapt to an unlabeled target domain, while the works in DG (Balaji et al., 2018; Bousmalis et al., 2016; Li et al., 2017, 2018, 2019; Mancini et al., 2018; Phung et al., 2021) 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 (Zhao et al., 2020) and DA in (Wang et al., 2021).

3 Proposed Approach

In this section, we first introduce the GLOT-DR framework and provide the theoretical development in Section 3.1. Then Section 3.2 presents the general training procedure of our proposed approach, and the detailed formulations of scenarios are described in the remainder of this section.

3.1 Our Framework

We propose a regularization technique based on optimal transport distributional robustness that can be 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

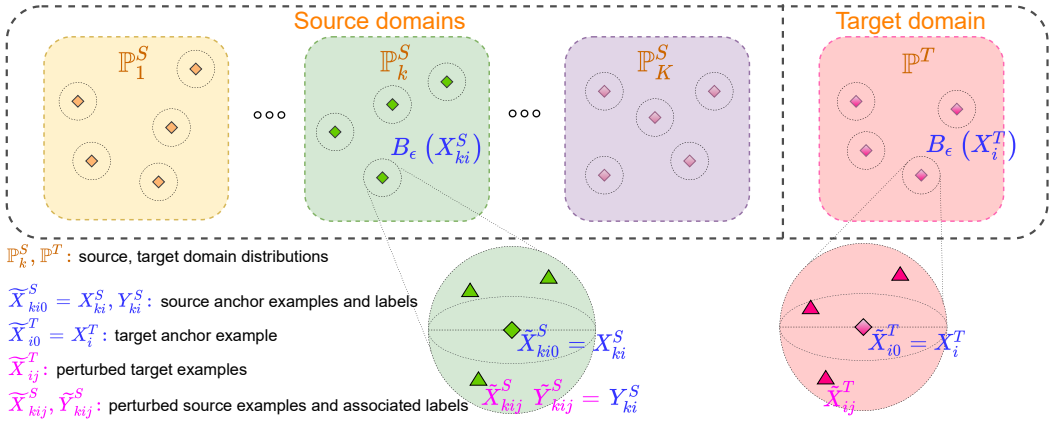

Figure 1: Overview of GLOT-DR. We sample $\left[X_{ki}^S, Y_{ki}^S\right]_{i=1}^{B_k^S}$ for each source domain, $\left[X_i^T\right]_{i=1}^{B^T}$ for the target domain, and define $Z, \tilde{Z}$ as in Eqs. (3,4). For $\left(Z, \tilde{Z}\right) \sim \gamma$ satisfying $\mathbb{E}_\gamma\left[\rho\left(Z, \tilde{Z}\right)\right]^{1/q} \leq \epsilon$, 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 $B_\epsilon\left(X_i^T\right)$.

the general setting along with the notations used throughout the paper and technical details of our framework.

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 $i = 1, \ldots, B_k^S$. 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 $\left(X_{ki}^S, Y_{ki}^S\right), X_i^T$ and their perturbed counterparts $\left(\tilde{X}_{kij}^S, \tilde{Y}_{kij}^S\right), \tilde{X}_{ij}^T$ (see Figure 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:

$$Z := \left[\left[\left[X_{kij}^S, Y_{kij}^S\right]_{k=1}^K\right]_{i=1}^{B_k^S}\right]_{j=0}^{n^S}, \left[\left[X_{ij}^T\right]_{i=1}^{B^T}\right]_{j=0}^{n^T}. \quad (3)$$

Here, each source sample is repeated $n^S + 1$ times $(X_{kij}^S, Y_{kij}^S) = (X_{ki}^S, Y_{ki}^S), \forall j$, while each target sample is repeated $n^T + 1$ times $X_{ij}^T = X_i^T, \forall j$. 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

$$\tilde{Z} := \left[\left[\left[\tilde{X}_{kij}^S, \tilde{Y}_{kij}^S\right]_{k=1}^K\right]_{i=1}^{B_k^S}\right]_{j=0}^{n^S}, \left[\left[\tilde{X}_{ij}^T\right]_{i=1}^{B^T}\right]_{j=0}^{n^T}. \quad (4)$$

Here we note that for $\tilde{X}_{kij}^S$, 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_{ki}^S. Similarly, for $\tilde{X}_{ij}^T$, 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_i^T.

We would like $\tilde{Z}$ to contain both: i) anchor examples, i.e., $\left(\tilde{X}_{ki0}^S, \tilde{Y}_{ki0}^S\right) = \left(X_{ki}^S, Y_{ki}^S\right)$ and $\tilde{X}_{i0}^T = X_i^T$; ii) n^S perturbed source samples $\left\{\left(\tilde{X}_{kij}^S, \tilde{Y}_{kij}^S\right)\right\}_{j=1}^{n^S}$ to $\left(X_{ki}^S, Y_{ki}^S\right)$ and n^T perturbed target samples $\left\{\tilde{X}_{ij}^T\right\}_{i=1}^{n^T}$ to X_i^T. 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., sat-

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

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

where ρ_l is a metric on the *label simplex* Δ_M and $q \geq 1$. 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 1 for the illustration). The reason is that if either (i) or (iii) is violated on a non-zero measurable set then $\mathcal{W}_\rho\left(\mathbb{P},\tilde{\mathbb{P}}\right)$ becomes infinity.

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 DR problem:

$$\min_{\theta,\phi}\max_{\tilde{\mathbb{P}}:\mathcal{W}_\rho\left(\mathbb{P},\tilde{\mathbb{P}}\right)\leq\epsilon} \mathbb{E}_{\tilde{Z}\sim\tilde{\mathbb{P}}}\left[r\left(\tilde{Z};\phi,\theta\right)\right]. \tag{5}$$

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. (5) 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.

We now define

$$\Gamma_\epsilon := \left\{\gamma : \gamma \in \bigcup_{\tilde{\mathbb{P}}}\Gamma\left(\mathbb{P},\tilde{\mathbb{P}}\right), \mathop{\mathbb{E}}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} \leq \epsilon\right\}$$

and show that the inner max problem in Eq. (5) is equivalent to searching in Γ_ϵ.

Lemma 3.1. *The optimization problem in Eq. (5) is equivalent to the following optimization problem:*

$$\min_{\theta,\phi}\max_{\gamma:\in\Gamma_\epsilon}\mathop{\mathbb{E}}_{(Z,\tilde{Z})\sim\gamma}\left[r\left(\tilde{Z};\phi,\theta\right)\right]. \tag{6}$$

To tackle the optimization problem (OP) in Eq. (6), we add the entropic regularization and arrive at the following OP:

$$\min_{\theta,\phi}\max_{\gamma:\in\Gamma_\epsilon}\left\{\mathop{\mathbb{E}}_{(Z,\tilde{Z})\sim\gamma}\left[r\left(\tilde{Z};\phi,\theta\right)\right] + \frac{1}{\lambda}\mathbb{H}\left(\gamma\right)\right\}, \tag{7}$$

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

It is worth noting that minimizing the entropy $\mathbb{H}\left(\gamma\right)$ encourages more uniform γ. Moreover, when λ becomes bigger, the optimal solution of the OP in Eq. (7) gets closer to that of (6). Additionally, the following theorem indicates the optimal solution of the inner max in the OP in Eq. (7).

Theorem 3.2. *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.(3) and Eq.(4), 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*

$$\gamma^*\left(Z,\tilde{Z}\right) = \prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S}p_k^S\left(X_{ki}^S,Y_{ki}^S\right)\prod_{i=1}^{B^T}\prod_{j=0}^{n^T}p^T\left(X_i^T\right) \prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S}q_{ki}^S\left(\tilde{X}_{kij}^S \mid X_{ki}^S,Y_{ki}^S;\psi\right)\prod_{i=1}^{B^T}\prod_{j=1}^{n^T}q_i^T\left(\tilde{X}_{ij}^T \mid X_i^T;\psi\right), \tag{8}$$

where $B_\epsilon\left(X\right) := \left\{X' : \left\|X'-X\right\|_p \leq \epsilon\right\}$ *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. (8) 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. (8) back to Eq. (6), we reach the following OP with $\psi = (\phi, \theta)$:

$$\min_{\psi} \mathop{\mathbb{E}}_{\forall k:\left(X_{ki}^S,Y_{ki}^S\right)_{i=1}^{B_k^S}\overset{\text{iid}}{\sim}\mathbb{P}_k^S, X_{1:B^T}^T\overset{\text{iid}}{\sim}\mathbb{P}^T} \left[r\left(\tilde{Z};\psi\right)\right], \quad (9)$$

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

$$\begin{aligned}\mathop{\mathbb{E}}_{\left[\tilde{X}_{kij}^S\right]_j\sim q_{ki}^S}&\left[\alpha s(X_{ki}^S,\tilde{X}_{kij}^S;\psi)+\ell(\tilde{X}_{kij}^S,Y_{ki}^S;\psi)\right]\\ &+\mathop{\mathbb{E}}_{\left[\tilde{X}_{ij}^T\right]_j\sim q_i^T}\left[\alpha s\left(X_i^T,\tilde{X}_{ij}^T;\psi\right)\right]\\ &+\beta r^g\left(\left[X_{ki}^S\right]_{k,i},\left[X_i^T\right]_i;\psi\right)\end{aligned} \quad (10)$$

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

As shown in Eq. (10), the perturbed examples $\tilde{X}_{kij}^S$ are sampled from the local distribution q_{ki}^S over the ball $B_\epsilon\left(X_{ki}^S\right)$, while the perturbed examples $\tilde{X}_{ij}^T$ are sampled from the local distribution q_i^T over the ball $B_\epsilon\left(X_i^T\right)$. Due to the formula of q_{ki}^S, the perturbed examples $\tilde{X}_{kij}^S$ tend to reach the high-likelihood region of q_{ki}^S or high-valued region for $\exp\left\{\lambda[\alpha s(X_{ki}^S,\tilde{X}_{kij}^S;\psi)+\ell(\tilde{X}_{kij}^S,Y_{ki}^S;\psi)]\right\}$. We hence can interpret $\tilde{X}_{kij}^S$ as adversarial examples that maximize $\lambda[\alpha s(X_{ki}^S,\tilde{X}_{kij}^S;\psi)+\ell(\tilde{X}_{kij}^S,Y_{ki}^S;\psi)]$. Subsequently, in (10), we update ψ to minimize $\lambda[\alpha s(X_{ki}^S,\tilde{X}_{kij}^S;\psi)+\ell(\tilde{X}_{kij}^S,Y_{ki}^S;\psi)]$ w.r.t. the perturbed adversarial examples. Similarly, we can interpret the perturbed examples $\tilde{X}_{ij}^T$.

Additionally, we can equip the global-regularization function $r^g\left(\left[X_{ki}^S\right]_{k,i},\left[X_i^T\right]_i;\psi\right)$ to suit various characteristics 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. Furthermore, 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.

3.2 Training Procedure of Our Approach

In what follows, we present how to solve the OP in Eq. (9) 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) (Liu and Wang, 2016; Phan et al., 2022) 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. (9) 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 regularization term $s\left(X,\tilde{X};\psi\right)$ as $\frac{1}{2}KL\left(f_\psi(X)\,\|f_\psi\left(\tilde{X}\right)\right)+\frac{1}{2}KL\left(f_\psi\left(\tilde{X}\right)\|f_\psi(X)\right)$.

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.

3.3 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 3.1, 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

$$\mathcal{W}_d\left(\frac{1}{B^S}\sum_{i=1}^{B^S}\delta_{U_i^S},\frac{1}{B^T}\sum_{j=1}^{B^T}\delta_{U_j^T}\right), \quad (11)$$

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

$$\begin{aligned}d\left(U_i^S,U_j^T\right) :=\ &\rho_d\left(g_\phi\left(X_i^S\right),g_\phi\left(X_j^T\right)\right)\\ &+\gamma\rho_l\left(h_\theta\left(g_\phi\left(X_i^S\right)\right),h_\theta\left(g_\phi\left(X_j^T\right)\right)\right),\end{aligned} \quad (12)$$

Table 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**

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

With the global term in Eq. (11), 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. (11) was inspected in DeepJDOT (Damodaran et al., 2018) for DA setting. Our approach is different from that approach in the local regularization term.

3.4 Setting for Domain Generalization

By setting $B^T = 0$ (i.e., not use any target data in training), our general framework in Section 3.1 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

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\frac{1}{K}\mathcal{W}_d\left(\tilde{\mathbb{P}}_{km}, \tilde{\mathbb{P}}_m\right), \tag{13}$$

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 $\tilde{\mathbb{P}}_m = \frac{1}{K}\sum_{k=1}^K \tilde{\mathbb{P}}_{km}$.

3.5 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 Eq. (9), to strengthen model robustness, we propose the following global term to move adversarial examples ($\sim \mathbb{P}^T$) to benign examples ($\sim \mathbb{P}_1^S$):

$$\mathcal{W}_d\left(\frac{1}{B_1^S}\sum_{i=1}^{B_1^S}\delta_{U_i^S}, \frac{1}{B_1^S n^S}\sum_{i=1}^{B_1^S}\sum_{j=1}^{n^S}\delta_{U_{ij}^S}\right), \tag{14}$$

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

$$d\left(U_i^S, U_{\bar{i}j}^S\right) = \mathbb{I}_{Y_{1i}^S = Y_{1\bar{i}}^S}\Bigg[\rho_d\left(g_\phi\left(X_{1i}^S\right), g_\phi\left(X_{1\bar{i}j}^S\right)\right) + \gamma\rho_l\left(h_\theta\left(g_\phi\left(X_{1i}^S\right)\right), h_\theta\left(g_\phi\left(X_{1\bar{i}j}^S\right)\right)\right)\Bigg], \tag{15}$$

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 Eq. (15), 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 equations. (11,13, and 14), we employ the entropic regularization dual form of WS, which was demonstrated to have favorable computational complexities (Lin et al., 2020, 2019a,b).

4 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 hyperparameters are presented in the supplementary material[1]. 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.

4.1 Experiments for DG

In DG experiments, our setup closely follows (Zhao et al., 2020). In particular, we validate our method on the CIFAR-C single domain generalization benchmark: train the model

[1]Our codes are available at https://github.com/VietHoang1512/GLOT

Table 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**

on either CIFAR-10 or CIFAR-100 dataset (Krizhevsky et al., 2009), then evaluate it on CIFAR-10-C or CIFAR-100-C (Hendrycks and Dietterich, 2019), correspondingly. In terms of network architectures, we use the exact backbones from (Zhao et al., 2020) 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 (Zhang et al., 2018), Cutout (DeVries and Taylor, 2017) and Cutmix (Yun et al., 2019), AutoDA (Cubuk et al., 2019), ADA (Volpi et al., 2018), and ME-ADA (Zhao et al., 2020).

Table 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 (Zhao et al., 2020) 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 our GLOT-DR.

Furthermore, we examine multi-source DG where the classifier needs to generalize from multiple source domains to an unseen target domain on the PACS dataset (Li et al., 2017). 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 (Bousmalis et al., 2016), L-CNN (Li et al., 2017), MLDG (Li et al., 2018), Fusion (Mancini et al., 2018), MetaReg (Balaji et al., 2018), Epi-FCR, AGG (Li et al., 2019), HEX (Wang et al., 2019b), and PAR (Wang et al., 2019a). Table 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).

4.2 Experiments for DA

In this section, we conduct experiments on the commonly used dataset for real-world unsupervised DA - Office-31 (Saenko et al., 2010), comprising images from three domains: Amazon (A), Webcam (W) and DSLR (D). Our proposed GLOT-DR is compared against baselines: ResNet-50 (He et al., 2016), DAN (Long et al., 2015), RTN (Long et al., 2016), DANN (Ganin et al., 2016), JAN (Long et al., 2017b), GTA (Sankaranarayanan et al., 2018), CDAN (Long et al., 2017a), DeepJDOT (Damodaran et al., 2018) and ETD (Li et al., 2020). 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 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→W task, 2.1% on W→A and 1.6% on average.

Table 3: Accuracy (%) on Office-31 (Saenko et al., 2010) of ResNet50 model (He et al., 2016) 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**

We further extensively investigate the role of different components in GLOT-DR. Specifically, the elimination of the global-regularization term in equation (11) 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 (Miyato et al., 2018). To be more specific, in the adversarial samples generation, we apply VAT by perturbing on the: (i) input space, (ii) latent space. Figure 2 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.

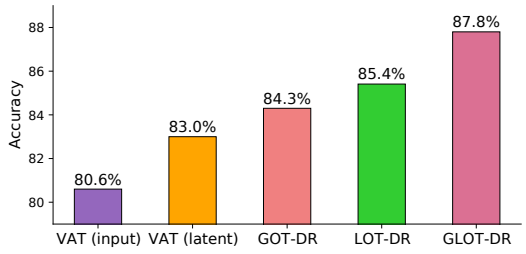

Figure 2: Average accuracy of ResNet50 (He et al., 2016) on Office-31: Comparision between GLOT-DR's variants and VAT (Miyato et al., 2018) on the input and latent spaces.

4.3 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 (Miyato et al., 2018), 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 3 (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.

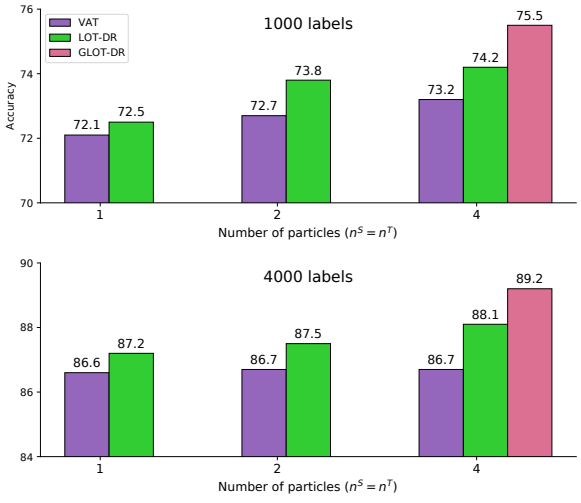

Figure 3: Accuracy (%) on CIFAR-10 of ConvLarge model in SSL settings when using 1,000 and 4,000 labeled examples (i.e. 100 and 400 labeled samples each class). Best viewed in color.

4.4 Experiments for AML

Table 4 shows the evaluation against adversarial examples. We compare our method with PGD-AT (Madry et al., 2018) and TRADES (Zhang et al., 2019), two well-known defense methods in AML and SAT (Bouniot et al., 2021). For the sake of fair comparison, we use the same adversarial training setting for all methods, which is carefully investigated in (Pang et al., 2020). We also compare with adversarial distributional training methods (Dong et al., 2020) (ADT-EXP and ADT-EXPAM), which assume that the adversarial distribution explicitly follows normal distribution. It can be seen from Table 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 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) (Madry et al., 2018), Auto-Attack (Croce and Hein, 2020) and B&B attack (Brendel et al., 2019), respectively, while NAT denotes the natural accuracy. Note that * results are taken from Pang et al. (Pang et al., 2020), while $^\diamond$ results are our reproduced results.

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
SAT	83.45	53.95	51.37	48.80	**49.40**
GLOT-DR	**84.13**	**54.13**	**53.1**8	**49.94**	**49.40**

5 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. Firstly, its formulation is not sufficiently rich to express a global regularization effect targeting many applications. Secondly, the dual form is not readily trainable to incorporate into the training of 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. Finally, 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 on standard benchmark datasets .

Acknowledgements. Trung Le and Dinh Phung were supported by the US Air Force grant FA2386-21-1-4049. Trung Le was supported by the ECR Seed grant of the Faculty of Information Technology, Monash University. Trung Le and Dinh Phung were also supported by the Australian Research Council (ARC) DP230101176 grant.

References

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008. 2, 7

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. (2013). Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357. 2

Blanchet, J., Kang, Y., and Murthy, K. (2019). Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857. 2

Blanchet, J. and Murthy, K. (2019). Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600. 1, 2

Bouniot, Q., Audigier, R., and Loesch, A. (2021). Optimal transport as a defense against adversarial attacks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5044–5051. IEEE. 8

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016). Domain separation networks. *Advances in neural information processing systems*, 29:343–351. 2, 7

Brendel, W., Rauber, J., Kümmerer, M., Ustyuzhaninov, I., and Bethge, M. (2019). Accurate, reliable and fast robustness evaluation. *arXiv preprint arXiv:1907.01003*. 8, 21

Bui, A., Le, T., Zhao, H., Montague, P., deVel, O., Abraham, T., and Phung, D. (2020). Improving adversarial robustness by enforcing local and global compactness. In *Proceedings of ECCV*, pages 209–223. Springer. 2

Bui, T. A., Le, T., Tran, Q., Zhao, H., and Phung, D. (2022). A unified wasserstein distributional robustness framework for adversarial training. *arXiv preprint arXiv:2202.13437*. 2

Croce, F. and Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of ICML*. 2, 8, 21

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7

Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018). Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–463. 6, 7

Denker, J. S., Gardner, W., Graf, H. P., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., Baird, H. S., and Guyon, I. (1989). Neural network recognizer for hand-written zip code digits. In *Advances in neural information processing systems*, pages 323–331. Citeseer. 17, 18

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

Dong, Y., Deng, Z., Pang, T., Zhu, J., and Su, H. (2020). Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283. 2, 8, 21

Duchi, J. C., Glynn, P. W., and Namkoong, H. (2021). Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*. 2

Duchi, J. C., Hashimoto, T., and Namkoong, H. (2019). Distributionally robust losses against mixture covariate shifts. *Under review*. 2

Ganin, Y. and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of ICML*, pages 1180–1189. PMLR. 17, 18

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030. 2, 7, 18

Gao, R. and Kleywegt, A. J. (2016). Distributionally robust stochastic optimization with wasserstein distance. *arXiv preprint arXiv:1604.02199*. 2

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc. 16

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. 2

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778. 7, 8, 18, 19

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*. 7, 17, 18

Hoang, Q., Le, T., and Phung, D. (2020). Parameterized rate-distortion stochastic encoder. In III, H. D. and Singh, A.,

editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4293–4303. PMLR. 2

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. 17

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*. 17

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *Citeseer*. 7, 17

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105. 17

Kuhn, D., Esfahani, P. M., Nguyen, V. A., and Shafieezadeh-Abadeh, S. (2019). Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations Research & Management Science in the Age of Analytics*, pages 130–166. INFORMS. 2

Le, T., Nguyen, T., Ho, N., Bui, H., and Phung, D. (2021). Lamda: Label matching deep domain adaptation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6043–6054. PMLR. 2

Le, T., Tuan Bui, A., Minh Tri Tue, L., Zhao, H., Montague, P., Tran, Q., and Phung, D. (2022). On global-view based defense via adversarial attack and defense risk guaranteed bounds. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11438–11460. PMLR. 2

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551. 17, 18

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. 17, 18

Levine, A. and Feizi, S. (2020). Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pages 3938–3947. PMLR. 2

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550. 2, 7, 17, 18

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2018). Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*. 2, 7

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. (2019). Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455. 2, 7

Li, M., Zhai, Y.-M., Luo, Y.-W., Ge, P.-F., and Ren, C.-X. (2020). Enhanced transport distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13944. 2, 7, 18

Lin, T., Ho, N., Chen, X., Cuturi, M., and Jordan, M. I. (2020). Fixed-support Wasserstein barycenters: Computational hardness and fast algorithm. In *NeurIPS*, pages 5368–5380. 6

Lin, T., Ho, N., and Jordan, M. (2019a). On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *ICML*, pages 3982–3991. 6

Lin, T., Ho, N., and Jordan, M. I. (2019b). On the efficiency of the Sinkhorn and Greenkhorn algorithms and their acceleration for optimal transport. *ArXiv Preprint: 1906.01437*. 6

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Proceedings of NeurIPS*, volume 29. 2, 5, 17

Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). Learning transferable features with deep adaptation networks. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 97–105, Lille, France. PMLR. 7, 18

Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2017a). Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*. 2, 7, 18

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2016). Unsupervised domain adaptation with residual transfer networks. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc. 7

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017b). Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR. 7, 18

Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*. 20

Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer. 19

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*. 2, 8, 21

Mancini, M., Bulo, S. R., Caputo, B., and Ricci, E. (2018). Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE international conference on image processing (ICIP)*, pages 1353–1357. IEEE. 2, 7

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 41(8):1979–1993. 2, 7, 8, 19, 20

Miyato, T., Maeda, S.-i., Koyama, M., Nakae, K., and Ishii, S. (2015). Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*. 2

Mohajerin Esfahani, P. and Kuhn, D. (2015). Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *arXiv e-prints*, pages arXiv–1505. 2

Namkoong, H. and Duchi, J. C. (2016). Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NIPS*, volume 29, pages 2208–2216. 2

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 17, 18

Nguyen, T., Le, T., Dam, N., Tran, Q. H., Nguyen, T., and Phung, D. Q. (2021a). Tidot: A teacher imitation learning approach for domain adaptation with optimal transport. In *IJCAI*, pages 2862–2868. 2

Nguyen, T., Le, T., Zhao, H., Tran, Q. H., Nguyen, T., and Phung, D. (2021b). Most: multi-source domain adaptation via optimal transport for student-teacher learning. In de Campos, C. and Maathuis, M. H., editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 225–235. PMLR. 2

Nguyen, T., Nguyen, V., Le, T., Zhao, H., Tran, Q. H., and Phung, D. (2022). Cycle class consistency with distributional optimal transport and knowledge distillation for unsupervised domain adaptation. In *Uncertainty in Artificial Intelligence*, pages 1519–1529. PMLR. 2

Nguyen, V.-A., Nguyen, T., Le, T., Tran, Q. H., and Phung, D. (2021c). Stem: An approach to multi-source domain adaptation with guarantees. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9352–9363. 2

Nguyen-Duc, T., Le, T., Zhao, H., Cai, J., and Phung, D. Q. (2022). Particle-based adversarial local distribution regularization. In *AISTATS*, pages 5212–5224. 2, 20

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. (2020). Bag of tricks for adversarial training. In *International Conference on Learning Representations*. 8, 20, 21

Phan, H., Tran, N., Le, T., Tran, T., Ho, N., and Phung, D. (2022). Stochastic multiple target sampling gradient descent. *Advances in Neural Information Processing Systems*. 5

Phung, T., Le, T., Vuong, T.-L., Tran, T., Tran, A., Bui, H., and Phung, D. (2021). On learning domain-invariant representations for transfer learning with multiple sources. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27720–27733. Curran Associates, Inc. 2

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252. 17

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer. 7, 18, 19

Samuli, L. and Timo, A. (2017). Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, volume 4, page 6. 20

Sankaranarayanan, S., Balaji, Y., Castillo, C. D., and Chellappa, R. (2018). Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512. 7

Shafieezadeh-Abadeh, S., Esfahani, P. M., and Kuhn, D. (2015). Distributionally robust logistic regression. *arXiv preprint arXiv:1509.09259*. 2

Sinha, A., Namkoong, H., and Duchi, J. (2018). Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*. 1, 2

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *Proceedings of the International Conference on Learning Representations Workshops*. 17

Tramer, F., Carlini, N., Brendel, W., and Madry, A. (2020). On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33. 21

Volpi, R., Namkoong, H., Sener, O., Duchi, J., Murino, V., and Savarese, S. (2018). Generalizing to unseen do-

mains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*. 7, 17

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. (2019a). Learning robust global representations by penalizing local predictive power. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 7, 17

Wang, H., He, Z., Lipton, Z. L., and Xing, E. P. (2019b). Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*. 7

Wang, H., Liu, A., Yu, Z., Yue, Y., and Anandkumar, A. (2021). Distributionally robust learning for unsupervised domain adaptation. 2

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500. 17

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032. 7

Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British Machine Vision Conference*. 17

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*. 7

Zhang, H. and Wang, J. (2019). Defense against adversarial attacks using feature scattering-based adversarial training. In *Advances in Neural Information Processing Systems*, pages 1829–1839. 2

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *Proceedings of ICML*, pages 7472–7482. PMLR. 1, 2, 8, 21

Zhao, L., Liu, T., Peng, X., and Metaxas, D. (2020). Maximum-entropy adversarial data augmentation for improved generalization and robustness. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14435–14447. Curran Associates, Inc. 2, 6, 7, 17

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

A Proofs of Our Theory Development

We here give the proof for the equivalence in optimizing two equations (5) and (6) in Section A.1. Then, we detail how to derive the optimization formulations (3.2) and (9) for solving the problems discussed in Section 3.1.

A.1 Proof of Lemma 3.1

Let

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

be the optimal solution of the inner max in equation (6). 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 (5). Let $\tilde{\mathbb{P}}$ be a feasible solution of the inner max in equation (5), 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 $\mathop{\mathbb{E}}_{(Z,\tilde{Z})\sim\gamma}\left[\rho\left(Z,\tilde{Z}\right)\right]^{1/q} \leq \epsilon$ or $\gamma \in \Gamma_\epsilon$. We have

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

We reach the conclusion that $\tilde{\mathbb{P}}^*$ is the optimal solution of the inner max in equation (5). That concludes our proof.

A.2 Proof of Theorem 3.2

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

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

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

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

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

$$\rho\left(Z, \tilde{Z}\right) := \sum_{i=1}^{B^T} \sum_{j=1}^{n^T} \left\| X_{ij}^T - \tilde{X}_{ij}^T \right\|_p^q + \sum_{k=1}^{K} \sum_{i=1}^{B_k^S} \sum_{j=1}^{n^S} \left\| X_{kij}^S - \tilde{X}_{kij}^S \right\|_p^q.$$

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

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

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

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

We derive as

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

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

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

We can equivalently turn the optimization problem in equation (7) as follows:

$$\max_{\gamma \in \Gamma} \mathop{\mathbb{E}}_{(Z, \tilde{Z}) \sim \gamma} \left[r\left(\tilde{Z}; \phi, \theta\right) \right] + \frac{1}{\lambda} \mathbb{H}(\gamma) \qquad \text{s.t. : } \operatorname{supp}(\gamma) = \bigcup_{Z \in \operatorname{supp}(\mathbb{P})} B_Z. \tag{16}$$

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:

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

where $p(Z)$ 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 (16) is as follows:

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

where the integral w.r.t Z over on $\operatorname{supp}(\mathbb{P})$ 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

$$\tilde{p}^{*}\left(\tilde{Z}|Z\right) = \frac{\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}}{\int_{B_Z}\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}d\tilde{Z}},$$
$$\gamma^{*}\left(Z,\tilde{Z}\right) = p\left(Z\right)\frac{\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}}{\int_{B_Z}\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}d\tilde{Z}}. \tag{17}$$

Finally, by noting that

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

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

And

$$\int_{B_Z}\exp\left\{\lambda r\left(\tilde{Z};\phi,\theta\right)\right\}d\tilde{Z}$$
$$= \exp\left\{\lambda\beta r^g\left(\tilde{Z};\psi\right)\right\}\prod_{k=1}^{K}\prod_{i=1}^{B_k^S}\prod_{j=0}^{n_k^S}\int_{B_\epsilon\left(X_{ki}^S\right)}\exp\left\{\lambda[\alpha s(X_{ki}^S, \tilde{X}_{kij}^S;\psi) + \ell(\tilde{X}_{kij}^S, Y_{ki}^S;\psi)]\right\}d\tilde{X}_{kij}^S$$
$$\prod_{i=1}^{B^T}\prod_{j=1}^{n^T}\int_{B_\epsilon\left(X_i^T\right)}\exp\left\{\lambda\alpha s\left(X_i^T, \tilde{X}_{ij}^T;\psi\right)\right\}d\tilde{X}_{ij}^T,$$

we reach the conclusion.

A.3 Proof of the optimization problem in equation (9)

By substituting $\gamma^{*}\left(Z,\tilde{Z}\right)$ in equation (17) back to the objective function in (6), we obtain

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

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

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

As a consequence, we gain the final optimization problem.

B 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 $\{X_i^0\}_{i=1}^n$.
Output: A set of particles $\{X_i\}_{i=1}^n$ that approximates the local distribution corresponding to $\tilde{p}(\cdot)$.
for $l = 1$ to L **do**
$X_i^{l+1} = \prod_{B_\epsilon(X)} \left[X_i^l + \eta_l \hat{\phi}^*(X_i^l)\right]$
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

B.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 (Genevay et al., 2016). First, they proposed to add an entropic regularization term to the primal form:

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

where ϵ is the regularization rate, $D_{KL}(\cdot\|\cdot)$ 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 (18) 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

$$
\begin{aligned}
\mathcal{W}_d^\epsilon(\mathbb{P}, \mathbb{Q}) &= \max_\phi \left\{ \int \phi_\epsilon^c(\mathbf{x}) \, \mathrm{d}\mathbb{Q}(\mathbf{x}) + \int \phi(\mathbf{y}) \, \mathrm{d}\mathbb{P}(\mathbf{y}) \right\} \\
&= \max_\phi \left\{ \mathop{\mathbb{E}}_{\mathbb{Q}} [\phi_\epsilon^c(\mathbf{x})] + \mathop{\mathbb{E}}_{\mathbb{P}} [\phi(\mathbf{y})] \right\}, \qquad (18)
\end{aligned}
$$

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

In order to calculate the global WS-related regularization terms in equations. . (11, 13, and 14), 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: $\mathrm{FC}_{\mathrm{latent\_dim} \times 512} \to \mathrm{ReLU} \to \mathrm{FC}_{512 \times 1}$ is used throughout experiments. Note that the $\mathrm{latent\_dim}$ depends on the main network.

Additionally, the distance ρ_d in equation (12) 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 (7) is 0.1.

B.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 paper (Liu and Wang, 2016).

B.3 Experiments for DG

B.3.1 Network Architecture and Hyperparamters

As mentioned in the main paper, we incorporate well-studied backbones for our experiments, following the implementation of for single domain generalization tasks in (Zhao et al., 2020). In particular:

- LeNet5 (LeCun et al., 1989) 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 (Kingma and Ba, 2014) with learning rate 10^{-5} and train for 15000 iteration in total with batch size of 32.
- CIFAR-C [2] experiment uses 4 different backbone architectures, namely: All Convolutional Network (AllConvNet) (Springenberg et al., 2014), DenseNet (Huang et al., 2017) , WideResNet (Zagoruyko and Komodakis, 2016), and ResNeXt (Xie et al., 2017). 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 (Krizhevsky et al., 2012) pretrained on ImageNet (Russakovsky et al., 2015) in the PACS experiment. 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 $\frac{\mathrm{iter}}{\#\mathrm{num\_iter}}$ in iter-th iteration. We train AlexNet for 45.000 iterations with SGD optimizer and 10^{-3} learning rate.

B.3.2 Datasets and Baselines

We present the details on each dataset used in domain generalization experiments in Table. 5. Digits datasets: MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), MNIST-M (Ganin and Lempitsky, 2015), SYN (Ganin and Lempitsky, 2015), USPS (Denker et al., 1989) - each contains 10 classes from $0 - 9$, which are resized to 32×32 images in our experiment. CIFAR-10-C (Hendrycks and Dietterich, 2019), and CIFAR-100-C (Hendrycks and Dietterich, 2019) consist of corrupted images from the original CIFAR (Krizhevsky et al., 2009) datasets with 15 corruptions types applied. In terms of multi-source domain generalization, we test our proposed model on PACS dataset (Li et al., 2017), 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 (Wang et al., 2019a), (iii) ADA (Volpi et al., 2018) and (iv) ME-ADA (Zhao et al., 2020). 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).

[2]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 5: Details on the domain generalization benchmark datasets

Dataset	# classes	Shape
MNIST (LeCun et al., 1998)	10	32×32
SVHN (Netzer et al., 2011)	10	32×32
MNIST-M (Ganin and Lempitsky, 2015)	10	32×32
SYN (Ganin and Lempitsky, 2015)	10	32×32
USPS (Denker et al., 1989)	10	32×32
CIFAR-10-C (Hendrycks and Dietterich, 2019)	15	$3 \times 32 \times 32$
CIFAR-100-C (Hendrycks and Dietterich, 2019)	15	$3 \times 32 \times 32$
PACS (Li et al., 2017)	7	$3 \times 224 \times 224$

B.3.3 Experimental Results

Table 6: Average classification accuracy (%) on MNIST benchmark, we first train the LeNet5 (LeCun et al., 1989) 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**

Table. 6 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 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, $(0.79\% - 1.12\%)$.

B.4 Experiments for DA

B.4.1 Network architectures and Hyperparameters

The ResNet50 (He et al., 2016) 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 (Long et al., 2017a), with data augmentation techniques like random flipping and cropping.

B.4.2 Dataset

The Office-31 (Saenko et al., 2010) dataset consists of $4,110$ images, divide into 31 classes from three domains as presented in the main paper, 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 (Long et al., 2015), DANN (Ganin et al., 2016), JAN (Long et al., 2017b), CDAN (Long et al., 2017a), and ETD (Li et al., 2020).

B.4.3 Experimental Results

As reported in Table. 7, 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 (Li et al., 2020). Our proposed method achieves 90.4% average

accuracy overall, which is the highest compared to all baselines.

Table 7: Accuracy (%) on ImageCLEF-DA of ResNet50 model (He et al., 2016) 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**

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 4, 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.

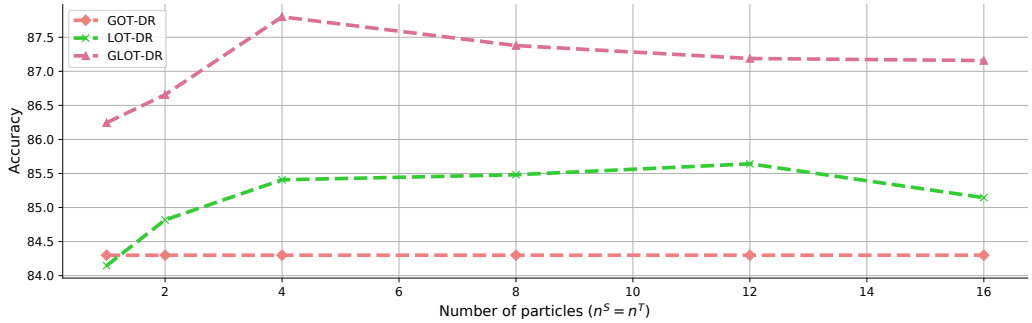

Figure 4: Classification accuracy (%) on on Office-31 (Saenko et al., 2010) of ResNet50 (He et al., 2016) model when varying the number of generated examples sampled from Project SVGD Algorithm.1.

B.5 Experiments for SSL

B.5.1 Network architectures and Hyperparameters

In the semi supervised learning experiment, our main competitor is Virtual Adversarial Training (VAT) (Miyato et al., 2018), we thus replicate their Conv-Large[3] architecture as:

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

[3]LReLU indicates the Leaky ReLU (Maas et al., 2013) activation function with the negative slope equal to 0.1.

We train the Conv-Large network in 600 epochs with batch size of 128 using SGD optimizer and cosine annealing learning rate scheduler (Loshchilov and Hutter, 2016). The global and local trade-off parameters are ajusted by exponential rampup from (Samuli and Timo, 2017):

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

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

B.5.2 Experimental Results

In this section, we compare the training time in section 4.3 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 5. 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.

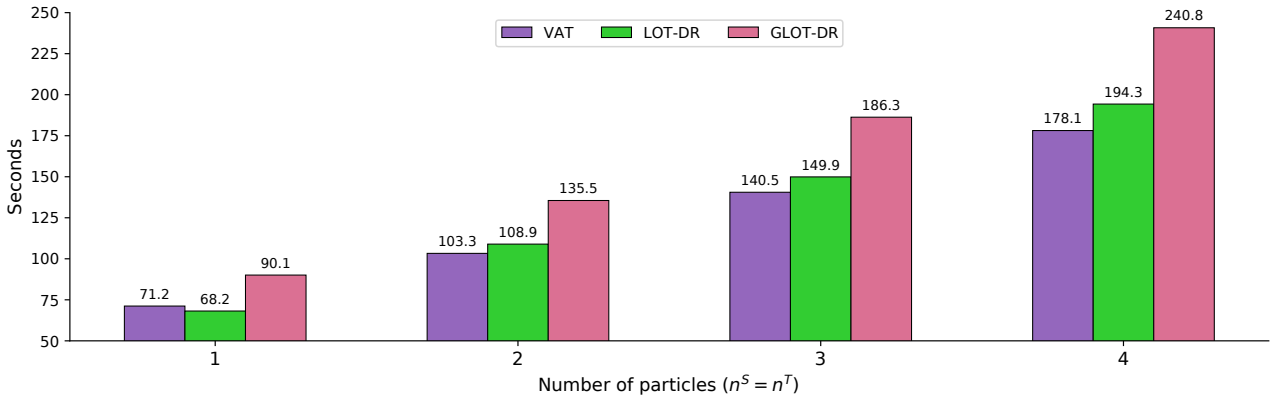

Figure 5: Running time of our proposed approach on: Intel(R) Xeon(R) CPU @ 2.00GHz CPU and Tesla P100 16GB VRAM GPU. Results are averaged over 3 runs.

Furthermore, we also compare our proposed GLOT-DR with VAT (Miyato et al., 2018) and Nguyen-Duc, et al. (Nguyen-Duc et al., 2022) in the SSL scenario, utilizing the protocol from table 1 from their paper. As can be seen from Table 8, our method is still better in all experiments ($> 1\%$), especially when the number of particles $n = 8$, it outperforms all baselines by large margins.

Table 8: Semi-supervised learning on Conv-Large backbone.

n particle(s)	1	2	4	8
VAT	0.8601	0.8611	0.858	0.856
Nguyen-Duc, et al.	0.867	0.876	0.883	0.872
GLOT-DR	**0.881**	**0.888**	**0.892**	**0.894**

B.6 Experiments for AML

B.6.1 General setting

We follow the setting in (Pang et al., 2020) for the experiment on adversarial machine learning domain. Specifically, the experiment has been conducted on CIFAR-10 dataset with 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 (Pang et al., 2020).

B.6.2 Attack setting

We use different SOTA attacks to evaluate the defense methods including: (1) PGD attack (Madry et al., 2018) 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) (Croce and Hein, 2020) which is an ensemble methods of four different attacks. We use standard version with $\epsilon = 8/255$. (3) B&B attack (Brendel et al., 2019) which is a decision based attack. Following (Tramer et al., 2020), 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.

B.6.3 Baseline setting

We compare our method with PGD-AT (Madry et al., 2018) and TRADES (Zhang et al., 2019) 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) $\mathcal{L}_{KL}\left(h_\theta\left(x_a\right) \parallel h_\theta\left(x\right)\right)$. 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 (Pang et al., 2020). We also compare with adversarial distributional training (Dong et al., 2020) (ADT-EXP and ADT-EXPAM) which assume that the adversarial distribution explicitly follows normal distribution.