arXiv:1805.12018v2 [cs.CV] 6 Nov 2018

Generalizing to Unseen Domains via Adversarial Data Augmentation

Riccardo Volpi[*,†]
Istituto Italiano di Tecnologia

Hongseok Namkoong[*]
Stanford University

Ozan Sener
Intel Labs

John Duchi
Stanford University

Vittorio Murino
Istituto Italiano di Tecnologia
Università di Verona

Silvio Savarese
Stanford University

Abstract

We are concerned with learning models that generalize well to different *unseen* domains. We consider a worst-case formulation over data distributions that are near the source domain in the feature space. Only using training data from a single source distribution, we propose an iterative procedure that augments the dataset with examples from a fictitious target domain that is "hard" under the current model. We show that our iterative scheme is an adaptive data augmentation method where we append adversarial examples at each iteration. For softmax losses, we show that our method is a data-dependent regularization scheme that behaves differently from classical regularizers that regularize towards zero (*e.g.*, ridge or lasso). On digit recognition and semantic segmentation tasks, our method learns models improve performance across a range of a priori unknown target domains.

1 Introduction

In many modern applications of machine learning, we wish to learn a system that can perform uniformly well across multiple populations. Due to high costs of data acquisition, however, it is often the case that datasets consist of a limited number of population sources. Standard models that perform well when evaluated on the validation dataset—usually collected from the same population as the training dataset—often perform poorly on populations different from that of the training data [15, 3, 1, 32, 38]. In this paper, we are concerned with generalizing to populations different from the training distribution, in settings where we have no access to any data from the unknown target distributions. For example, consider a module for self-driving cars that needs to generalize well across weather conditions and city environments unexplored during training.

A number of authors have proposed domain adaptation methods (for example, see [9, 39, 36, 26, 40]) in settings where a fully labeled source dataset and an unlabeled (or partially labeled) set of examples from fixed target distributions are available. Although such algorithms can successfully learn models that perform well on known target distributions, the assumption of a priori fixed target distributions can be restrictive in practical scenarios. For example, consider a semantic segmentation algorithm used by a robot: every task, robot, environment and camera configuration will result in a different target distribution, and these diverse scenarios can be identified only after the model is trained and deployed, making it difficult to collect samples from them.

In this work, we develop methods that can learn to better *generalize* to new unknown domains. We consider the restrictive setting where training data only comes from a single source domain. Inspired

[*]Equal contribution.
[†]Work done while author was a Visiting Student Researcher at Stanford University.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

by recent developments in distributionally robust optimization and adversarial training [34, 20, 12], we consider the following worst-case problem around the (training) source distribution P_0

$$\underset{\theta \in \Theta}{\text{minimize}} \sup_{P: D(P, P_0) \leq \rho} \mathbb{E}_P[\ell(\theta; (X, Y))]. \tag{1}$$

Here, $\theta \in \Theta$ is the model, $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ is a source data point with its labeling, $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is the loss function, and $D(P, Q)$ is a distance metric on the space of probability distributions.

The solution to worst-case problem (1) guarantees good performance against data distributions that are distance ρ away from the source domain P_0. To allow data distributions that have different support to that of the source P_0, we use Wasserstein distances as our metric D. Our distance will be defined on the semantic space [3], so that target populations P satisfying $D(P, P_0) \leq \rho$ represent realistic covariate shifts that preserve the same semantic representation of the source (*e.g.*, adding color to a greyscale image). In this regard, we expect the solution to the worst-case problem (1)—the model that we wish to learn—to have favorable performance across covariate shifts in the semantic space.

We propose an iterative procedure that aims to solve the problem (1) for a small value of ρ at a time, and does stochastic gradient updates to the model θ with respect to these fictitious worst-case target distributions (Section 2). Each iteration of our method uses small values of ρ, and we provide a number of theoretical interpretations of our method. First, we show that our iterative algorithm is an adaptive data augmentation method where we add adversarially perturbed samples—at the current model—to the dataset (Section 3). More precisely, our adversarially generated samples roughly correspond to *Tikhonov regularized Newton-steps* [21, 25] on the loss in the semantic space. Further, we show that for softmax losses, each iteration of our method can be thought of as a data-dependent regularization scheme where we regularize towards the parameter vector corresponding to the true label, instead of regularizing towards zero like classical regularizers such as ridge or lasso.

From a practical viewpoint, a key difficulty in applying the worst-case formulation (1) is that the magnitude of the covariate shift ρ is a priori unknown. We propose to learn an ensemble of models that correspond to different distances ρ. In other words, our iterative method generates a collection of datasets, each corresponding to a different inter-dataset distance level ρ, and we learn a model for each of them. At test time, we use a heuristic method to choose an appropriate model from the ensemble.

We test our approaches on a simple digit recognition task, and a more realistic semantic segmentation task across different seasons and weather conditions. In both settings, we observe that our method allows to learn models that improve performance across a priori unknown target distributions that have varying distance from the original source domain.

Related work

The literature on adversarial training [10, 34, 20, 12] is closely related to our work, since the main goal is to devise training procedures that learn models robust to fluctuations in the input. Departing from imperceptible attacks considered in adversarial training, we aim to learn models that are resistant to larger perturbations, namely out-of-distribution samples. Sinha et al. [34] proposes a principled adversarial training procedure, where new images that maximize some risk are generated and the model parameters are optimized with respect to those adversarial images. Being devised for defense against *imperceptible* adversarial attacks, the new images are learned with a loss that penalizes differences between the original and the new ones. In this work, we rely on a minimax game similar to the one proposed by Sinha et al. [34], but we impose the constraint in the semantic space, in order to allow our adversarial samples from a fictitious distribution to be different at the pixel level, while sharing the same semantics.

There is a substantial body of work on *domain adaptation* [15, 3, 32, 9, 39, 36, 26, 40], which aims to better generalize to a priori *fixed* target domains whose labels are unknown at training time. This setup is different from ours in that these algorithms require access to samples from the target distribution during training. *Domain generalization* methods [28, 22, 27, 33, 24] that propose different ways to better generalize to unknown domains are also related to our work. These algorithms require

[3] By *semantic space* we mean learned representations since recent works [7, 16] suggest that distances in the space of learned representations of high capacity models typically correspond to semantic distances in visual space.

the training samples to be drawn from different domains (while having access to the domain labels during training), not a single source, a limitation that our method does not have. In this sense, one could interpret our problem setting as *unsupervised* domain generalization. Tobin et al. [37] proposes *domain randomization*, which applies to simulated data and creates a variety of random renderings with the simulator, hoping that the real world will be interpreted as one of them. Our goal is the same, since we aim at obtaining data distributions more similar to the real world ones, but we accomplish it by actually *learning* new data points, and thus making our approach applicable to any data source and without the need of a simulator.

Hendrycks and Gimpel [13] suggest that a good empirical way to detect whether a test sample is out-of-distribution for a given model is to evaluate the statistics of the softmax outputs. We adapt this idea in our setting, learning ensemble of models trained with our method and choosing at test time the model with the greatest maximum softmax value.

2 Method

The worst-case formulation (1) over domains around the source P_0 hinges on the notion of distance $D(P, P_0)$, that characterizes the set of unknown populations we wish to generalize to. Conventional notions of Wasserstein distance used for adversarial training [34] are defined with respect to the original input space $\mathcal{X}$, which for images corresponds to raw pixels. Since our goal is to consider fictitious target distributions corresponding to realistic covariate shifts, we define our distance on the semantic space. Before properly defining our setup, we first give a few notations. Letting p the dimension of output of the last hidden layer, we denote $\theta = (\theta_c, \theta_f)$ where $\theta_c \in \mathbb{R}^{p\times m}$ is the set of weights of the final layer, and θ_f is the rest of the weights of the network. We denote by $g(\theta_f; x)$ the output of the embedding layer of our neural network. For example, in the classification setting, m is the number of classes and we consider the softmax loss

$$\ell(\theta; (x, y)) := -\log \frac{\exp\left(\theta_{c,y}^\top g(\theta_f; x)\right)}{\sum_{j=1}^m \exp\left(\theta_{c,j}^\top g(\theta_f; x)\right)} \tag{2}$$

where $\theta_{c,j}$ is the j-th column of the classification layer weights $\theta_c \in \mathbb{R}^{p\times m}$.

Wasserstein distance on the semantic space On the space $\mathbb{R}^p \times \mathcal{Y}$, consider the following transportation cost c—cost of moving mass from (z, y) to (z', y')

$$c((z, y), (z', y')) := \frac{1}{2}\|z - z'\|_2^2 + \infty \cdot \mathbf{1}\{y \neq y'\}.$$

The transportation cost takes value ∞ for data points with different labels, since we are only interested in perturbation to the marginal distribution of Z. We now define our notion of distance on the semantic space. For inputs coming from the original space $\mathcal{X} \times \mathcal{Y}$, we consider the transportation cost c_θ defined with respect to the output of the last hidden layer

$$c_\theta((x, y), (x', y')) := c((g(\theta_f; x), y), (g(\theta_f; x'), y'))$$

so that c_θ measures distance with respect to the feature mapping $g(\theta_f; x)$. For probability measures P and Q both supported on $\mathcal{X} \times \mathcal{Y}$, let $\Pi(P, Q)$ denote their couplings, meaning measures M with $M(A, \mathcal{X} \times \mathcal{Y}) = P(A)$ and $M(\mathcal{X} \times \mathcal{Y}, A) = Q(A)$. Then, we define our notion of distance by

$$D_\theta(P, Q) := \inf_{M\in\Pi(P,Q)} \mathbb{E}_M[c_\theta((X, Y), (X', Y'))]. \tag{3}$$

Armed with this notion of distance on the semantic space, we now consider a variant of the worst-case problem (1) where we replace the distance with D_θ (3), our adaptive notion of distance defined on the semantic space

$$\underset{\theta\in\Theta}{\text{minimize}} \ \sup_P \{\mathbb{E}_P[\ell(\theta; (X, Y))] : D_\theta(P, P_0) \leq \rho\}.$$

Computationally, the above supremum over probability distributions is intractable. Hence, we consider the following Lagrangian relaxation with penalty parameter γ

$$\underset{\theta\in\Theta}{\text{minimize}} \ \sup_P \{\mathbb{E}_P[\ell(\theta; (X, Y))] - \gamma D_\theta(P, P_0)\}. \tag{4}$$

Algorithm 1 Adversarial Data Augmentation

Input: original dataset $\{X_i, Y_i\}_{i=1,...,n}$ and initialized weights θ_0
Output: learned weights θ

Initialize: $\theta \leftarrow \theta_0$
for $k = 1, ..., K$ **do** ▷ Run the minimax procedure K times
  for $t = 1, ..., T_{\min}$ **do**
    Sample (X_t, Y_t) uniformly from dataset
    $\theta \leftarrow \theta - \alpha \nabla_\theta \ell(\theta; (X_t, Y_t))$
  Sample $\{X_i, Y_i\}_{i=1,...,n}$ uniformly from the dataset
  for $i = 1, \ldots, n$ **do**
    $X_i^k \leftarrow X_i$
    for $t = 1, \ldots, T_{\max}$ **do**
      $X_i^k \leftarrow X_i^k + \eta \nabla_x \left\{ \ell(\theta; (X_i^k, Y_i)) - \gamma c_\theta((X_i^k, Y_i), (X_i, Y_i)) \right\}$
    Append (X_i^k, Y_i^k) to dataset
for $t = 1, \ldots, T$ **do**
  Sample (X, Y) uniformly from dataset
  $\theta \leftarrow \theta - \alpha \nabla_\theta \ell(\theta; (X, Y))$

Taking the dual reformulation of the penalty relaxation (4), we can obtain an efficient solution procedure. The following result is a minor adaptation of [2, Theorem 1]; to ease notation, let us define the robust surrogate loss

$$\phi_\gamma(\theta; (x_0, y_0)) := \sup_{x \in \mathcal{X}} \{ \ell(\theta; (x, y_0)) - \gamma c_\theta((x, y_0), (x_0, y_0)) \} . \tag{5}$$

Lemma 1. *Let $\ell : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$ be continuous. For any distribution Q and any $\gamma \geq 0$, we have*

$$\sup_P \{ \mathbb{E}_P[\ell(\theta; (X, Y))] - \gamma D_\theta(P, Q) \} = \mathbb{E}_Q[\phi_\gamma(\theta; (X, Y))]. \tag{6}$$

In order to solve the penalty problem (4), we can now perform stochastic gradient descent procedures on the robust surrogate loss ϕ_γ. Under suitable conditions [5], we have

$$\nabla_\theta \phi_\gamma(\theta; (x_0, y_0)) = \nabla_\theta \ell(\theta; (x_\gamma^\star, y_0)), \tag{7}$$

where $x_\gamma^\star = \arg\max_{x \in \mathcal{X}} \{ \ell(\theta; (x, y_0)) - \gamma c_\theta((x, y_0), (x_0, y_0)) \}$ is an adversarial perturbation of x_0 at the current model θ. Hence, computing gradients of the robust surrogate ϕ_γ requires solving the maximization problem (5). Below, we consider an (heuristic) iterative procedure that iteratively performs stochastic gradient steps on the robust surrogate ϕ_γ.

Iterative Procedure We propose an iterative training procedure where two phases are alternated: a *maximization* phase where new data points are learned by computing the inner maximization problem (5) and a *minimization* phase, where the model parameters are updated according to stochastic gradients of the loss evaluated on the adversarial examples generated from the maximization phase. The latter step is equivalent to stochastic gradient steps on the robust surrogate loss ϕ_γ, which motivates its name. The main idea here is to iteratively learn "hard" data points from fictitious target distributions, while preserving the semantic features of the original data points.

Concretely, in the k-th *maximization* phase, we compute n adversarially perturbed samples at the current model $\theta \in \Theta$

$$X_i^k \in \arg\max_{x \in \mathcal{X}} \left\{ \ell(\theta; (x, Y_i)) - \gamma c_\theta((x, Y_i), (X_i^{k-1}, Y_i)) \right\} \tag{8}$$

where X_i^0 are the original samples from the source distribution P_0. The *minimization phase* then performs repeated stochastic gradient steps on the augmented dataset $\{X_i^k, Y_i\}_{0 \leq k \leq K, 1 \leq i \leq n\}$. The maximization phase (8) can be efficiently computed for smooth losses if $x \mapsto c_{\theta^{k-1}}((x, Y_i), (X_i^{k-1}, Y_i))$ is strongly convex [34, Theorem 2]; for example, this is provably true for any linear network. In practice, we use gradient ascent steps to solve for worst-case examples (8); see Algorithm 1 for the full description of our algorithm.

Ensembles for classification The hyperparameter γ—which is inversely proportional to ρ, the distance between the fictitious target distribution and the source—controls the ability to generalize outside the source domain. Since target domains are unknown, it is difficult to choose an appropriate level of γ a priori. We propose a heuristic ensemble approach where we train s models $\left\{\theta^0, ..., \theta^s\right\}$. Each model is associated with a different value of γ, and thus to fictitious target distributions with varying distances from the source P_0. To select the best model at test time—inspired by Hendrycks and Gimpel [13]—given a sample x, we select the model $\theta^{u^\star(x)}$ with the greatest softmax score

$$u^\star(x) := \underset{1\le u\le s}{\arg\max} \max_{1\le j\le k} \theta_{c,j}^{u\top} g(\theta_f^u; x). \tag{9}$$

3 Theoretical Motivation

In our iterative algorithm (Algorithm 1), the *maximization* phase (8) was a key step that augmented the dataset with adversarially perturbed data points, which was followed by standard stochastic gradient updates to the model parameters. In this section, we provide some theoretical understanding of the augmentation step (8). First, we show that the augmented data points (8) can be interpreted as *Tikhonov regularized Newton-steps* [21, 25] in the semantic space under the current model. Roughly speaking, this gives the sense in which Algorithm 1 is an adaptive data augmentation algorithm that adds data points from fictitious "hard" target distributions. Secondly, recall the robust surrogate loss (5) whose stochastic gradients were used to update the model parameters θ in the *minimization* step (Eq (7)). In the classification setting, we show that the robust surrogate (5) roughly corresponds to a novel data-dependent regularization scheme on the softmax loss ℓ. Instead of penalizing towards zero like classical regularizers (*e.g.*, ridge or lasso), our data-dependent regularization term penalizes deviations from the parameter vector corresponding to that of the true label.

3.1 Adaptive Data Augmentation

We now give an interpretation for the augmented data points in the maximization phase (8). Concretely, we fix $\theta \in \Theta$, $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$, and consider an ϵ-maximizer

$$x_\epsilon^\star \in \epsilon\text{-}\underset{x\in\mathcal{X}}{\arg\max} \left\{\ell(\theta; (x, y_0)) - \gamma c_\theta((x, y_0), (x_0, y_0))\right\}.$$

We let $z_0 := g(\theta_f; x_0) \in \mathbb{R}^p$, and abuse notation by using $\ell(\theta; (z_0, y_0)) := \ell(\theta; (x_0, y_0))$. In what follows, we show that the feature mapping $g(\theta_f; x_\epsilon^\star)$ satisfies

$$g(\theta_f; x_\epsilon^\star) = \underbrace{g(\theta_f; x_0) + \frac{1}{\gamma}\left(I - \frac{1}{\gamma}\nabla_{zz}\ell(\theta; (z_0, y_0))\right)^{-1} \nabla_z \ell(\theta; (z_0, y_0))}_{=:\, \widehat{g}_{\mathrm{newton}}(\theta_f; x_0)} + O\left(\sqrt{\frac{\epsilon}{\gamma}} + \frac{1}{\gamma^2}\right). \tag{10}$$

Intuitively, this implies that the adversarially perturbed sample $x_\epsilon^\star$ is drawn from a fictitious target distribution where probability mass on $z_0 = g(\theta_f; x_0)$ was transported to $\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)$. We note that the transported point in the semantic space corresponds to a *Tikhonov regularized Newton-step* [21, 25] on the loss $z \mapsto \ell(\theta; (z, y_0))$ at the current model θ. Noting that computing $\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)$ involves backsolves on a large dense matrix, we can interpret our gradient ascent updates in the maximization phase (8) as an iterative scheme for approximating this quantity.

We assume sufficient smoothness, where we use $\|H\|$ to denote the ℓ_2-operator norm of a matrix H.

Assumption 1. *There exists $L_0, L_1 > 0$ such that, for all $z, z' \in \mathbb{R}^p$, we have $|\ell(\theta; (z, y_0)) - \ell(\theta; (z', y_0))| \le L_0 \|z - z'\|_2$ and $\|\nabla_z \ell(\theta; (z, y_0)) - \nabla_z \ell(\theta; (z', y_0))\|_2 \le L_1 \|z - z'\|_2$.*

Assumption 2. *There exists $L_2 > 0$ such that, for all $z, z' \in \mathbb{R}^p$, we have $\|\nabla_{zz}\ell(\theta; (z, y_0)) - \nabla_{zz}\ell(\theta; (z', y_0))\| \le L_2 \|z - z'\|_2$.*

Then, we have the following bound (10) whose proof we defer to Appendix A.1.

Theorem 1. *Let Assumptions 1, 2 hold. If Im$(g(\theta_f; \cdot)) = \mathbb{R}^p$ and $\gamma > L_1$, then*

$$\|g(\theta_f; x_\epsilon^\star) - \widehat{g}_{\mathrm{newton}}(\theta_f; x_0)\|_2^2 \le \frac{2\epsilon}{\gamma - L_1} + \frac{L_2}{3(\gamma - L_1)}\left\{\left(\frac{5L_0}{\gamma}\right)^3 + \left(\frac{L_0}{\gamma - L_1}\right)^3 + \left(\frac{2\epsilon}{\gamma}\right)^{\frac{3}{2}}\right\}.$$

3.2 Data-Dependent Regularization

In this section, we argue that under suitable conditions on the loss,

$$\phi_\gamma(\theta;(z,y)) = \ell(\theta;(z,y)) + \frac{1}{\gamma}\left\|\nabla_z \ell(\theta;(z,y))\right\|_2^2 + O\left(\frac{1}{\gamma^2}\right).$$

For classification problems, we show that the robust surrogate loss (5) corresponds to a particular data-dependent regularization scheme. Let $\ell(\theta;(x,y))$ be the m-class softmax loss (2) given by

$$\ell(\theta;(x,y)) = -\log p_y(\theta,x) \;\text{ where }\; p_j(\theta,x) := \frac{\exp(\theta_{c,j}^\top g(\theta,x))}{\sum_{l=1}^m \exp(\theta_{c,l}^\top g(\theta_f;x))}.$$

where $\theta_{c,j} \in \mathbb{R}^p$ is the j-th row of the classification layer weight $\theta_c \in \mathbb{R}^{p\times m}$. Then, the robust surrogate ϕ_γ is an approximate regularizer on the classification layer weights θ_c

$$\phi_\gamma(\theta;(x,y)) = \ell(\theta;(x,y)) + \frac{1}{\gamma}\left\|\theta_{c,y} - \sum_{j=1}^m p_j(\theta,x)\theta_{c,j}\right\|_2^2 + O\left(\frac{1}{\gamma^2}\right). \tag{11}$$

The expansion (11) shows that the robust surrogate (5) is roughly equivalent to data-dependent regularization where we minimize the distance between $\sum_{j=1}^m p_j(\theta,x)\theta_{c,j}$, our "average estimated linear classifier", to $\theta_{c,y}$, the linear classifier corresponding to the true label y. Concretely, for any fixed $\theta \in \Theta$, we have the following result where we use $L(\theta) := 2\max_{1\le j'\le m}\|\theta_{c,j'}\|_2 \sum_{j=1}^m \|\theta_{c,j}\|_2$ to ease notation. See Appendix A.3 for the proof.

Theorem 2. *If $Im(g(\theta_f;\cdot)) = \mathbb{R}^p$ and $\gamma > L(\theta)$, the softmax loss* (2) *satisfies*

$$\frac{1}{\gamma + L(\theta)}\left\|\theta_{c,y} - \sum_{j=1}^m p_j(\theta,x)\theta_{c,j}\right\|_2^2 \le \phi_\gamma(\theta,(x,y)) - \ell(\theta,(x,y)) \le \frac{1}{\gamma - L(\theta)}\left\|\theta_{c,y} - \sum_{j=1}^m p_j(\theta,x)\theta_{c,j}\right\|_2^2.$$

4 Experiments

We evaluate our method for both classification and semantic segmentation settings, following the evaluation scenarios of domain adaptation techniques [9, 39, 14], though in our case the target domains are unknown at training time. We summarize our experimental setup including implementation details, evaluation metrics and datasets for each task.

Digit classification We train on MNIST [19] dataset and test on MNIST-M [9], SVHN [30], SYN [9] and USPS [6]. We use $10,000$ digit samples for training and evaluate our models on the respective test sets of the different target domains, using accuracy as a metric. In order to work with comparable datasets, we resized all the images to 32×32, and treated images from MNIST and USPS as RGB. We use a ConvNet [18] with architecture *conv-pool-conv-pool-fc-fc-softmax* and set the hyperparameters $\alpha = 0.0001$, $\eta = 1.0$, $T_{\min} = 100$ and $T_{\max} = 15$. In the minimization phase, we use Adam [17] with batch size equal to 32[4]. We compare our method against the Empirical Risk Minimization (ERM) baseline and different regularization techniques (Dropout [35], ridge).

Semantic scene segmentation We use the SYTHIA[31] dataset for semantic segmentation. The dataset contains images from different locations (we use *Highway*, *New York-like City* and *Old European Town*), and different weather/time/date conditions (we use *Dawn*, *Fog*, *Night*, *Spring* and *Winter*. We train models on a source domain and test on other domains, using the standard mean Intersection Over Union *(mIoU)* metric to evaluate our performance [8]. We arbitrarily chose images from the left front camera throughout our experiments. For each one, we sample 900 random images (resized to 192×320 pixels) from the training set. We use a Fully Convolutional Network (FCN) [23], with a ResNet-50 [11] body and set the hyperparameters $\alpha = 0.0001$, $\eta = 2.0$, $T_{\min} = 500$ and $T_{\max} = 50$. For the minimization phase, we use Adam [17] with batch size equal to 8. We compare our method against the ERM baseline.

4.1 Results on Digit Classification

In this section, we present and discuss the results on the digit classification experiment. Firstly, we are interested in analyzing the role of the semantic constraint we impose. Figure 1a *(top)* shows

[4]Models were implemented using Tensorflow, and training procedures were performed on NVIDIA GPUs. Code is available at `https://github.com/ricvolpi/generalize-unseen-domains`

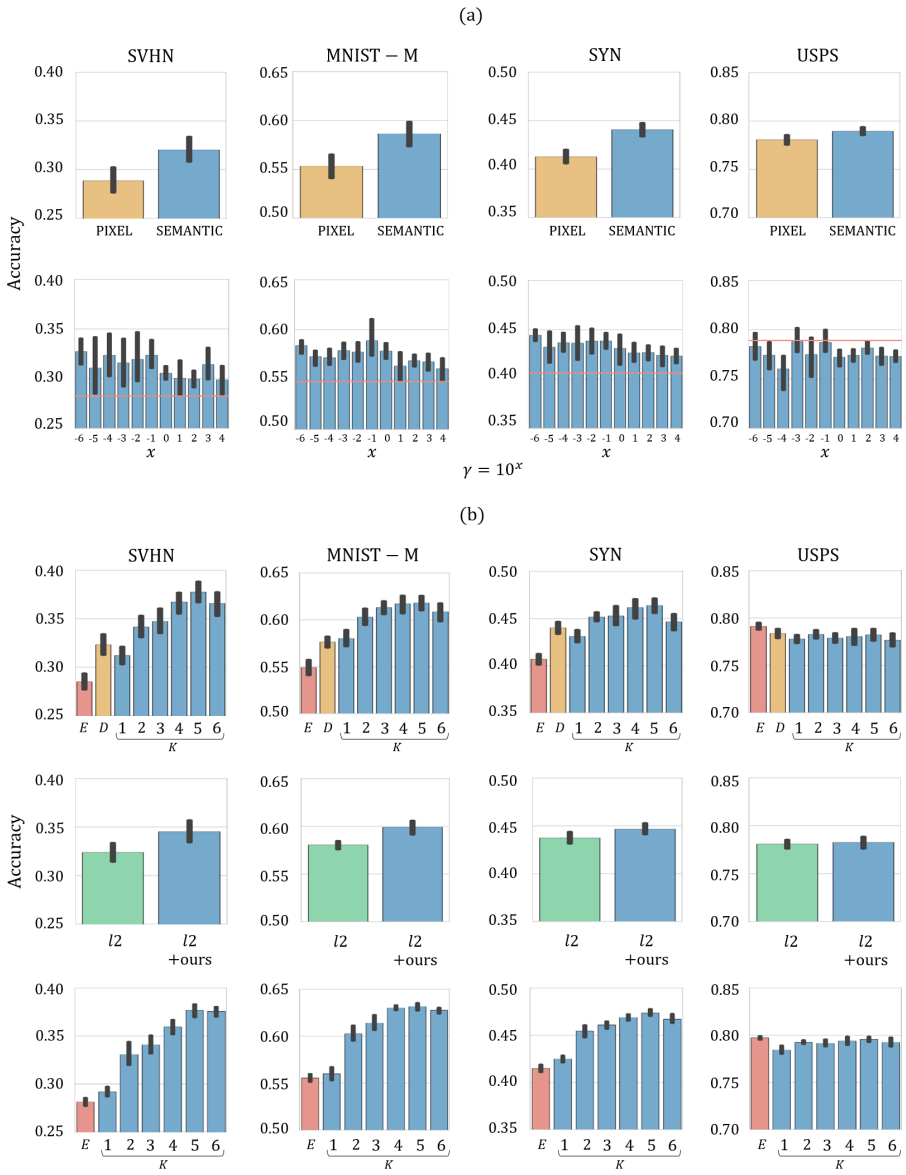

Figure 1. Results associated with models trained with $10,000$ MNIST samples and tested on SVHN, MNIST-M, SYN and USPS (1^{st}, 2^{nd}, 3^{rd} and 4^{th} columns, respectively). *Panel (a), top:* comparison between distances in the pixel space (yellow) and in the semantic space (blue), with $\gamma = 10^4$ and $K = 1$. *Panel (a), bottom:* comparison between our method with $K = 2$ and different γ values (blue bars) and ERM (red line). *Panel (b), top:* comparison between our method with $\gamma = 1.0$ and different number of iterations K (blue), ERM (red) and Dropout [35] (yellow). *Panel (b), middle:* comparison between models regularized with ridge (green) and with ridge + our method with $\gamma = 1.0$ and $K = 1$ (blue). *Panel (b), bottom:* results related to the ensemble method, using models trained with our methods with different number of iterations K (blue) and using models trained via ERM (red). The reported results are obtained by averaging over 10 different runs; black bars indicate the range of accuracy spanned.

performances associated with models trained with Algorithm 1 with $K = 1$ and $\gamma = 10^4$, with the constraint in the semantic space (as discussed in Section 2) and in the pixel space [34] (blue and yellow bars, respectively). Figure 1a *(bottom)* shows performances of models trained with our method using different values of the hyperparameter γ (with $K = 2$) and with ERM (blue bars and red lines, respectively). These plots show (i) that moving the constraint on the semantic space carries benefits when models are tested on unseen domains and (ii) that models trained with Algorithm 1 outperform models train with ERM for any value of γ on out-of-sample domains (SVHN, MNIST-M

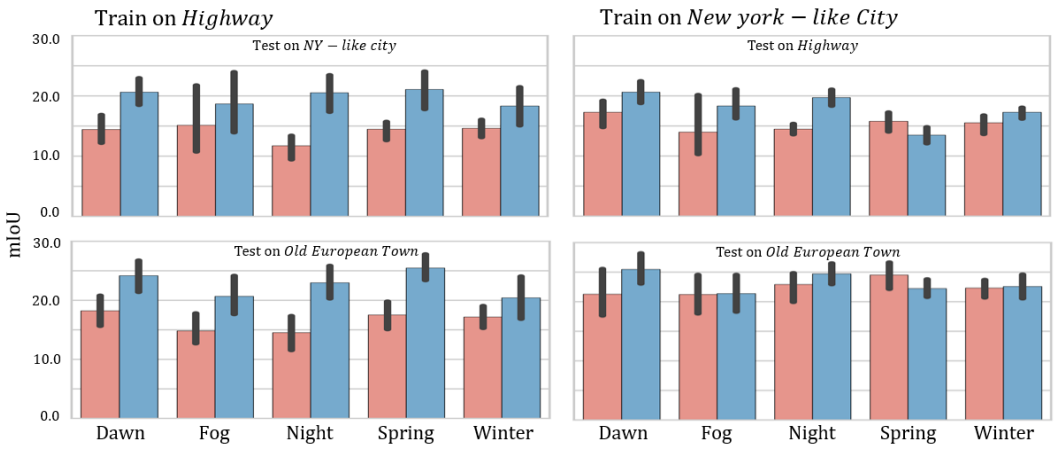

Figure 2. Results obtained with semantic segmentation models trained with ERM *(red)* and our method with $K = 1$ and $\gamma = 1.0$ *(blue)*. *Leftmost* panels are associated with models trained on *Highway*, *rightmost* panels are associated with models trained on *New York-like City*. Test datasets are *Highway*, *New York-like City* and *Old European Town*.

and SYN). The latter result is a rather desired achievement, since this hyperparameter cannot be properly cross-validated. On USPS, our method causes accuracy to drop since MNIST and USPS are very similar datasets, thus the image domain that USPS belongs to is not explored by our algorithm during the training procedure, which optimizes for worst case performance.

Figure 1b *(top)* reports results related to models trained with our method (blue bars), varying the number of iterations K and fixing $\gamma = 1.0$, and results related to ERM (red bars) and Dropout [35] (yellow bars). We observe that our method improves performances on SVHN, MNIST-M and SYN, outperforming both ERM and Dropout [35] statistically significantly. In Figure 1b *(middle)*, we compare models trained with ridge regularization (green bars) with models trained with Algorithm 1 (with $K = 1$ and $\gamma = 1.0$) and ridge regularization (blue bars); these results show that our method can potentially benefit from other regularization approaches, as in this case we observed that the two effects sum up. We further report in Appendix B a comparison between our method and an unsupervised domain adaptation algorithm (ADDA [39]), and results associated with different values of the hyperparameters γ and K.

Finally, we report the results obtained by learning an ensemble of models. Since the hyperparameter γ is nontrivial to set a priori, we use the softmax confidences (9) to choose which model to use at test time. We learn ensemble of models, each of which is trained by running Algorithm 1 with different values of the γ as $\gamma = 10^{-i}$, with $i = \{0, 1, 2, 3, 4, 5, 6\}$. Figure 1b *(bottom)* shows the comparison between our method with different numbers of iterations K and ERM (blue and red bars, respectively). In order to separate the role of ensemble learning, we learn an ensemble of baseline models each corresponding to a different initialization. We fix the number of models in the ensemble to be the same for both the baseline (ERM) and our method. Comparing Figure 1b *(bottom)* with Figure 1b *(top)* and Figure 1a *(bottom)*, our ensemble approach achieves higher accuracy in different testing scenarios. We observe that our out-of-sample performance improves as the number of iterations K gets large. Also in the ensemble setting, for the USPS dataset we do not see any improvement, which we conjecture to be an artifact of the trade-off between good performance on domains far away from training, and those closer.

4.2 Results on Semantic Scene Segmentation

We report a comparison between models trained with ERM and models trained with our method (Algorithm 1 with $K = 1$). We set $\gamma = 1.0$ in every experiment, but stress that this is an arbitrary value; we did not observe a strong correlation between the different values of γ and the general behavior of the models in this case. Its role was more meaningful in the ensemble setting where each model is associated with a different level of robustness, as discussed in Section 2. In this setting, we do not apply the ensemble approach, but only evaluate the performances of the single models. The

main reason for this choice is the fact that the heuristics developed to choose the correct model at test time in effect cannot be applied in a straightforward fashion to a semantic segmentation problem.

Figure 2 reports numerical results obtained. Specifically, leftmost plots report results associated with models trained on sequences from the *Highway* split and tested on the *New York-like City* and the *Old European Town* splits (*top-left* and *bottom-left*, respectively); rightmost plots report results associated with models trained on sequences from the *New York-like City* split and tested on the *Highway* and the *Old European Town* splits (*top-right* and *bottom-right*, respectively). The training sequences (*Dawn*, *Fog*, *Night*, *Spring* and *Winter*) are indicated on the x-axis. Red and blue bars indicate average mIoUs achieved by models trained with ERM and by models trained with our method, respectively. These results were calculated by averaging over the mIoUs obtained with each model on the different conditions of the test set. As can be observed, models trained with our method mostly better generalize to unknown data distributions. In particular, our method always outperforms the baseline by a statistically significant margin when the training images are from *Night* scenarios. This is since the baseline models trained on images from *Night* are strongly biased towards dark scenery, while, as a consequence of training over worst-case distributions, our models can overcome this strong bias and better generalize across different unseen domains.

5 Conclusions and Future Work

We study a new adversarial data augmentation procedure that learns to better generalize across unseen data distributions, and define an ensemble method to exploit this technique in a classification framework. This is in contrast to domain adaptation algorithms, which require a sufficient number of samples from a known, a priori fixed target distribution. Our experimental results show that our iterative procedure provides broad generalization behavior on digit recognition and cross-season and cross-weather semantic segmentation tasks.

For future work, we hope to extend the ensemble methods by defining novel decision rules. The proposed heuristics (9) only apply to classification settings, and extending them to a broad realm of tasks including semantic segmentation is an important direction. Many theoretical questions still remain. For instance, quantifying the behavior of data-dependent regularization schemes presented in Section 3 would help us better understand adversarial training methods in general.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 137–144. MIT Press, 2007.

[2] Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *arXiv:1604.01446 [math.PR]*, 2016.

[3] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[4] J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] J. S. Denker, W. R. Gardner, H. P. Graf, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, H. S. Baird, and I. Guyon. Advances in neural information processing systems 1. chapter Neural Network Recognizer for Hand-written Zip Code Digits, pages 323–331. 1989.

[7] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[9] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1180–1189, 2015.

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

[13] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[14] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

[15] Hal Daumé III and Daniel Marcu. Domain adaptation for statistical classifiers. *CoRR*, abs/1109.6341, 2011.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[18] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[20] Jaeho Lee and Maxim Raginsky. Minimax statistical learning and domain adaptation with wasserstein distances. *arXiv preprint arXiv:1705.07815*, 2017.

[21] K Levenberg. A method for the solution of certain problems in least squares. quart. appl. math. 2. 1944.

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. *CoRR*, abs/1710.03077, 2017.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[24] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci. Robust place categorization with deep domain generalization. *IEEE Robotics and Automation Letters*, 3(3):2093–2100, July 2018.

[25] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[26] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *International Conference on Learning Representations*, 2018.

[27] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[28] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 10–18, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[29] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[31] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[32] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 213–226, Berlin, Heidelberg, 2010. Springer-Verlag.

[33] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018.

[34] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1), January 2014.

[36] Baochen Sun and Kate Saenko. Deep CORAL: correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

[37] Joshua Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *CoRR*, abs/1703.06907, 2017.

[38] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1521–1528, Washington, DC, USA, 2011. IEEE Computer Society.

[39] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[40] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

A Proofs

A.1 Proof of Theorem 1

Recall that we consider a fixed $\theta \in \Theta$, $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$, and $z_0 = g(\theta_f; x_0)$. We begin by noting that since $\mathrm{Im}(g(\theta_f; \cdot)) = \mathbb{R}^p$, we have

$$\begin{aligned}\phi_\gamma(\theta; (x_0, y_0)) &= \sup_{x \in \mathcal{X}} \{\ell(\theta; (x, y_0)) - \gamma c_\theta((x, y_0), (x_0, y_0))\} \\ &= \sup_{z \in \mathbb{R}^p} \left\{\ell(\theta; (z, y_0)) - \frac{\gamma}{2} \|z - z_0\|_2^2 =: h(z)\right\}. \end{aligned} \tag{12}$$

Similarly as $x_\epsilon^\star$, let $z_\epsilon^\star$ be an ϵ-optimizer to the problem (12)

$$z_\epsilon^\star \in \epsilon\text{-}\arg\max_{z \in \mathbb{R}^p} \left\{\ell(\theta; (z, y_0)) - \frac{\gamma}{2} \|z - z_0\|_2^2\right\}.$$

To further ease notation, let us denote

$$\ell_1(\theta; (z, y_0)) := \ell(\theta; (z_0, y_0)) + \nabla_z \ell(\theta; (z_0, y_0))^\top (z - z_0)$$

$$\ell_2(\theta; (z, y_0)) := \ell(\theta; (z_0, y_0)) + \nabla_z \ell(\theta; (z_0, y_0))^\top (z - z_0) + \frac{1}{2}(z - z_0)^\top \nabla_{zz} \ell(\theta; (z_0, y_0))(z - z_0),$$

the first- and second-order approximation of $z \mapsto \ell(\theta; (z, y_0))$ around $z = z_0$ respectively.

First, we note that $\|\nabla_{zz} \ell(\theta; (z, y))\| \leq L_1 < \gamma$ by hypothesis and hence, $\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)$ attains the maximum in the problem

$$\begin{aligned}\widehat{g}_{\mathrm{newton}}(\theta_f; x_0) &= z_0 + \frac{1}{\gamma}\left(I - \frac{1}{\gamma}\nabla_{zz}\ell(\theta; (z_0, y_0))\right)^{-1} \nabla_z \ell(\theta; (z_0, y_0)) \\ &= \arg\max_{z \in \mathbb{R}^p} \left\{\ell_2(\theta; (z, y_0)) - \frac{\gamma}{2} \|z - z_0\|_2^2 := h_2(z)\right\}\end{aligned} \tag{13}$$

Now, note that $h_2(z) = \ell_2(\theta; (z, y_0)) - \frac{\gamma}{2} \|z - z_0\|_2^2$ is $(\gamma - L_1)$ - strongly concave since

$$\lambda_{\min}(-\nabla_{zz} h(z)) \geq \gamma - \lambda_{\max}(\nabla_{zz} \ell_2(\theta; (z, y_0))) \geq \gamma - L_1$$

by Assumption 1, where $\lambda_{\max}$ and $\lambda_{\min}$ denotes the maximum and minimum eigenvalue respectively. Recalling the definition of $h(z)$ given in Eq (12), we then have

$$\begin{aligned}\frac{\gamma - L_1}{2} \|z_\epsilon^\star - \widehat{g}_{\mathrm{newton}}(\theta_f; x_0)\|_2^2 &\leq h_2(z_\epsilon^\star) - h_2(\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)) \\ &= h(z_\epsilon^\star) - h(\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)) + h_2(z_\epsilon^\star) - h(z_\epsilon^\star) \\ &\quad + h(\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)) - h_2(\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)) \\ &\leq \epsilon + h_2(z_\epsilon^\star) - h(z_\epsilon^\star) \\ &\quad + h(\widehat{g}_{\mathrm{newton}}(\theta_f; x_0)) - h_2(\widehat{g}_{\mathrm{newton}}(\theta_f; x_0))\end{aligned} \tag{14}$$

where we used the definition of $z_\epsilon^\star$ in the last inequality.

Next, we note that h_2 and h are close by Taylor expansion.

Lemma 2 ([29, Lemma 1]). *Let $f : \mathbb{R}^p \to \mathbb{R}$ have a L-Lipschitz Hessian so that for all $z, z' \in \mathbb{R}^p$, $\|\nabla_{zz} f(z) - \nabla_{zz} f(z')\| \leq L \|z - z'\|_2$. Then, for all $z, z' \in \mathbb{R}^p$,*

$$\left|f(z') - f(z) - \nabla f(z)^\top (z' - z) - \frac{1}{2}(z' - z)^\top \nabla_{zz} f(z)(z' - z)\right| \leq \frac{L}{6} \|z' - z\|_2^2.$$

Applying Lemma 2, we have that

$$|h_2(z) - h(z)| \leq \frac{L_2}{6} \|z - z_0\|_2^3.$$

Using this inequality in the bound (14), we arrive at

$$\begin{aligned}&\frac{\gamma - L_1}{2} \|z_\epsilon^\star - \widehat{g}_{\mathrm{newton}}(\theta_f; x_0)\|_2^2 \\ &\leq \epsilon + \frac{L_2}{6}\left(\|z_0 - z_\epsilon^\star\|_2^3 + \|z_0 - \widehat{g}_{\mathrm{newton}}(\theta_f; x_0)\|_2^3\right)\end{aligned} \tag{15}$$

From definition (13) of $\widehat{g}_{\text{newton}}(\theta_f; x_0)$, we have

$$\|z_0 - \widehat{g}_{\text{newton}}(\theta_f; x_0)\|_2^3 \leq \left(\frac{1}{\gamma}\right)^3 \left(\frac{\gamma}{\gamma - L_1}\right)^3 L_0^3. \tag{16}$$

Next, to bound $\|z_0 - z_\epsilon^\star\|_2$ in the bound (15), we show that $z_\epsilon^\star$ and z_0 are at most $O(1/\gamma)$-away. We defer the proof of the following lemma to Appendix A.2

Lemma 3. *Let Assumption 1 hold and Im$(g(\theta_f; \cdot)) = \mathbb{R}^p$. Then,*

$$\left\| z_\epsilon^\star - z_0 - \frac{1}{\gamma}\nabla_z \ell(\theta; (z_0, y_0)) \right\|_2 \leq \frac{4L_0}{\gamma} + \sqrt{\frac{2\epsilon}{\gamma}}.$$

Applying Lemma 3 to bound $\|z_0 - z_\epsilon^\star\|_2^3$ on the right hand side of inequality (15), and using the bound (16) for $\|z_0 - \widehat{g}_{\text{newton}}(\theta_f; x_0)\|_2^3$, we obtain

$$\frac{\gamma - L_1}{2} \|z_\epsilon^\star - \widehat{g}_{\text{newton}}(\theta_f; x_0)\|_2^2 \leq \epsilon + \frac{L_2}{6}\left[\left(\frac{5L_0}{\gamma}\right)^3 + \left(\frac{2\epsilon}{\gamma}\right)^{\frac{3}{2}} + \left(\frac{L_0}{\gamma - L_1}\right)^3\right].$$

This gives the final result.

A.2 Proof of Lemma 3

We use the following key lemma which says that for functions that satisfy a growth condition, its minimum is stable to perturbations to the function.

Lemma 4 ([4, Proposition 4.32])**.** *Suppose that f_0 satisfies the second-order growth condition: there exists a $c > 0$ such that if we denote by $z^\star$ the minimizer of f so that $f_0(z^\star) = \inf_{z \in \mathbb{R}^p} f_0(z)$, we have for all z*

$$f_0(z) \geq f_0(z^\star) + c\|z - z^\star\|_2^2.$$

If there is a function $f_1 : \mathbb{R}^p \to \mathbb{R}$ such that $f_0 - f_1$ is κ-Lipschitz on a neighborhood N of $x^\star$, then z, any ϵ-approximate minimizer of f_1 in N, satisfies

$$\|z - z^\star\|_2 \leq c^{-1}\kappa + c^{-1/2}\epsilon^{1/2}$$

Letting $f_0(z) := -\ell_1(\theta; (z, y_0)) + \frac{\gamma}{2}\|z - z_0\|_2^2$ and $f_1(z) := -h(z) = -\ell(\theta; (z, y_0)) + \frac{\gamma}{2}\|z - z_0\|_2^2$, note first that f_0 is γ-strongly convex. Further, $f_0(z) - f_1(z) = \ell(\theta; (z, y_0)) - \ell_1(\theta; (z, y_0))$ is $2L_0$-Lipschitz by Assumption 1. Applying Lemma 4, we obtain the result.

A.3 Proof of Theorem 2

Again, we abuse notation by writing $\ell(\theta; (z, y)) = \ell(\theta; (x, y))$ for $z = g(\theta_f; x) \in \mathbb{R}^p$, and similarly $p_j(\theta; z)$ and $\phi_\gamma(\theta; z)$. We begin by noting that since $\text{Im}(g(\theta, \cdot)) = \mathbb{R}^p$, we have

$$\phi_\gamma(\theta; (x, y)) = \sup_{z' \in \mathbb{R}^p} \left\{\ell(\theta; (z', y)) - \frac{\gamma}{2}\|z - z'\|_2^2\right\}.$$

The following claim will be crucial.

Claim 5. *If $z \mapsto \nabla_z \ell(\theta; (z, y))$ is L-Lipschitz with respect to the $\|\cdot\|_2$-norm, then*

$$\frac{1}{\gamma + L}\|\nabla_z \ell(\theta; (z, y))\|_2^2 \leq \phi_\gamma(\theta; (z, y)) - \ell(\theta; (z, y)) \leq \frac{1}{\gamma - L}\|\nabla_z \ell(\theta; (z, y))\|_2^2.$$

Proof of Claim From Taylor's theorem, we have

$$\left|\ell(\theta; (z', y)) - \ell(\theta; (z, y)) - \nabla_z \ell(\theta; (z, y))^\top (z' - z)\right| \leq \frac{1}{2}L\|z - z'\|_2^2.$$

Using this approximation in the definition of $\phi_\gamma(\theta; (z, y))$, we get

$$\begin{aligned}\phi_\gamma(\theta; (z, y)) &\leq \sup_{z'} \left\{\ell(\theta; (z, y)) + \nabla_z \ell(\theta; (z, y))^\top (z' - z) - \frac{\gamma - L}{2}\|z - z'\|_2^2\right\} \\ &= \ell(\theta; (z, y)) + \frac{1}{2(\gamma - L)}\|\nabla_z \ell(\theta; (z, y))\|_2^2.\end{aligned}$$

Similarly, we can compute the lower bound

$$\begin{aligned}\phi_\gamma(\theta;(z,y)) &\geq \sup_{z'}\left\{\ell(\theta;(z,y)) + \nabla_z\ell(\theta;(z,y))^\top(z-z') - \frac{\gamma+L}{2}\left\|z-z'\right\|_2^2\right\}\\ &= \ell(\theta;(z,y)) + \frac{1}{2(\gamma+L)}\left\|\nabla_z\ell(\theta;(z,y))\right\|_2^2.\end{aligned}$$

Combining the two bounds, the claim follows. □

From the claim, it suffices to show that $z \mapsto \nabla_z\ell(\theta;(z,y))$ is L-Lipschitz. From $\nabla_z\ell(\theta;(z,y)) = -\theta_{c,y} + \sum_{j=1}^m p_j(\theta;z)\theta_{c,j}$, we have

$$\left\|\nabla_z\ell(\theta;(z',y)) - \nabla_z\ell(\theta;(z,y))\right\|_2 = \left\|\sum_{j=1}^m (p_j(\theta;z) - p_j(\theta;z'))\theta_j\right\|_2.$$

Now, since

$$\left\|\nabla_z p_j(\theta;z)\right\|_2 = \left\|-p_j(\theta;z)\left(\theta_j - \sum_{l=1}^m p_l(\theta;z)\theta_l\right)\right\|_2 \leq 2\max_{1\leq j\leq m}\left\|\theta_{c,j}\right\|_2,$$

we conclude that

$$\left\|\nabla_z\ell(\theta;(z',y)) - \nabla_z\ell(\theta;(z,y))\right\|_2 \leq L(\theta)\left\|z-z'\right\|_2.$$

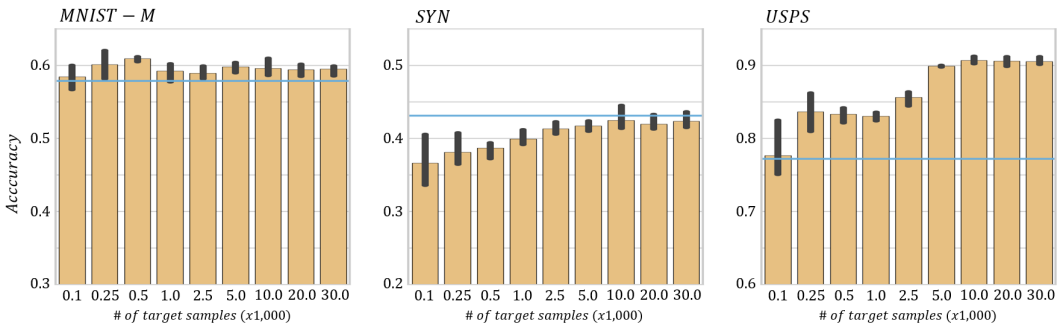

Figure 3. Results obtained by running ADDA algorithm [39] using $10,000$ labeled MNIST samples and a number of target samples indicated on the x-axis. The *blue* lines indicate results obtained with our method with $K = 2$ and $\gamma = 1.0$. Test sets are MNIST-M *(left)*, SYN *(middle)* and USPS *(right)*.

B Additional Experimental Results

Table 1 reports results associated with the digit experiment (Section 4.1, Figure 2). In particular, it reports numerical results (averaged over 10 different runs) obtained with models trained with Algorithm 1 by varying the hyperparameters K and γ. Training set is constituted by $10,000$ MNIST samples, models were tested on SVHN, MNIST-M, SYN and USPS (see Figure 1 *(top)*). The baselines (accuracies achieved by models trained with ERM) are:

- SVHN: 0.283 ± 0.032
- MNIST-M: 0.548 ± 0.021
- SYN: 0.406 ± 0.022
- USPS: 0.789 ± 0.017

Table 2 reports results associated with the semantic segmentation experiment (Section 4.2, Figure 3). To summarize, it reports results obtained by training models on *Highway* and testing them on *New York-like City* and *Old European Town*, and by training models on *New York-like City* and testing them on *Highway* and *Old European Town* (see Figure 1 *(bottom)* to observe the different weather/time/date conditions). The comparison is between models trained with ERM *(ERM rows)* and our method *(Ours rows)*, *e.g.*Algorithm 1 with $K = 1$ and $\gamma = 1.0$.

Finally, Figure 4 reports a comparison between our method *(blue)* and the unsupervised domain adaptation algorithm ADDA [39] *(yellow)*, by varying the number of target images fed to the latter during training. Note that, since unsupervised domain adaptation algorithms make use of target data during training while our method does not, the comparison is not fair. However, we are interested in evaluating to which extent our method can compete with a well performing unsupervised domain adaptation algorithm [39]. While on MNIST $\rightarrow$ USPS split ADDA clearly outperforms our method, on MNIST $\rightarrow$ MNIST-M the accuracies reached by our method are just slightly lower than the ones reached by ADDA, and on MNIST $\rightarrow$ SYN our method outperforms it, even if the domain adaptation algorithm has access to a large number of samples from the target domain. Finally, note that MNIST $\rightarrow$ SVHN results are not provided because ADDA would not converge on this split (in effect, these results are neither reported in the original work [39]). Instead, models trained on MNIST samples using our method better generalize to SVHN, as shown in Section 4.1.

Table 1. Results obtained by training models with Algorithm 1 on $10,000$ MNIST samples and testing them on SVHN, MNIST-M, SYN and USPS. Results are averaged over 20 different runs.

	K=1	K=2	K=3	K=4
SVHN				
$\gamma = 10^{-6}$	0.287 ± 0.006	0.327 ± 0.016	0.334 ± 0.031	0.328 ± 0.033
$\gamma = 10^{-5}$	0.284 ± 0.036	0.311 ± 0.033	0.316 ± 0.036	0.331 ± 0.026
$\gamma = 10^{-4}$	0.331 ± 0.018	0.324 ± 0.026	0.336 ± 0.020	0.325 ± 0.030
$\gamma = 10^{-3}$	0.294 ± 0.023	0.316 ± 0.029	0.309 ± 0.024	0.343 ± 0.017
$\gamma = 10^{-2}$	0.290 ± 0.041	0.320 ± 0.030	0.341 ± 0.030	0.346 ± 0.033
$\gamma = 10^{-1}$	0.284 ± 0.007	0.324 ± 0.017	0.307 ± 0.026	0.323 ± 0.029
$\gamma = 10^{0}$	0.284 ± 0.012	0.306 ± 0.008	0.314 ± 0.022	0.335 ± 0.029
$\gamma = 10^{1}$	0.305 ± 0.031	0.301 ± 0.035	0.316 ± 0.027	0.343 ± 0.030
$\gamma = 10^{2}$	0.304 ± 0.032	0.300 ± 0.017	0.327 ± 0.026	0.321 ± 0.034
$\gamma = 10^{3}$	0.289 ± 0.030	0.314 ± 0.032	0.300 ± 0.017	0.304 ± 0.025
$\gamma = 10^{4}$	0.300 ± 0.020	0.299 ± 0.028	0.325 ± 0.015	0.340 ± 0.026
MNIST-M				
$\gamma = 10^{-6}$	0.561 ± 0.013	0.584 ± 0.008	0.581 ± 0.009	0.588 ± 0.013
$\gamma = 10^{-5}$	0.564 ± 0.024	0.573 ± 0.010	0.573 ± 0.024	0.589 ± 0.017
$\gamma = 10^{-4}$	0.583 ± 0.011	0.572 ± 0.010	0.586 ± 0.015	0.578 ± 0.031
$\gamma = 10^{-3}$	0.562 ± 0.026	0.579 ± 0.010	0.567 ± 0.023	0.601 ± 0.018
$\gamma = 10^{-2}$	0.539 ± 0.037	0.578 ± 0.013	0.590 ± 0.014	0.598 ± 0.014
$\gamma = 10^{-1}$	0.556 ± 0.017	0.589 ± 0.021	0.576 ± 0.018	0.576 ± 0.019
$\gamma = 10^{0}$	0.557 ± 0.017	0.579 ± 0.009	0.571 ± 0.010	0.584 ± 0.024
$\gamma = 10^{1}$	0.568 ± 0.022	0.564 ± 0.028	0.579 ± 0.024	0.589 ± 0.016
$\gamma = 10^{2}$	0.564 ± 0.025	0.569 ± 0.013	0.579 ± 0.019	0.578 ± 0.021
$\gamma = 10^{3}$	0.558 ± 0.016	0.568 ± 0.017	0.568 ± 0.010	0.567 ± 0.021
$\gamma = 10^{4}$	0.567 ± 0.022	0.561 ± 0.023	0.570 ± 0.015	0.579 ± 0.016
SYN				
$\gamma = 10^{-6}$	0.415 ± 0.013	0.445 ± 0.007	0.440 ± 0.012	0.443 ± 0.013
$\gamma = 10^{-5}$	0.409 ± 0.029	0.432 ± 0.020	0.437 ± 0.024	0.443 ± 0.014
$\gamma = 10^{-4}$	0.439 ± 0.011	0.437 ± 0.011	0.446 ± 0.018	0.440 ± 0.022
$\gamma = 10^{-3}$	0.417 ± 0.018	0.437 ± 0.021	0.436 ± 0.017	0.450 ± 0.010
$\gamma = 10^{-2}$	0.417 ± 0.022	0.439 ± 0.015	0.447 ± 0.020	0.450 ± 0.014
$\gamma = 10^{-1}$	0.405 ± 0.011	0.439 ± 0.009	0.438 ± 0.018	0.439 ± 0.021
$\gamma = 10^{0}$	0.418 ± 0.004	0.431 ± 0.017	0.426 ± 0.021	0.441 ± 0.013
$\gamma = 10^{1}$	0.421 ± 0.016	0.427 ± 0.020	0.436 ± 0.020	0.445 ± 0.016
$\gamma = 10^{2}$	0.427 ± 0.017	0.427 ± 0.016	0.436 ± 0.021	0.432 ± 0.014
$\gamma = 10^{3}$	0.410 ± 0.027	0.424 ± 0.019	0.422 ± 0.019	0.418 ± 0.015
$\gamma = 10^{4}$	0.422 ± 0.018	0.423 ± 0.015	0.441 ± 0.010	0.443 ± 0.016
USPS				
$\gamma = 10^{-6}$	0.778 ± 0.019	0.783 ± 0.016	0.784 ± 0.012	0.784 ± 0.012
$\gamma = 10^{-5}$	0.775 ± 0.016	0.774 ± 0.017	0.778 ± 0.010	0.782 ± 0.016
$\gamma = 10^{-4}$	0.781 ± 0.010	0.760 ± 0.021	0.772 ± 0.013	0.774 ± 0.021
$\gamma = 10^{-3}$	0.758 ± 0.012	0.788 ± 0.014	0.771 ± 0.011	0.784 ± 0.011
$\gamma = 10^{-2}$	0.765 ± 0.012	0.775 ± 0.024	0.772 ± 0.021	0.775 ± 0.011
$\gamma = 10^{-1}$	0.773 ± 0.011	0.787 ± 0.013	0.774 ± 0.011	0.776 ± 0.018
$\gamma = 10^{0}$	0.778 ± 0.007	0.772 ± 0.010	0.774 ± 0.017	0.768 ± 0.021
$\gamma = 10^{1}$	0.767 ± 0.018	0.774 ± 0.013	0.779 ± 0.016	0.773 ± 0.014
$\gamma = 10^{2}$	0.774 ± 0.014	0.782 ± 0.013	0.776 ± 0.018	0.771 ± 0.021
$\gamma = 10^{3}$	0.774 ± 0.013	0.774 ± 0.017	0.775 ± 0.012	0.763 ± 0.025
$\gamma = 10^{4}$	0.778 ± 0.013	0.773 ± 0.012	0.774 ± 0.012	0.781 ± 0.011

Table 2. Results *(mIoUs)* associated with the experiments on SYNTHIA dataset. The *first* column indicate the training set. The *second* column indicate the method used: Empirical Risk Minimization *(ERM)* and our method *(Ours)* with $K = 1$ and $\gamma = 1.0$. Remaining columns indicate the test set.

		New York-like City					Old European Town				
		Dawn	Fog	Night	Spring	Winter	Dawn	Fog	Night	Spring	Winter
Highway/Dawn	*ERM*	18.9	14.7	10.7	14.5	13.4	22.0	20.8	14.5	18.6	15.3
	Ours	**24.0**	**17.0**	**19.1**	**22.9**	**20.2**	**27.6**	**25.0**	**22.4**	**27.1**	**19.0**
Highway/Fog	*ERM*	12.6	27.8	9.0	12.9	13.4	13.6	20.7	12.1	15.1	12.7
	Ours	**17.4**	**28.4**	**11.0**	**18.4**	18.4	**18.5**	**27.5**	**16.4**	**22.0**	**19.0**
Highway/Night	*ERM*	13.0	7.7	13.9	13.2	10.9	16.6	11.5	19.0	15.7	9.9
	Ours	**18.5**	**14.5**	24.8	**22.9**	**22.0**	**22.2**	**20.1**	**28.1**	**25.5**	**19.1**
Highway/Spring	*ERM*	15.2	16.0	10.8	15.8	14.8	18.8	21.2	14.7	19.2	13.9
	Ours	**22.6**	**19.4**	14.6	**25.5**	**23.5**	**25.1**	**26.5**	**21.5**	**29.9**	**24.5**
Highway/Winter	*ERM*	14.1	15.9	11.7	14.8	16.8	15.2	19.3	14.6	16.9	20.0
	Ours	**16.9**	**17.4**	**12.5**	**21.0**	**24.0**	**17.0**	**20.5**	**14.9**	**23.1**	**26.8**
		Highway					Old European Town				
		Dawn	Fog	Night	Spring	Winter	Dawn	Fog	Night	Spring	Winter
NY.Like C./ Dawn	*ERM*	19.6	19.1	13.1	18.8	15.9	27.9	23.5	16.3	21.7	17.0
	Ours	**22.8**	**22.8**	**17.8**	**21.4**	**18.5**	**31.0**	**25.9**	**22.4**	**26.0**	22.3
NY.Like C./Fog	*ERM*	12.5	15.9	9.1	11.8	10.7	24.2	26.5	17.8	21.7	16.0
	Ours	**15.4**	**23.1**	**16.3**	**18.7**	**18.2**	**17.3**	**26.4**	**17.5**	**24.3**	21.6
NY.Like C./Night	*ERM*	14.9	14.7	16.3	13.5	13.1	25.4	24.7	24.4	23.3	17.0
	Ours	**19.4**	**20.2**	**22.1**	**19.7**	**17.3**	**23.3**	**23.9**	**27.2**	**27.2**	**22.1**
NY.Like C./Spring	*ERM*	**17.1**	**18.0**	**12.8**	**16.3**	**14.8**	**26.6**	**27.0**	**20.4**	**26.3**	**22.5**
	Ours	14.5	14.7	11.8	15.2	11.2	21.9	21.9	19.7	24.8	22.9
NY.Like C./Winter	*ERM*	16.1	17.3	11.9	16.5	16.0	**21.3**	**23.8**	19.4	24.1	23.2
	Ours	**18.1**	**18.2**	**15.2**	**17.8**	**17.3**	21.0	21.0	**19.9**	**25.5**	**25.6**