ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BIWI Computer Vision

# Distribution Aligned Semi-Supervised Learning

Master's Thesis

Qin Wang
D-ITET

**A**dvisor: Dr. Wen Li, Eirikur Agustsson
**S**upervisor: Prof. Dr. Luc van Gool

August 30, 2018

# Abstract

In this work, we propose a novel approach to improve the deep networks under the semi-supervised learning scenario. Existing semi-supervised learning methods usually employ unlabeled data to form a regularization loss term for improving the robustness of learning. Such a term usually relies on a model trained from labeled data, and applied to unlabeled data. While labeled and unlabeled data are often assumed to be sampled from the same distribution, we reveal that the sampling bias exists and can lead to a considerable distribution mismatch between labeled and unlabeled data. To this end, we propose a new method to address this issue by aligning the two distributions by using adversarial training. However, effectiveness of the alignment is limited by insufficient distribution support due to limited number of labeled samples. To tackle this problem, we further improve the alignment by augmenting distribution supports, which is achieved by interpolating between labeled and unlabeled data. We also show that our proposed approach could also be used for improving the performance of domain adaptation, where the labeled and unlabeled data are from different domains. Effectiveness of the proposed method is demonstrated on multiple benchmark datasets: 1) SVHN and Cifar-10 for the semi-supervised learning tasks, and 2) MNIST to MNIST-M and Office-31 for domain adaptation tasks. 3) Webvision-Tiny for domain-aware supervised learning task.

# Acknowledgements

I would like to thank my supervisors Dr. Wen Li, Eirikur Agustsson, and Prof. Dr. Luc Van Gool for their guidance and help. I would like to specially thank Dr. Wen Li for his inspiring suggestions and helpful insights. I would also like to thank Prof. Limin Wang for his advice.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Current deep learning models for computer vision usually rely on large numbers of labeled training samples. However, for many practical problems and scenarios, collecting a large labeled dataset is usually infeasible due to various reasons. For example, the need for domain-specific knowledge, the lack of financial support, or sometimes simply insufficient resources. One possible approach to tackle this issue is semi-supervised learning, where performance is expected to be improved by learning from both labeled and unlabeled data, instead of the small amount of labeled data alone. Fortunately, in many cases, it is easier to obtain a large number of unlabeled data. Many studies have shown that in certain cases, these additional unlabeled data can improve the model performance[19].

In order to make use of the unlabeled data, most recent semi-supervised learning methods add an additional regularization term to the training of neural networks. From an Bayesian point of view, these regularization terms can be interpreted as priori knowledge. For example, entropy minimum methods [7] utilize the prori of *predictor should make confident and low-entropy predictions for unlabeled images*. Similarly, consistency regularization [23] [13] utilizes the assumption *predictor should be robust against random and local perturbation*, and is one of the most popular prior in recent years. This simple but yet effective principle has yielded a number of successful regularization term for semi-supervised learning problems, including the state-of-the-art methods. These regularization terms usually assume that labeled and unlabeled data are of the same distribution, and thus train models from labeled data, and apply them to unlabeled data in the regularization term.

However, we reveal a weakness of these methods, namely the distribution mismatch between labeled and unlabeled data. As we mentioned before, in most semi-supervised learning scenarios, only a small number of labeled data are available. The labeled samples are thus most likely to be insufficient to represent the true data distribution. This sampling bias further leads to a distribution mismatch between labeled and unlabeled data, and therefore contradicts the assumption existing regularization methods made: *labeled and unlabeled data are of the same distribution*.

Our proposed semi-supervised learning method alleviates this issue by simply aligning the two distributions. In order to quantify this idea, we utilize the idea of adversarial training and force the neural network to produce *domain-invariant* features, i.e. it is hard for the discriminator to tell whether a feature is from labeled/unlabeled data. The alignment is further improved by augmenting distribution support. This is achieved by interpolating between labeled and unlabeled data.

The proposed method has the following advantages. First, the method is applicable to any problem where two or more sets of data are involved, which means that the method is applicable not only to semi-supervised learning tasks, but also other scenarios such as domain adaptation or multi-domain supervised learning. Second, only a small number of hyperparameters are needed. In fact, only one additional hyperparameter is introduced in the proposed method. Unlike most semi-supervised learning methods that require a heavy

tuning on the ramp-up weight function to balance the trade-off between labeled loss and unlabeled loss, the hyperparameter is usually acceptable even with a default value for different datasets.

The proposed model is trained and evaluated under three different scenarios: semi-supervised learning, domain adaptation, and domain-aware supervised learning. When we applied our methods to semi-supervised learning tasks such as Cifar-10 and SVHN, our method demonstrated better or comparable performance. The method also achieved state-of-the-art result on domain adaptation tasks such as Office-31. Furthermore, we evaluated our interpolation method in a supervised learning setting, and demonstrated that augmenting distribution support by interpolating between two distributions can also improve generalization performance for supervised learning in certain cases.

## 1.1 Focus of this Work

The focus of this work is as follows:

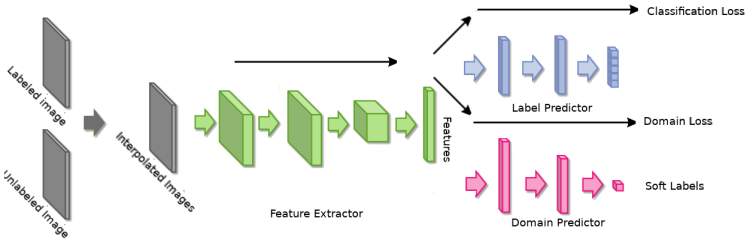
- We consider the scenario where only a small amount of labeled data and a large amount of unlabeled data are available for training. Specifically, we consider the image classification problem for semi-supervised learning tasks and domain adaptation tasks.
- We reveal that sampling bias exists in this case and can lead to a considerable distribution mismatch between labeled and unlabeled data. This distribution mismatch is largely overlooked by existing semi-supervised learning methods.
- We show that distribution alignment via adversarial training can alleviate the distribution mismatch, and improve the generalization performance.
- We further show that interpolating between labeled and unlabeled data can augment distribution support, and help the adversarial alignment. We observe a considerable improvement on image classification.
- Our proposed method only requires minor changes in the original classification networks. Namely a domain predictor as discriminator, and some multiplyaccumulate operations for interpolation.

## 1.2 Thesis Organization

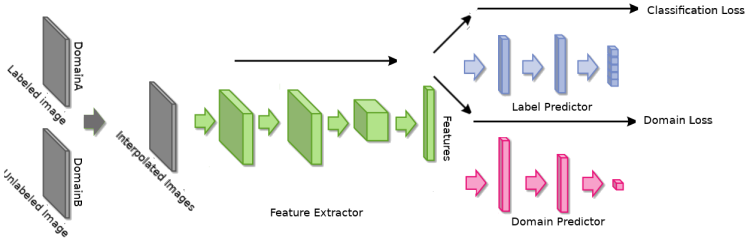
This thesis is organized as follows. We first present an overview of related methods in Chapter 2. Then we formally define the problem and propose our method in Chapter 3. To prove the effectiveness of the proposed method, we present the experiment details on three different tasks, including semi-supervised learning, domain adaptation, and domain-aware supervised learning in Chapter 4. In the end, we discuss our findings and possible future directions in Chapter 5.

## 1.3 Settings Overview

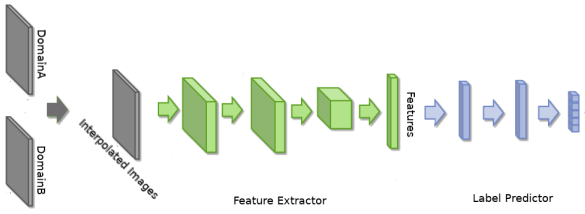
As our work contains multiple experiments with different setups, we give an overview of the pipelines we used for different tasks in Fig 1.3.




(a) Pipeline for semi-supervised learning tasks. Interpolation is done between labeled and unlabeled data. Adversarial alignment is implemented to address the distribution mismatch problem.




(b) Pipeline for Domain-adaptation tasks. Interpolation is done between two domains. Adversarial alignment is implemented to address the distribution mismatch problem.




(c) Pipeline for semi-supervised learning tasks. Interpolation is done between two domains. No Adversarial alignment is involved.

Figure 1.1: Overview of settings discussed in this thesis.

# Chapter 2

# Related Work

Most recent semi-supervised learning methods are based on the following intuitive idea: a small perturbation $\epsilon$ added to unlabeled data $x_u \in \mathcal{D}_{ul}$ should not change the output of the model $f$. The idea is sometimes referenced as *consistency regularization*, and is often achieved by minimizing the distance between outputs of the original and perturbed data $d(f(x_u), f(x_u + \epsilon))$. A number of papers have proposed a series of successful methods based on this idea. These methods share the same idea but varies on the choice of perturbation $\epsilon$ and distance function. One simple approach is to introduce perturbation from common regularization methods such as additive noise, dropout and data augmentation. This approach results in the popular $\Pi$ Model [13], where stochastic perturbations are built in through regularization methods, and mean square error is chosen as the distance function. The distance term is added to the original loss function and serves as a regularization term.

One significant problem in the $\Pi$ Model is that it utilizes a potentially unstable prediction $f(x_u + \epsilon)$, which can rapidly change over the course of training [19]. Several methods are proposed to solve this potential issue. For example, [13] proposes to use an averaged prediction using the outputs of the network-in-training on different epochs. Similarly, [23] instead proposes to use accumulated parameters of $f$ to construct a different $\hat{f}$ for prediction. While all these methods are proposed to solve the potentially unstable prediction problem, we argue that they fail to notice the reason behind this problem. We will show in the next chapter, the distribution of labeled and unlabeled data can be different due to sampling bias, though we usually assume they are the same. This overlooked distribution mismatch and the fact that we are using a model trained from one distribution and applying it to a different distribution are one of the reasons behind the unstable prediction problem.

Aligning the distributions is a straightforward idea to address the distribution mismatch problem. Although rarely mentioned in semi-supervised learning, distribution alignment is a widely discussed topic in the area of domain adaptation, where data from two different domains are involved [3] [4] [22]. [4] proposes to use a domain predictor as discriminator to tell the domain of features, while the feature extractor try to generate features that can fool the discriminator. The network thus generates domain-invariant features and aligns the two distributions. This method is closely related to [6], and shares similar idea of adversarial training.

Another area of work that are worth mentioning is the family of interpolation-based regularization methods. These methods are proposed as augmentation approaches that are able to improve generalization performance without any modification on the original networks. For example, *Mixup* method [25] trains neural networks using convex combinations of pairs examples and their labels, and uses a beta distribution to control the intensity of the interpolation. At the same time, [10] proposes similar ideas to use convex combinations, but uses 0.5 as the fixed weight for interpolation. These methods have been proved to improve generalization performance through extensive experiments. [25]

# Chapter 3

# Methods

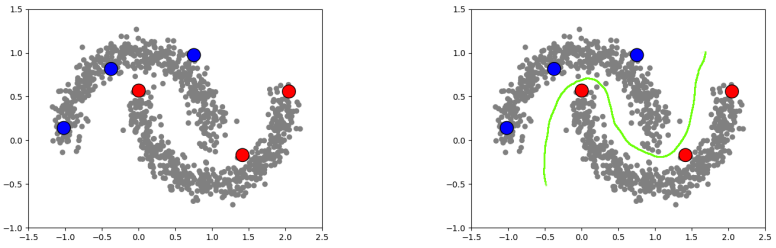
In this chapter, we present our method for semi-supervise learning problems. We start from a formal problem definition with a brief recap of existing method. We then show that distribution mismatch between labeled and unlabeled data is one of the potential issues behind existing methods. We finally propose our method to align the distributions.

## 3.1 Problem Description

In this work, we mainly consider the following semi-supervised classification problem.
Given:

- A small set of labeled training data $\mathcal{D}_l = \{(x_{l1}, y_{l1}), ...(x_{ln}, y_{ln})\}$
- A large set of unlabeled training data $\mathcal{D}_{ul} = \{(x_{u1}, y_{u1}), ...(x_{um}, y_{um})\}$, where $y_{ui}$ is unknown.
- Test data $\mathcal{D}_t = \{(x_{t1}, y_{t1}), ...(x_{tk}, y_{tk})\}$, where $y_{ti}$ is unknown.
- A neural network $f := f_l(f_e(\cdot))$ parameterized by $\theta$ including two parts $\theta_e$ and $\theta_l$. $f_e$ is the feature extractor parameterized by $\theta_e$ that maps the input $x$ to a feature space. $f_l$ is the label predictor parameterized by $\theta_l$ that makes predictions based on features.

The model is asked to learn from both labeled and unlabeled data and improve the model's performance on test data.



(a) Labeled data: blue and red; Unlabeled data: gray

(b) Expected classifier (green line as decision boundary)

Figure 3.1: A toy semi-supervised learning example: two-moon classification.

## 3.2 Backgrounds

### 3.2.1 General Form of Semi-Supervised Learning

Most existing semi-supervised learning methods have the following general form:

$$\mathcal{L} = \mathcal{L}_{labeled}(y, f(x_l)) + \lambda \mathcal{L}_{unlabeled}(f(x_u)), \tag{3.1}$$

where $\mathcal{L}_{labeled}$ is the supervised loss function, usually cross entropy, $\mathcal{L}_{unlabeled}$ is the unsupervised regularization loss function, and $\lambda$ is the weight.

In particular, the current state-of-the-art methods utilize the prior that realistic perturbations should not change the output of classification $f(x_u)$, and thus has the distance $d(f_\theta(x_u), f_\theta(\hat{x}_u))$, as the regularization term:

$$\mathcal{L} = CE(y, f(x_l)) + \lambda d(f(x_u), f(\hat{x}_u)), \tag{3.2}$$

where $\hat{x}_u$ is the perturbed data, $CE$ means cross entropy, and $d$ can be any function that measures difference, such as mean square error and cross entropy.

### 3.2.2 Distribution Mismatch

The general loss function (3.1) used by existing methods train the model from labeled data and apply it directly to unlabeled data. It is acceptable based on the assumption that labeled and unlabeled data are of the same distribution. Although the assumption holds to some extent, the fact that we have only a very limited amount of labeled data can strongly weaken the validity of this assumption.

In supervised learning, we usually minimize the empirical risk:

$$R(f) = \int \mathcal{L}(f(x), y) dP(x, y),$$

where $P(X, Y)$ is the sample distribution. In practice, we only have access to a limited number of training data $\mathcal{D} = (x_i, y_i)_{i=1}^n$, where $(x_i, y_i) \sim P$ and we don't know P in its analysis form. Thus an empirical distribution is usually used instead to approximate the sample distribution:

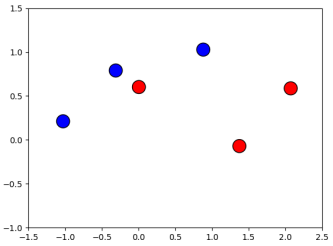
$$P_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \delta(x = x_i, y = y_i),$$

where $\delta$ is a Dirac measure centered at $(x_i, y_i)$. The empirical risk now can be estimated by:

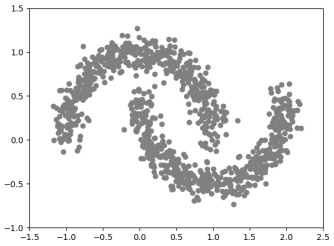
$$R(f) \approx \int \mathcal{L}(f(x), y) dP_\delta(x, y) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i)$$

This approximation relies on a large sample size, and thus raises a problem in semi-supervised learning, where the labeled sample size is extremely small. In this case, , the approximation deviates from the real sample distribution. Thus the model trained from this empirical distribution is unlikely to generalize well on the unlabeled data. We argue that this is one of the hidden reasons behind potentially unstable problem for consistency regularization methods described in [19].

We visualize this issue in a toy example as shown in Fig 3.2. Due to the small sample size, the six labeled points cannot represent the two moon data well, while the 1000 unlabeled data clearly better reflects the distribution. We quantify this distribution difference by conducting t-test between the six points and 1000 unlabeled points. We report the p-value as 93.45% for x-axis and 26.22% for y-axis. The same t-test is also conducted between six random Gaussian points and 1000 unlabeled points. In this settings, the



(a) Six labeled examples are not representative due to small sample size.



(b) 1000 unlabeled examples represents the two moon distribution well.

Figure 3.2: Distribution mismatch due to small sample size.

averaged p-value over ten runs is 93.15% for x-axis and 56.18% for y-axis. This clearly shows that the six labeled points are not more representative than random Gaussian points because of the small sample size, and the distribution mismatch can exist when sample size is small.

The distribution mismatch problem can even be worse in higher dimensions because the number of samples required to represent the distribution well is a lot larger.

The main contribution of our work is that we reveal this distribution mismatch in semi-supervised learning, which was overlooked in existing methods.

## 3.3 Proposed Method

To address this issue, we need to align the two distributions. This can be achieved via two components of our method.

### 3.3.1 Distribution Alignment

We now introduce our distribution alignment component. Distribution alignment is a widely discusses topic in domain adaptation. The basic idea used in our approach was originally proposed in [4], and we modify it for our semi-supervised learning scenario.

The alignment can be achieved by introducing a discriminator $\theta_d$ to tell whether the features come from labeled or unlabeled data and training the full network in an adversarial way. After introducing the discriminator, the neural network includes three component: a feature extractor, a label predictor, and a discriminator.

During learning stage, on one side, we are trying to achieve the traditional training objective that minimize the label prediction error. At the same time, we are also pushing the features to be invariant towards its origin, i.e. We want the distribution between $f_e(x_l)$ and $f_e(x_u)$ to be similar. This is monitored by the discriminator, where a successful alignment should always have high domain prediction loss. To avoid a trivial solution of a poor discriminator, we also need to train the discriminator separately in a supervised way.

This can be achieved by solving a min-max problem:

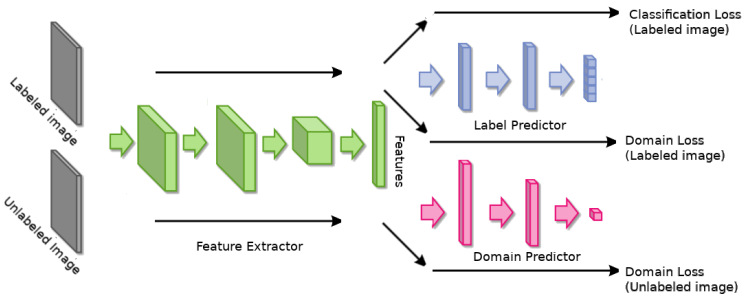



Figure 3.3: Adversarial distribution alignment.

$$\mathcal{L}(\theta_e, \theta_l, \theta_d) = \mathcal{L}_{labeled}(\theta_e, \theta_l) - \lambda \mathcal{L}_d(\theta_f, \theta_d) \tag{3.3}$$

$$(\hat{\theta}_e, \hat{\theta}_l) = \arg\min_{\theta_e, \theta_l} \mathcal{L}(\theta_e, \theta_l, \hat{\theta}_d) \tag{3.4}$$

$$\hat{\theta}_d = \arg\max_{\theta_d} \mathcal{L}(\hat{\theta}_e, \hat{\theta}_l, \theta_d) \tag{3.5}$$

The loss function is divided into two part, label prediction loss and domain prediction loss. The first term is the usual supervised loss for labeled data, and intends to train the feature extractor and label predictor. The second term is an adversarial loss that ensures the features to be domain-invariant and thus aligns the two distributions.

In (3.4), we are minimizing the label prediction loss as well as maximizing the domain prediction loss to achieve a domain-invariant features. In (3.5), we are minimizing the domain prediction loss, and thus training the domain predictor to provide precise prediction of the origin of features.

This min-max problem is solved by adding a gradient reverse layer between feature layer and discriminator as described in [4].

### 3.3.2 Support Augmentation

The fact that we have a very limited amount of labeled examples can limit the performance of our adversarial alignment method, because of insufficient distribution support for labeled data. In order to provide more distribution support, we propose an interpolation method to augment the labeled data and make use of unlabeled data:

$$\lambda \sim \beta(\alpha, \alpha) \tag{3.6}$$

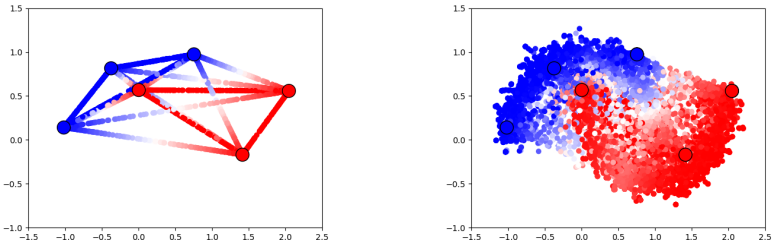
$$\tilde{x} = \lambda x_l + (1 - \lambda) x_u \tag{3.7}$$

$$\tilde{y} = \lambda y_l + (1 - \lambda) \hat{y}_u, \tag{3.8}$$

where $\hat{y}_u$ is the propagated label of $x_u$ using the current trained model. By interpolating between labeled and unlabeled data, we create a new dataset from both distributions and expect it provides better representation for the real distribution.

The motivation behind this design is to not only provide more support for the labeled data, but also make the supports for the two distributions overlap. This means that ideally, this interpolation method itself can also acts as a distribution aligner. Meanwhile, the method also acts as a regularizor that force the network to be more linear.

Our method is closely related to Mixup method proposed in [25], where the interpolation between labeled examples are concerned. However, our method is significantly different from Mixup in the following way:



(a) Mixup: interpolation between random labeled data.

(b) Proposed method: interpolation between labeled and unlabeled data.

Figure 3.4: Difference between Mixup and our proposed method.

- The interpolation is done between labeled and unlabeled data instead of labeled data only.
- Label propagation is used to generate labels for unlabeled data before the interpolation.

Both Mixup and our proposed methods are trying to augment the original data. Formally, the two methods propose generic vicinal distributions [2] instead of empirical distribution to approximate the real data distribution $P$:

$$P_\upsilon(\tilde{x}, \tilde{y}) = \frac{1}{n}\sum_{i=1}^{n} \upsilon(\tilde{x}, \tilde{y}|x_i, y_i),$$

where $\upsilon$ is the proposed vicinity distribution that use $\tilde{x}, \tilde{y}$ instead of $(x, y)$ to approximate $P$. One famous example of vicinal distribution is the Gasussian vicinities

$$\upsilon(\tilde{x}, \tilde{y}|x_i, y_i) = \mathcal{N}(\tilde{x} - x_i, \sigma^2)\delta(\tilde{y}, y_i),$$

where we add Gaussian noise to the input data to achieve better generalization ability.

The proposed method provides better augmented distribution support by using the unlabeled data than Mixup, as shown in Fig 3.4.

The proposed support augmentation method can also be regarded as a consistency regularization method. We will discuss this relationship later in 3.3.2.

**Relationship with Expectation Maximization**

By using the propagated labels, our method has a close relationship with Generalized Expectation Maximization(GEM) algorithm as:

- E-step: Use the current learned model $\hat{f}$ to estimate likelihood of unlabeled data on each class, i.e. the propagated label $\hat{y}_u$.
- M-step: Re-estimate the model $f$, given the estimated label of unlabeled data $(x_u, \hat{y}_u)$.

In this section, we discuss why our proposed interpolation approach can help the EM algorithm.

In a general semi-supervised setting, the complete-data likelihood is:

$$p(x, y, x_u, y_u) = (\Pi_{i=1}^{n} p(x_i, y_i))(\Pi_{i=1}^{m} p(x_{ui}, y_{ui})),$$

here we assume that all examples are iid.

If we use the assumption that $p(y_i|x_i,\theta) \propto \exp(f(x_i|\theta))$, and $\log p(x_i|y_i,\theta) = \log p(x_i) = \frac{1}{n}$, [21] proves that the Q function for unsupervised EM is:

$$Q(\theta|\theta^t) = \mathbb{E}_{y_u|x_u,\theta^t}[\log p(x,y,x_u,y_u|\theta)] \tag{3.9}$$

$$\propto -\frac{1}{n}\sum_{i=1}^{n} CE(f(x_i),y_i) - \frac{1}{m}\sum_{i=1}^{m} CE(f(x_{u_i}),\hat{y}_{u_i}), \tag{3.10}$$

where $\hat{y}_{u_i}$ is the predicted label for unlabeled data $x_{u_i}$ based on the parameters trained on step t.

Since $\arg\max(Q)$ is analyticaly intactible, we apply stochastic gradient descent to maximize Q, which is equivalent to minimizing the following loss:

$$\mathcal{L} \propto \frac{1}{n}\sum_{i=1}^{n} CE(f(x_i),y_i) + \frac{1}{m}\sum_{i=1}^{m} CE(f(x_{u_i}),\hat{y}_{u_i})$$

This loss function can be reformulated using a Bernoulli weight and our augmented data:

$$\mathcal{L} \propto \sum_{i=1}^{N} \lambda_i CE(f(\tilde{x_i}),y_i) + \sum_{i=1}^{N}(1-\lambda_i) CE(f(\tilde{x_i}),\hat{y}_{u_i}), \tag{3.11}$$

where $\lambda \sim bernoulli(0.5)$, $\tilde{x}$ is generated using $\lambda$.

Under this setting, our algorithm falls into the scope of GEM [17]. GEM promises convergence as long as $Q(\theta^{(i+1)},\theta^{(i)}) > Q(\theta^{(i)},\theta^{(i)})$.

The (3.11) term can be seen as a special case of our interpolation approach, where we use $\lambda \sim \beta(\alpha,\alpha)$ instead of $\lambda \sim Bernoulli(0.5)$. This brings us several benefits:

- $p(y_i|x_i,\theta) \propto \exp(f(x_i|\theta))$ is a rather strong assumption in the standard EM setting. In a general backpropagated neural network setting, it is only valid when $p(y_i|x_i,\theta)$ is Bernoulli, which means that we can only use a hard-EM setting. Meanwhile, in our proposed interpolation setting, $\exp(f(x_i|\theta))$ is used to generate meaningful likelihood $\in (0,1)$ instead of hard 0, 1.
- We are able to better approximate the loss function in (3.11).
- We enjoy all the benefits that vicinal distribution brings us as described in previous sections.

**Relationship with Consistency Regularization**

The proposed interpolation approach alone can be viewed as a consistency regularization variant if we reformulate the loss function.

$$\mathcal{L}_{main} = CE(f(\tilde{x}),\tilde{y}) \tag{3.12}$$

$$= \lambda CE(f(\tilde{x}),y_l) + (1-\lambda) CE(f(\tilde{x}),y_u) \tag{3.13}$$

$$= \lambda CE(f(\tilde{x}),y_l) + (1-\lambda) CE(f(\tilde{x}),f(x_u)) \tag{3.14}$$

where $\tilde{x} = \lambda x + (1-\lambda)x_u$, and we use the fact that cross entropy function is linear in its second argument, and $y_u$ is the label we propogated using unlabeled data $x_u$.

The left side of the loss function is a supervised loss term. The input x is augmented by noises from unlabeled data and weighted by the noise intensity. When $1-\lambda$ is large, the noise $(1-\lambda)x_u$ is small, thus weight $\lambda$ is large.

The right term implements the consistency regularization with a $\tilde{x}$-dependent weight. This term penalizes the classifier if $f(\tilde{x})$ and $f(x_u)$ gives different predictions. When $\lambda$ is small, $\tilde{x}$ is close to $x_u$, thus the weight $(1-\lambda)$ is large, giving a high confidence of the penalization. When $\lambda$ is large, $\tilde{x}$ is far away from $x_u$, thus the weight $(1-\lambda)$ is small, giving a low confidence of the penalization.

The main difference between our approach and traditional consistency regularization is that, the perturbation in this case is anisotropic. Instead of isotropically smoothing around each unlabeled data point, we specifically smooth in the direction of labeled points. By doing so, we are assuming these directions are the most vulnerable to our network. This idea is to some extent related to virtual adversarial direction proposed by [16]. As we show later in experiments, our choice of virtual adversarial direction is able to improve generalization performance.

**Support Augmentation: A Different Approach**

Another way to augment the support and make use of unlabeled information is to use the interpolation without propagated label. This is achieved by reformulating the expectation of our interpolated loss function:

$$
\begin{aligned}
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}\ell(f(\lambda x_l+(1-\lambda)x_u),\lambda y_l+(1-\lambda)y_u)=\\
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}\lambda\ell(f(\lambda x_l+(1-\lambda)x_u),y_l)+(1-\lambda)\ell(f(\lambda x_l+(1-\lambda)x_u),y_u)=\\
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}\lambda\ell(f(\lambda x_u+(1-\lambda)x_l),y_u))+\\
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}(1-\lambda)\ell(f(\lambda x_u+(1-\lambda)x_l),y_l)=\\
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}\lambda\ell(f(\lambda x_l+(1-\lambda)x_u),y_l))+\\
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}\lambda\ell(f((1-\lambda)x_u+\lambda x_l),y_l)=\\
&\mathbb{E}_{x_l,y_l\sim p_{\mathcal{D}_l}}\mathbb{E}_{x_u,y_u\sim p_{\mathcal{D}_u}}\mathbb{E}_{\lambda\sim\beta(\alpha,\alpha)}2\lambda\ell(f(\lambda x_l+(1-\lambda)x_u),y_l))
\end{aligned}
$$

The requirement of a label for unlabeled samples is canceled out in the reformulation, and thus yields a simpler form requiring no propagated label. Namely:

$$
\begin{aligned}
\lambda &\sim \beta(\alpha,\alpha)\\
\tilde{x} &= \lambda x_l+(1-\lambda)x_u\\
\tilde{y} &= y_l
\end{aligned}
$$

Although this approach has a conciser form, we found its performance not comparable to our proposed interpolation method.

### 3.3.3 Final Form

We now present the final form of our algorithm, which includes both the adversarial alignment and support augmentation, as shown in Fig 3.5.

In the adversarial alignment part, our method asks the domain predictor to predict soft domain labels, namely $\{\lambda, 1-\lambda\}$ instead of 0-1, thus deviates from its original implementation in [4].

In the interpolation part, the method interpolates between input labeled and unlabeled data, and uses label propagation to generate *pseudo* labels for unlabeled data. This is clearly different from the original implementation, where the interpolation is done between labeled samples. The toy example in Fig 3.4 also shows that this modification can significantly improves the performance in semi-supervised learning.

These modifications along with our revelation on distribution mismatch construct our main contribution and exhibit our insights on semi-supervised learning problems.

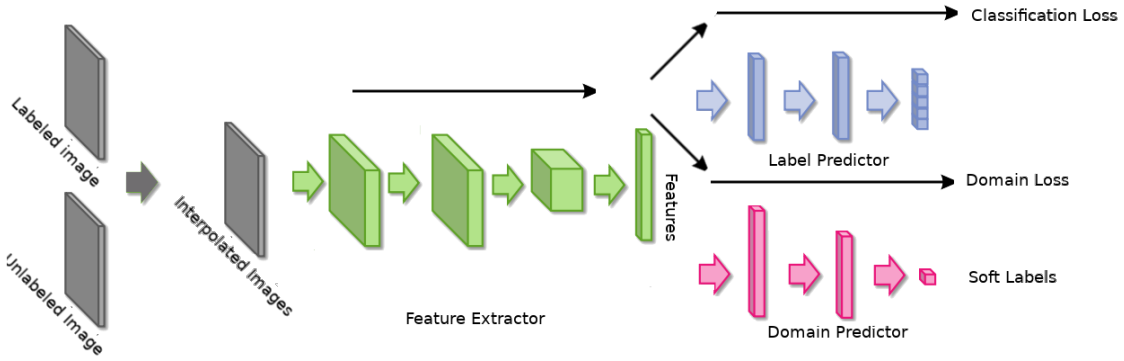



Figure 3.5: Model pipeline discussed in semi-supervised learning tasks. Interpolation is done between labeled and unlabeled data. Adversarial alignment is implemented to address the distribution mismatch problem.

# Chapter 4

# Experiments and Results

In this chapter, we first evaluate our proposed method by conducting ablation study on a semi-supervised learning task. We then compare our method to existing semi-supervised learning methods. We then further extend our scope and apply our method to two different but related tasks, namely domain adaptation, and domain-aware supervised learning task.

## 4.1 Ablation Study on Semi-Supervised Learning Task

To evaluate how the two components work individually in the algorithm, we conduct ablation study on Cifar-10 dataset.

### 4.1.1 Benchmark

#### Cifar-10 Dataset

Cifar-10 [11] is one of the most popular semi-supervised image classification task. The dataset originally consists of 60000 32x32 images in 10 classes. The dataset is balanced, meaning there are 6000 images for each class. The training set includes 50000 images, while the testset includes 10000 images. In semi-supervised learning setting, we split the training set to two parts, one part includes 400 images per class, and the other consists of 4600 images per class. We then erase labels for the latter part. This constructs 4000 labeled images and 46000 unlabeled images.

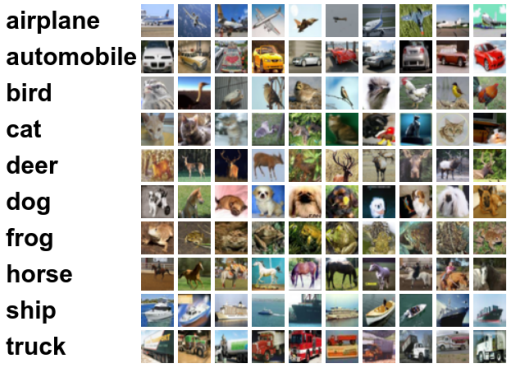
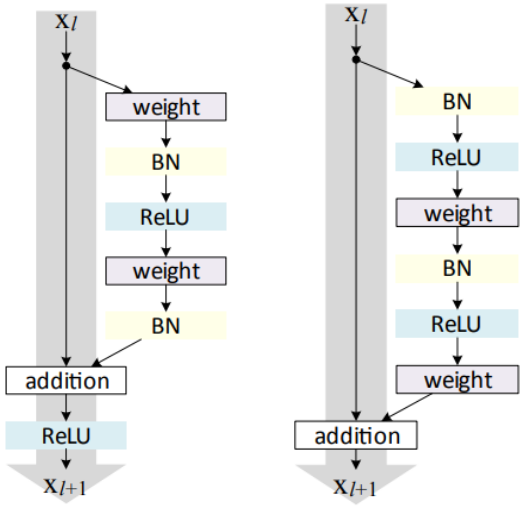



Figure 4.1: Sample images from Cifar-10 dataset [11].




(a) ResNet (b) PreAct-ResNet

Figure 4.2: Difference between ResNet and PreAct-ResNet block. Image from [9].

**Evaluation Metric**

Following [19], we simply use error rate as our evaluation metric for this 10-class classification problem.

### 4.1.2 Architecture

PreAct-ResNet-18 [9] is used in the ablation study as our feature extractor. The main difference between PreAct-ResNet and ResNet [8] is shown in Fig 4.2, and the modification leads to improved results on Cifar-10, and Cifar-100. This 18-layer architecture has been proved to have enough capacity for standard supervised learning tasks on Cifar-10 dataset.

For the label predictor, we use a single dense layer to map the features to logits. Our domain predictor includes two hidden layers, each with 256 units, following a dense layer that map the hidden layers to two channel soft domain labels.

### 4.1.3 Training Procedure

For training, we follow standard data augmentation methods on Cifar-10 dataset: random crops, and horizontal flips. We do not use any augmentation methods for testing. For the ablation study, we use the open-source PreAct-Resnet-18 implementation [24] with a minibatch size of 128. The learning rate starts from 0.1, and is divided by 10 when 50%, and 75% epochs are reached. The network is trained for 300 epochs in total, where one epoch is defined as one iteration over all unlabeled data, i.e. 46000 images. We use a momentum optimizer with 0.9 as the momentum. To choose hyperparameters, we split 4000 images from the unlabeled data as validation data. When hyperparameters are determined, we rerun the experiments to generate our final report numbers on the Cifar-10 test set. The following hyperparameters are used for our report results: weight-decay$= 0.0001$, interpolation $\alpha = 1.0$.

### 4.1.4 Results

The result of our ablation study is listed in Table 4.1. We train our model based on labeled data only as our baseline. Adversarial alignment alone leads to a 1.3% improvement, indicating the alignment helps

to improve generalization performance, but the effect is not significant. Our interpolation approach alone, on the other hand, yields a 6.3% improvement, suggesting the interpolated dataset is more representative than the original dataset. This validates our idea of support augmentation. Finally, we combine the two components and evaluate our proposed method, this gives us a 10.4% improvement over the baseline. This improvement is higher than the simple addition of 1.3% and 6.3%, indicating that the interpolation method helps the distribution alignment, as designed in the method chapter.

Note that, to our knowledge, our method yields highest reported performance on Cifar-10 semi-supervised learning task. As a reference, if label information is given for all training images, the Pre-Act-ResNet-18 model can achieve an error rate of 5.5%.

Table 4.1: Ablation study on Cifar-10.

| **Method** | **Error Rate** |
|---|---|
| Labeled data only | 19.97% |
| Adversarial alignment | 18.67% |
| Interpolation between distributions | 13.79% |
| Adv. alignment + Interpolation | 9.57% |

## 4.2 Semi-Supervised Learning Tasks

To understand the performance of our proposed method, we further compare it with other existing methods in the field. As mentioned in [19], any modification in the network structure and augmentation method can make potentially different result, and lead to an unfair comparison between semi-supervised learning methods. Therefore, we carefully conduct the following semi-supervised learning experiments.

### 4.2.1 Benchmark

We evaluate our method on two popular semi-supervised learning datasets: Cifar-10 and SVHN.

#### Cifar-10 Dataset

The description and preprocessing procedure of Cifar-10 is the same as stated in the previous section.

#### SVHN Dataset

SVHN [18] is a real world digit dataset. The dataset consists of 32x32 training images in 10 classes. Originally, the training set includes 73257 images, while the testset consists of 26032 images. In semi-supervised learning setting, we split the training set to two parts, one part includes 1000 labeled images, and the other consists of 72257 images. We then erase labels for the latter part. In sum, the dataset consists of 1000 labeled images and 72257 unlabeled images for training, and 26032 images for testings.

### 4.2.2 Architecture

Similar to the architecture used in the previous section, we adopt a two-hidden-layer domain predictor, each has 1024 units, and we use a simple dense layer as label predictor. To ensure a fair comparison over other methods, we implement the exact architecture called Conv-Large used in many semi-supervised learning methods [16]. The detailed design is shown in Table 4.2. Furthermore, we use the exact same augmentation and data pipeline, as stated in the paper [16] to ensure a fair comparison.

Figure 4.3: Sample images from SVHN dataset [18].

Table 4.2: CNN feature extractor used in semi-supervised learning tasks.

| **Conv-Large** |
|---|
| 32 × 32 RGB Image |
| 3 × 3 Conv 128 lReLU |
| 3 × 3 Conv 128 lReLU |
| 3 × 3 Conv 128 lReLU |
| 2 × 2 Max-Pool(2 stride) |
| dropout(0.5) |
| 3 × 3 Conv 256 lReLU |
| 3 × 3 Conv 256 lReLU |
| 3 × 3 Conv 256 lReLU |
| 2 × 2 Max-Pool(2 stride) |
| dropout(0.5) |
| 3 × 3 Conv 512 lReLU |
| 1 × 1 Conv 256 lReLU |
| 1 × 1 Conv 128 lReLU |
| Global-Average-Pool |
| Feature |

### 4.2.3 Training Procedure

For both datasets, we use the Tensorflow implementation of the feature extractor from [16] to ensure the same network settings. a minibatch size of 128, Adam optimizer as the trainer, and learning rate starts from 0.001. We train the model for 500 Epochs, and the learning rate starts to decay linearly from Epoch 460. One epoch is defined as 400 updates here.

For Cifar-10 training, we follow standard data augmentation methods: random crops, and horizontal flips. To choose hyperparameters, we split 4000 images from the unlabeled data as validation data. When hyperparameters are determined, we rerun the experiments to generate our final report numbers on the Cifar-10 test set. The following hyperparameters are used for our report results: interpolation $\alpha = 1.0$.

For SVHN training, we use random crop as the only augmentation method for SVHN. For hyperparameters tuning, we split 1000 images from the unlabeled data as validation data. When hyperparameters are determined, we rerun the experiments to generate our final report numbers on the Cifar-10 test set. The following hyperparameters are used for our report results: interpolation $\alpha = 0.1$.

To conclude, we follow the exact training procedure as [16] did.

### 4.2.4 Results

We report the results on Cifar-10 and SVHN in Table 4.3. For Cifar-10 task, the proposed method outperformed the previous state-of-the-art method by 0.2%. For SVHN, the method is very comparable. This shows that our proposed method achieves superior or comparable results on semi-supervised learning tasks.

Table 4.3: Performance of proposed method on semi-supervised learning tasks using Conv-Large described in Table 4.2. Error rates on Cifar-10 and SVHN are reported.

| **Method** | **Cifar-10** | **SVHN** |
|---|---|---|
| Source Only | 34.85% | 19.30% |
| Π Model [13] | 12.36% | 4.82% |
| VAT [16] | 11.36% | 5.42% |
| VAT+Ent [16] | 10.55% | 3.86% |
| Ours | 10.30% | 4.04% |

## 4.3 Domain Adaptation Tasks

### 4.3.1 Background

As mentioned in previous chapter, our method can be further extended to any scenario where two or more data distributions are involved. Domain adaptation is one these cases where images from two domains are available as training data. Images from both domains share the same set of labels. Formally, we consider the following classification problem:
Given:

- A small set of labeled training data from Domain A. $\mathcal{D}_l = \{(x_{l1}, y_{l1}), ...(x_{ln}, y_{ln})\}$
- A large set of unlabeled training data from Domain B. $\mathcal{D}_{ul} = \{(x_{u1}, y_{u1}), ...(x_{um}, y_{um})\}$, where $y_{ui}$ is unknown.

The goal of the problem is to improve the classification result on images from Domain B.

The modified pipeline for this task is shown in Fig 4.4. Note that the only minor difference is that we are interpolating between the two different domains now.

### 4.3.2 Benchmark

We evaluate our method on two popular domain adaptation tasks: MNIST to MNIST-M and Office-31.

#### MNIST to MNIST-M

MNIST [14] is the famous black and white digit dataset. MNIST-M [4] is a modified version of MNIST. The dataset is obtained by extracting digits from the original MNIST dataset, and replaced the background with random color photos from BSDS500. Both dataset consists of 10 classes and 55000 images.

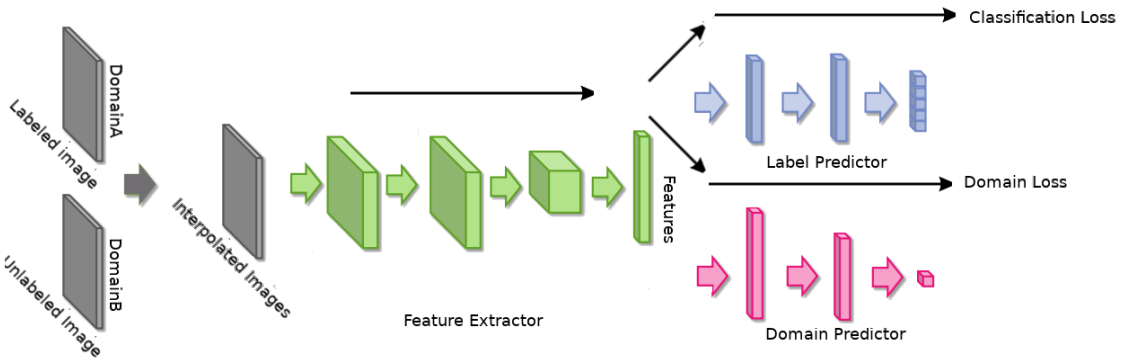



Figure 4.4: Model pipeline discussed in domain adaptation tasks. Interpolation is done between two different domains. Adversarial alignment is implemented to address the distribution mismatch problem.

(a) MNIST images

(b) MNIST-M images

Figure 4.5: Sample images from MNIST and MNIST-M [14].

**Office-31**

Office-31 [20] is a dataset containing three domains: Amazon, DSLR, and Webcam. The dataset consists of 2817 images for Amazon, 795 for DSLR, and 498 for DSLR. Thirty-one different classes are included in the dataset. The domain Amazon is significantly different from DSLR and Webcam. Images from the Amazon domain usually has a white background, and reflects the preference of the shopping website. DSLR and Webcam images are rather similar, but have a domain difference due to the difference between cameras.

### 4.3.3 Architecture

We mainly follow architecture choices from DANN [5] to have a fair comparison.


Figure 4.6: Sample images from Office-31 dataset [20]

**MNIST to MNIST-M**

Following DANN [5], we use the architecture described in Table 4.4.

Table 4.4: Network architecture used in MNIST to MNIST-M task.

| **Feature Extractor** | **Label Predictor** | **Domain Predictor** |
|---|---|---|
| RGB Image | 256D Feature | 256D Feature |
| $5 \times 5$ Conv 32 lReLU | 100 lReLU | Gradient Reverse Layer |
| $2 \times 2$ Max-Pool(2 stride) | 100 lReLU | 100 lReLU |
| $5 \times 5$ Conv 48 lReLU | 10 dense + Softmax | 1 dense + Softmax |
| $2 \times 2$ Max-Pool(2 stride) | | |
| Feature | Label | Domain |

**Office-31**

Because the limited number of images in Office-31, previous studies use pretrained models for this domain adaptation task. We follow the same procedure and finetune AlexNet [12] using Office-31 images. Pre-trained AlexNet parameters from ImageNet is used. AlexNet and a 256 unit bottleneck layer are used as our feature extractor.

Similar to previous experiments, we use a two-hidden-layer domain predictor and a single layer label predictor. We additionally add dropout layers with 0.5 dropout rate in the domain predictor for Office-31 task.

### 4.3.4 Training Procedure

We follow DANN [5] for most of the training procedures.

For MNIST to MNIST-M training, we use minibatch size of 64. We train the model for 1000 epochs, and use momentum optimizer with 0.9 as momentum. The learning rate starts from 0.01 and is decayed based on the following equation:

$$lr = \frac{0.01}{(1. + 10 * p)^{0.75}}$$

For Office-31 training, we use 227x227 random crops as augmentation method. We train the network with minibatch sizes of 128. The base learning rate starts from 0.001, and is decayed to 0.0001 in the last 20 Epochs. Layers without pre-trained parameters use a learning rate ten times higher than the base rate. To mimic the same learning procedure in Caffe, we times the learning rate for bias by two. The network is trained for 400 epochs in total, where one epoch is defined as one iteration over all labeled and unlabeled data. We use stochastic gradient for training.

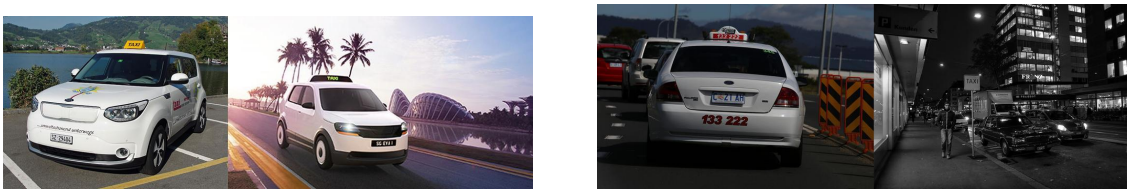
To choose hyperparameters, we split 1000 images from the target domain as validation data. When hyperparameters are determined, we rerun the experiments to generate our final report numbers on the full target set. The following hyperparameters are used for our reported results: no weight-decay, interpolation $\alpha = 1.0$, and the following for Office-31: weight-decay$= 0.0001$, interpolation $\alpha = 0.2$.

### 4.3.5 Results

We report the result of MNIST to MNIST-M task in Table 4.5. As shown in the table, our proposed method gives a 16.3% of performance improvement. Note that the only difference between the adversarial-alignment-only method and our proposed method is the interpolation component.

Table 4.5: Performance of proposed method on MNIST to MNIST-M domain adaptation task.

| Method | Error Rate |
|---|---|
| DANN [5] (Adversarial Alignment Only) | 23.3% |
| Ours | 7.0% |



(a) Google images (b) Flickr images

Figure 4.7: Sample images from Webvision-Tiny [15].

In addition, we present the performance of our model on Office-31 in Table 4.6. Our model outperforms the previous state-of-the-art method in five out of six tasks, and is rather comparable on the remaining one. This demonstrates the effectiveness of the proposed method in domain adaptation tasks.

Table 4.6: Performance of proposed method on Office-31 Dataset as domain adaptation Finetuning AlexNet task.

| Method | A-W | D-W | W-D | A-D | D-A | W-A | Average Accuracy |
|---|---|---|---|---|---|---|---|
| Source only | 61.6% | 95.4% | 99.0% | 63.8% | 51.1% | 49.8% | 70.1% |
| DANN [5] | 73.0% | 96.4% | 99.2% | 72.3% | 53.4% | 51.2% | 74.3% |
| AutoDial [1] | 73.6% | 96.6% | 99.6% | 73.6% | 58.3% | 59.4% | 77.1% |
| Proposed Method | 78.4% | 96.4% | 99.8% | 74.1% | 58.4% | 59.5% | 77.8% |

## 4.4 Domain-Aware Supervised Learning Task

### 4.4.1 Background

So far, we mainly concern problems where unlabeled data are involved, but we are also interested in how our method can improve the performance on supervised learning problems. In this section, we extend our experiment to a supervised classification problem where two domains are involved.

### 4.4.2 Benchmark

#### Webvision-Tiny Dataset

We use a subset of Webvision dataset [15] as Webvision-Tiny. This sub-sampled dataset includes 1000 classes with two different domains: Google and Flickr. The two domains reflects the preference of search engines of Google and Flickr. Google images are in general with clearer light conditions, with the main object in the center. In contrast, Flickr images includes more artistic effects and complex light conditions. We sample 179 images per class per domain for our experiment to ensure there is no class-balance problem.

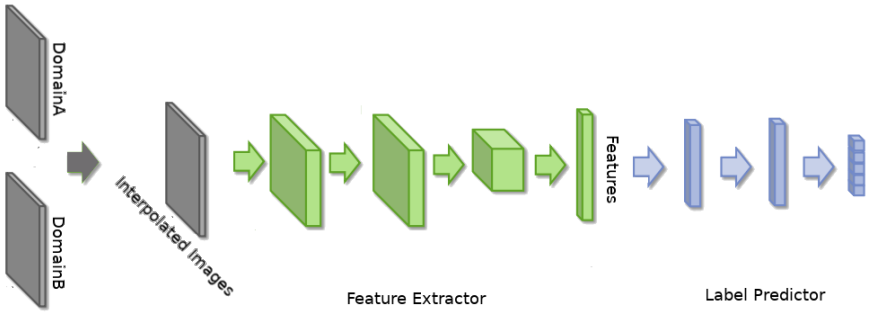



Figure 4.8: Model pipeline discussed in domain-aware supervised learning task. Interpolation is done between two different domains. No adversarial alignment is involved.

#### Classification Metrics

The classification metric we use for supervised image classification tasks on Webvsion-Tiny is Top-5 error. The last layer of our model is softmax and outputs a probability distribution over all classes. The top-N error rate is calculated based on it. If the ground truth is not among the 5 most probable classes given by the last layer, then the number of erroneous predictions is accumulated.

### 4.4.3 Architecture

The main pipeline of the model is shown in Fig 4.8. We use the famous Resnet-50 [8] as our main architecture. The distributed Tensorpack implementation [24] of ResNet-50 is modified for our experiment.

### 4.4.4 Training Procedure

For training, we follow standard data augmentation methods described in [8], including color augmentation, random crops, and horizontal flips. We train our Resnet with a minibatch size of 128. The learning rate starts from 0.1, and is divided by 10 at Epoch 60, 90, 110. The network is trained for 130 epochs in total. We use Adam optimizer for training. The following hyperparameters are used for our report results: weight-decay$= 0.0001$, interpolation $\alpha = 1.0$.

### 4.4.5 Results

We report single-crop top-5 error rate on webvision-Tiny in Table 4.7. While random interpolation outperforms the baseline, our proposed interpolation approach further improves the performance by 1.08%, thus supports our assumption that interpolation between distributions can provide better representations for both distributions.

Table 4.7: Performance of proposed interpolation method on Webvision-Tiny dataset (Single Crop).

| | **No Interpolation** | **Random Interpolation [25]** | **Interpolation between Google and Flickr** |
|---|---|---|---|
| Top5 Error | 20.30% | 19.62% | 18.56% |

# Chapter 5

# Discussion

We proposed a new semi-supervised learning method based on distribution alignment and interpolation.

We revealed that, in semi-supervised learning, when labeled data are rare, sampling bias and distribution mismatch exist between labeled and unlabeled data. We showed that aligning the two distributions by adversarial training can alleviate the mismatch. Furthermore, by extensive experiments, we showed augmenting distribution support by interpolating between labeled and unlabeled data can further improve the alignment. Effectiveness of the proposed method was demonstrated on multiple semi-supervised learning tasks and domain adaptation tasks such as CIFAR-10, SVHN, MNIST-M, and Office-31.

The proposed method has the following strong points:

- **Performance** We achieved state-of-the-art results on the semi-supervised learning task Cifar-10, as well as de facto standard domain adaptation task Office-31.
- **Robustness** The proposed methods was validated for multiple datasets of different nature under various setups such as semi-supervised learning, domain adaptation, and supervised learning. This convinced us that our proposed method can be further extended to related tasks such as semi-supervised image segmentation tasks.
- **Complexity** The key interpolation component is simply a multiplyaccumulate operation, which can be efficiently implemented in modern CPUs and GPUs.

The work also opens up several directions for future work. First, robustness of our method against adversarial examples is one interesting topic. We have shown by experiments, through augmenting distribution support, the interpolated data lead to improved generalization performance. We have also shown that the interpolation is equivalent to choosing virtual adversarial direction as the directions of labeled data, but will this augmented dataset lead to a model more resilient to adversarial examples? Intuitively in 2D, the interpolation is likely to provide more support in areas where adversarial examples might end up, but the effect in higher dimensions need to be carefully evaluated. In addition, extending the interpolation in input space to hidden states is also an promising direction. The interpolation in input space is non-instinctive because it yields an augmented dataset that deviates from the original manifold. Interpolation between hidden states such as neurons from feature layer does not have this problem. Although theoretically it is hard to determine whether such a modification can lead to further improvement, it is an interesting direction to explore.

# Bibliography


[1] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, pages 5077–5085, 2017.

[2] Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in neural information processing systems*, pages 416–422, 2001.

[3] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[10] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

[11] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[16] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[17] Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[18] 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*, volume 2011, page 5, 2011.

[19] Augustus Odena, Avital Oliver, Colin Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of semi-supervised learning algorithms. 2018.

[20] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[21] Mark Schmidt. Em for semi-supervised learning and mixture models, February 2016.

[22] Baochen Sun and Kate Saenko. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*, pages 24–1, 2015.

[23] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[24] Yuxin Wu et al. Tensorpack, 2016.

[25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

**Statement regarding plagiarism when submitting written work at ETH Zurich**

By signing this statement, I affirm that I have read the information notice on plagiarism, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Information notice on plagiarism:
http://www.ethz.ch/students/semester/plagiarism_s_en.pdf

| | | | |
|---|---|---|---|
| Supervisor | Dr. Wen Li Eirikur Agustsson | Prof. Dr. Luc van Gool | |
| Student | Qin Wang | qwang@student.ethz.ch | |
| Place and date | Zurich, 24/08/2018 | Signature |  |