Min-Max Statistical Alignment for Transfer Learning

Samitha Herath[1,4], Mehrtash Harandi[2,4], Basura Fernando[1,5], and Richard Nock[1,3,4]
[1]The Australian National University, [2]Monash University, [3]The University of Sydney
[4]DATA61-CSIRO, Australia
[5]Human-Centric AI Programme, A*STAR, Singapore
Samitha.Herath@data61.csiro.au, Mehrtash.Harandi@monash.edu,
Fernando_Basura@scei.a-star.edu.sg, Richard.Nock@data61.csiro.au

Abstract

A profound idea in learning invariant features for transfer learning is to align statistical properties of the domains. In practice, this is achieved by minimizing the disparity between the domains, usually measured in terms of their statistical properties. We question the capability of this school of thought and propose to minimize the maximum *disparity between domains. Furthermore, we develop an end-to-end learning scheme that enables us to benefit from the proposed min-max strategy in training deep models. We show that the min-max solution can outperform the existing statistical alignment solutions, and can compete with state-of-the-art solutions on two challenging learning tasks, namely, Unsupervised Domain Adaptation (UDA) and Zero-Shot Learning (ZSL).*

1. Introduction

Minimizing the statistical disparity between distributions is a fundamental approach used to learn domain invariant features [25, 26, 14]. In this work and in contrast to the previous attempts, we propose to learn features by minimizing the maximum statistical disparity. We show that by minimizing the maximum (*i.e.*, min-max) statistical disparity, we can learn better domain invariant features (compared to features attained by only minimizing the disparities). In particular, we demonstrate that in Unsupervised Domain Adaptation (UDA) [8, 24] and Zero-Shot Learning (ZSL) [31], the learned features by min-max alignment lead to comparable performances to the very involved state-of-the-art methods specifically designed to address each task (*e.g.*, adversarial solutions).

Recent techniques for UDA and ZSL aim to learn task-independent and discriminative features through end-to-end learning [9, 26, 30]. A prominent idea here is to learn a mutual space where examples from the source and target domains behave similarly from a statistical point of view. For example, the CORrelation ALignment (CORAL) [25] and its variants [27, 26] opt to minimize the statistical disparity of data measured by the second order statistics.

Our idea goes beyond minimizing statistical disparities and makes use of a novel structure, namely the *confusion network* to align distributions in a min-max framework. The confusion network, as the name implies, is by itself a neural network. Therefore, the proposed min-max solution can be seamlessly used in deep learning frameworks for end-to-end training (*see* Fig. 1). Our code is available at https://bitbucket.org/sherath.

One may wonder why min-max? In other words, what power such a solution endows that a min solution does not. In learning theory, methods such as SVM [3] , aim at minimizing the maximal loss. Nevertheless, very few studies in deep learning [23, 20] make use of the min-max framework, our paper being one. This can be attributed to the fact that minimizing a loss can be conveniently achieved using stochastic techniques.

For the problem of interest in this paper, the min-max solution has a somehow intuitive meaning. We are interested in finding invariant features across domains with mismatched statistics. Learning representations by minimization may result in degeneracy (*e.g.*, by collapsing the space into a point). On the other hand, maximization can preserve the variance of the distributions, avoiding degeneracies.

To visualize the difference between min and min-max frameworks, we designed a toy example (*see* Fig. 1) where the task is to align an input distribution (given in purple, yellow and red points) to a fixed target distribution (given in blue). The top row shows the alignment by minimization according to [26]. In the second row, we aligned the yellow points, again by minimizing the disparities using the KullbackLeibler (KL) divergence. Finally, the third row shows our min-max alignment. The figure is self-explanatory, with the proposed min-max solution showing the most consistent alignment on all studied cases.

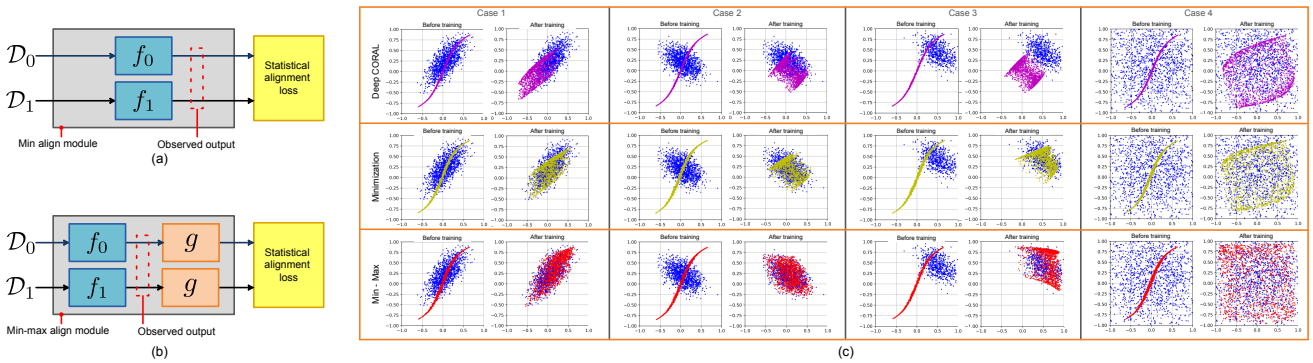

Figure 1. **Toy data demonstration (Should be viewed in color).** A comparison of aligning the input data, $\mathcal{D}_1$ (*i.e.*, Gaussian noise) to synthetic target data, $\mathcal{D}_0$ (given in blue color points) with Deep CORAL [27], minimization of KL divergence and min-max alignment with KL divergence. **(a)** The schematic diagram for aligning the output features of f_1 with $\mathcal{D}_0$ by minimizing a statistical alignment loss (*e.g.*, correlation loss as in Deep CORAL, KL divergence). **(b)** The schematic diagram using the proposed **confusion network**, g for min-max alignment of the output features from f_1. **(c)** Comparison of statistically aligned outputs of f_1 at the beginning and after training for Deep CORAL, minimization and Min-max alignment. We provide details of this experiment in our supplementary material.

In summary, our contributions in this paper are;

- We propose min-max statistical alignment for domain invariant feature learning using a novel confusion network.
- We provide two frameworks to use the proposed statistical alignment for UDA and ZSL.

2. Min-Max Statistical Alignment

In this section, we introduce our proposed min-max solution. We start by describing the notations. Bold capital letters denote matrices (*e.g.*, $\boldsymbol{X}$) and bold lower case letters show column vectors (*e.g.*, $\boldsymbol{x}$). We represent sets with curled capital letters, (*e.g.*, $\mathcal{X}$) and their cardinality with $|\cdot|$ (*e.g.*, $|\mathcal{X}|$). The Frobenius norm of a matrix is shown by $\|\cdot\|_F$. We use $D_{KL}(P_0\|P_1)$ to denote the Kullback-Leibler divergence between two distributions P_0 and P_1. We use the notation $\mathcal{D}$ to represent a domain (*e.g.*, $\mathcal{D}_0$ for domain 0). For simplicity, we also use this notation $\mathcal{D}$ when referring to the data samples from a domain (*e.g.*, $\boldsymbol{x}_i^{(0)} \in \mathcal{D}_0$, for $i = 1, 2, 3, \cdots, N_0$ to denote the samples with $y_i^{(0)} \in \mathcal{D}_0$ being their labels from domain 0.). We consider the dimensionality of samples from a domain $\mathcal{D}_k$ to be n_k (*e.g.*, $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^{n_0}$).

Our objective is to learn two non-linear mappings, $f_k(\cdot, \boldsymbol{\theta}_k) : \mathbb{R}^{n_k} \rightarrow \mathbb{R}^d$, $k \in \{0, 1\}$ to embed samples from domains $\mathcal{D}_k$, $k \in \{0, 1\}$ to a shared feature space such that they are statistically aligned . The non-linear mappings are parametrized by $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ and realized by two neural networks. In particular, we consider the case where there is no direct pair-wise correspondences between instances from the two domains. This is due to the fact that statistical alignment is widely used to address problems such as UDA and ZSL where associations are not available.

As such, statistical alignment can be performed by minimizing a loss reflecting the statistical disparity such as KL-divergence (D_{KL}) between domain feature distributions.When the feature distribution, P_k of domain k is parameterized with the mean, $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} f_k(\boldsymbol{x}_i^{(k)})$ and the covariance, $\boldsymbol{\Sigma}_k = \frac{1}{(N_k-1)}\sum_{i=1}^{N_k}(f_k(\boldsymbol{x}_i^{(k)}) - \boldsymbol{\mu}_k)(f_k(\boldsymbol{x}_i^{(k)}) - \boldsymbol{\mu}_k)^T$ the statistical misalignment loss, $\mathcal{L}_u$ between $\mathcal{D}_0$ and $\mathcal{D}_1$ can be expressed using symmetric KL-divergence as,

$$\mathcal{L}_u = \frac{1}{2}\big(D_{KL}(P_0\|P_1) + D_{KL}(P_1\|P_0)\big). \tag{1}$$

A widely accepted assumption is to model distributions as Gaussians, leading to

$$D_{KL}(P_0\|P_1) = \frac{1}{2}\Big(\mathrm{tr}\big(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\big) + \log\left(\frac{\det\boldsymbol{\Sigma}_1}{\det\boldsymbol{\Sigma}_0}\right) + \big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\big)^T\boldsymbol{\Sigma}_1^{-1}\big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\big) - d\Big). \tag{2}$$

In realizing min-max alignment we propose to make use of an additional mapping (*i.e.*, the *confusion network*), $g(\cdot, \boldsymbol{\theta}_g) : \mathbb{R}^d \rightarrow \mathbb{R}^p$. The inputs to the confusion network are the domain features from the functions $f_0(\cdot, \boldsymbol{\theta}_0)$ and $f_1(\cdot, \boldsymbol{\theta}_1)$. We refer to the output features of $g(\cdot, \boldsymbol{\theta}_g)$ as the confused features. We implement the confusion network, $g(\cdot, \boldsymbol{\theta}_g)$ using a neural network parameterized by $\boldsymbol{\theta}_g$. Thereafter, we propose to perform the min-max alignment by optimizing,

$$\min_{\boldsymbol{\theta}_0,\, \boldsymbol{\theta}_1}\ \max_{\boldsymbol{\theta}_g} \tilde{\mathcal{L}}_u\ , \tag{3}$$

with

$$\tilde{\mathcal{L}}_u = \frac{1}{2}\big(D_{KL}(\tilde{P}_0\|\tilde{P}_1) + D_{KL}(\tilde{P}_1\|\tilde{P}_0)\big). \tag{4}$$

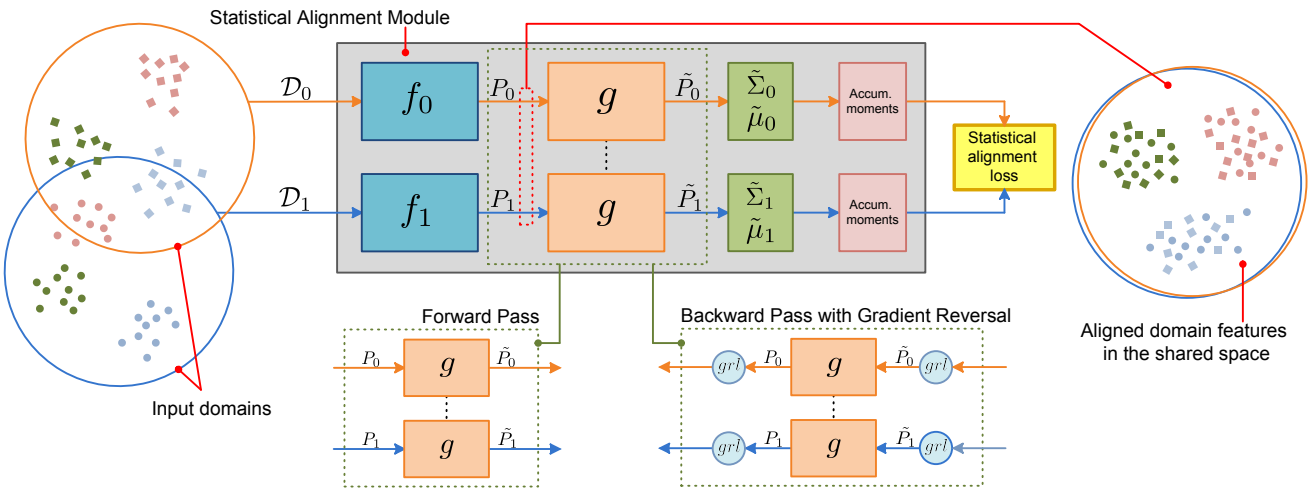

Figure 2. **Schematic diagram for the proposed min-max statistical alignment (Best viewed in color).** The input domains $\mathcal{D}_0$ (square markers) and $\mathcal{D}_1$ (circular markers) are aligned by min-max optimization of the statistical loss $\tilde{\mathcal{L}}_u$.To realize the min-max training we propose using the additional confusion model, g. We propose accumulation of computed mini-batch moments (*i.e.*, means and covariances) prior to computing the statistical alignment loss. The statistically aligned features are obtained through the feature extraction models f_0 and f_1. We use marker colors to represent the class labels of the instances. To incorporate the min-max optimization we contain our proposed confusion network between two gradient reversal layers (*i.e.*, grl).

However, distinct to the defined statistical loss in equation (1), the feature distributions, $\tilde{P}_k$, $k \in \{0, 1\}$ are parameterized with,

$$\tilde{\boldsymbol{\mu}}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} g \circ f_k(\boldsymbol{x}_i^{(k)}), \tag{5}$$

$$\tilde{\boldsymbol{\Sigma}}_k = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} \Big(g \circ f_k(\boldsymbol{x}_i^{(k)}) - \tilde{\boldsymbol{\mu}}_k \Big) \Big(g \circ f_k(\boldsymbol{x}_i^{(k)}) - \tilde{\boldsymbol{\mu}}_k \Big)^\top.$$

The objective of the confusion network is to maximize the statistical disparity (*see* (3)). Furthermore, we learn f_0 and f_1 to minimize the statistical disparity of the confused features. Therefore, we perform a minimization of the maximum statistical disparity between the two domain features.

In a way, our confusion network can be considered as an attention model. The attention is in particular given for features that maximizes the statistical disparity between the domains. The theorem below establishes the condition to recover minimization as an especial case of the min-max framework.

Theorem 1. *If the confusion function, g is a linear invertible transformation, $\boldsymbol{Q}$ with $\boldsymbol{Q}\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{-1}\boldsymbol{Q} = \mathbf{I} \in \mathbb{R}^{d \times d}$ then the proposed min-max statistical alignment by confusion is equivalent to statistical alignment by minimization. Here, d is the dimensionality of the domain input and confused features.*

Proof. The proof is provided in the supplementary material due to space limitations. ■

The behaviour of the proposed confusion network is similar in spirit to the discriminator in Generative Adversarial Networks (GANs) [11]. However, unlike the classification objective of the GAN's discriminator, the confusion network is trained to maximize a statistical misalignment.

2.1. Maximization with the Confusion Network

To train a Deep Neural Network (DNN), parameters of the network are updated such that the end loss is minimized. That is, each parameter is updated in the negative direction of the gradient of the loss function with respect to it. However, in the case of the confusion network, we need to learn the parameters in a way that the end loss is maximized (*see* (3)). In other words, we require to perform a gradient ascent for the confusion network. To seamlessly integrate this gradient ascent into our solution, we perform a direction reversal of the back propagated gradients into the confusion network. For this we use the gradient reversal layer (*i.e.*, *grl* layer) of Ganin and Lempitsky, [8]. The grl layer acts as an identity mapping in the forward pass. However, during the backward pass it reverses the back-propagated gradient direction by negation. Furthermore, we enclose the confusion network between two grl layers (*see* Fig. 2). Thereby, we make sure that the gradients back-propagated into f_0 and f_1 are compatible with gradient descent for minimization.

2.2. Moment Accumulation

Training a network by stochastic optimization is central to deep learning. In the context of statistical alignment, this translates into computing statistics (*i.e.*, means and covari-

ances) per mini-batch, followed by aligning the mini-batch statistics. For the min-max framework, we propose to make use of the accumulated statistics instead. When $\tilde{\boldsymbol{\Sigma}}^{(t)}$ and $\tilde{\boldsymbol{\mu}}^{(t)}$ are the computed covariance matrix and the mean vector for mini-batch at iteration t, the accumulated moments, $\tilde{\boldsymbol{\Sigma}}^{(t)}_{accu.}$ and $\tilde{\boldsymbol{\mu}}^{(t)}_{accu.}$ are computed as,

$$\tilde{\boldsymbol{\Sigma}}^{(t)}_{accu.} = m \times \tilde{\boldsymbol{\Sigma}}^{(t-1)}_{accu.} + (1-m) \times \tilde{\boldsymbol{\Sigma}}^{(t)}, \tag{6}$$

$$\tilde{\boldsymbol{\mu}}^{(\boldsymbol{t})}_{\boldsymbol{accu.}} = m \times \tilde{\boldsymbol{\mu}}^{(t-1)}_{accu.} + (1-m) \times \tilde{\boldsymbol{\mu}}^{(\boldsymbol{t})}. \tag{7}$$

Here, $0 \leq m < 1$ is the momentum hyper-parameter for the accumulation. We integrate this moment accumulation along with the proposed min-max alignment into a single statistical alignment module as in Fig. 2. In the supplementary material, we provide a study of the effect of accumulation.

3. Case studies : Min-Max Statistical Align

In this section, we will show how the min-max framework can be used to address two case studies, namely UDA and ZSL.

3.1. Case 1 : Unsupervised Domain Adaptation

In UDA, labeled samples from a source domain, $\mathcal{D}_s$ are used to train a classifier to classify unlabeled samples from the target domain, $\mathcal{D}_t$. Samples from both domains are assumed to share the same set of classes, $\mathcal{C} = \{1,\ 2,\ 3,\ \cdots,\ c\}$. Here, we use letters "*s*" and "*t*" to refer to **s**ource and **t**arget domains respectively [1].

It is typical to use a two-stream network in deep UDA where each stream corresponds to a specific domain (*i.e.*, either $\mathcal{D}_s$ or $\mathcal{D}_t$). Thereafter, the source domain stream is trained on classifying labeled source samples and the target domain stream is trained to generate features that match the source features distribution [8, 27]. We realize our UDA model with two deep network streams, $h_s = \text{softmax} \circ h \circ f_s$ and $h_t = \text{softmax} \circ h \circ f_t$. Here, h_s and h_t are the source and target domain model streams, respectively. The model, $h(\cdot, \boldsymbol{\theta}_h) : \mathbb{R}^d \rightarrow \mathbb{R}^c$ represents a shared classifier with parameters, $\boldsymbol{\theta}_h$. The source and target domain feature extraction models, f_s and f_t are parameterized with $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_t$, respectively (*see* Fig. 3(a) for a schematic).

For UDA, our objective is to jointly learn the shared feature space and the classifier $h(\cdot, \boldsymbol{\theta}_h)$. Per the discussion in § 2, the proposed min-max alignment will be used for learning the shared feature space. We use the softmax cross-entropy loss on labeled source domain samples to train the classifier. All in all, the UDA model is trained end-to-end by optimizing,

$$\min_{\boldsymbol{\theta}_s,\ \boldsymbol{\theta}_t,\ \boldsymbol{\theta}_h} \max_{\boldsymbol{\theta}_g} \ \mathcal{L}_{d,s} + \lambda_t \mathcal{L}_{d,t} + \lambda_u \tilde{\mathcal{L}}_u. \tag{8}$$

Here, $\mathcal{L}_{d,s}$ is the softmax cross-entropy loss computed using the labeled source samples, the loss term $\mathcal{L}_{d,t}$ is the entropy loss,

$$\mathcal{L}_{d,t} \ = \ -\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}_t}[h_t(\boldsymbol{x})^T \log h_t(\boldsymbol{x})], \tag{9}$$

computed from unlabeled target domain samples as in [24]. We denote the proposed statistical alignment loss with $\tilde{\mathcal{L}}_u$. The parameter λ_u and λ_t are training hyper-parameters. We will provide a study on the effect of these parameters in the supplementary material.

3.2. Case 2 : Zero-Shot Learning

ZSL is the problem of identifying instances never seen during the training. We use $\mathcal{C}$ and $\tilde{\mathcal{C}}$ to represent the set of seen and unseen classes, respectively[2]. We define the domain, $\mathcal{D}_s$ to contain the labeled training instances from classes in $\mathcal{C}$. For training, we are provided with semantic descriptions for all the classes, $\tilde{\mathcal{C}} \cup \mathcal{C}$. Without losing generality and inline with general practice (*e.g.*, [31]), semantic descriptions are in the form of attribute vectors. Each element of an attribute vector represents a meaningful property of the seen and unseen classes (*e.g.*, has stripes, has four legs, has a long tail). We will use $\mathcal{D}_{att.}$ and $\tilde{\mathcal{D}}_{att.}$ to represent the domains related to attribute vectors describing classes $\mathcal{C}$ and $\tilde{\mathcal{C}}$, respectively.

Following [30], we propose a two-stage ZSL solution. In the first stage, we will train a conditional generator, $f_{att.}(\cdot, \boldsymbol{\theta}_{att.}) : \mathbb{R}^{n_{att.}} \rightarrow \mathbb{R}^d$ with $n_{att.}$ denoting the dimensionality of the attribute vectors. To be specific, we use our min-max statistical alignment to train a model to generate discriminative instance features given an attribute vector from a seen class. Note that $\mathcal{D}_0 \sim \mathcal{D}_s$ and $\mathcal{D}_1 \sim \mathcal{D}_{att.}$ per notations used in § 2. Later, we use $f_{att.}$ to generate features for unseen classes, $\tilde{\mathcal{C}}$ given their attributes.

As for the first stage, we define two network streams, $h_s = \text{softmax} \circ h \circ f_s$ and $h_{att.} = \text{softmax} \circ h \circ f_{att.}$ (*see* Fig. 3 for a schematic). Here, $f_s(\cdot, \boldsymbol{\theta}_s) : \mathbb{R}^s \rightarrow \mathbb{R}^d$ is a feature extraction network for real seen class instances. The function $h(\cdot, \boldsymbol{\theta}_h) : \mathbb{R}^d \rightarrow \mathbb{R}^{|\mathcal{C}|}$ is a shared classifier for real and generated seen class features. We parameterize h, $f_{att.}$ and f_s with $\boldsymbol{\theta}_h$, $\boldsymbol{\theta}_{att.}$ and $\boldsymbol{\theta}_s$, respectively. Thereafter, we learn our generator network by following the two optimizations,

$$\min_{\boldsymbol{\theta}_s,\ \boldsymbol{\theta}_h} \ \mathcal{L}_{d,s}, \tag{10}$$

$$\min_{\boldsymbol{\theta}_{att.}} \max_{\boldsymbol{\theta}_g} \ \mathcal{L}_{d,att.} + \lambda_u \tilde{\mathcal{L}}_u. \tag{11}$$

[1] Note the domain equivalences $\mathcal{D}_0 \sim \mathcal{D}_s$ and $\mathcal{D}_1 \sim \mathcal{D}_t$ with the discussion in § 2.

[2] Note $\mathcal{C} \cap \tilde{\mathcal{C}} = \emptyset$

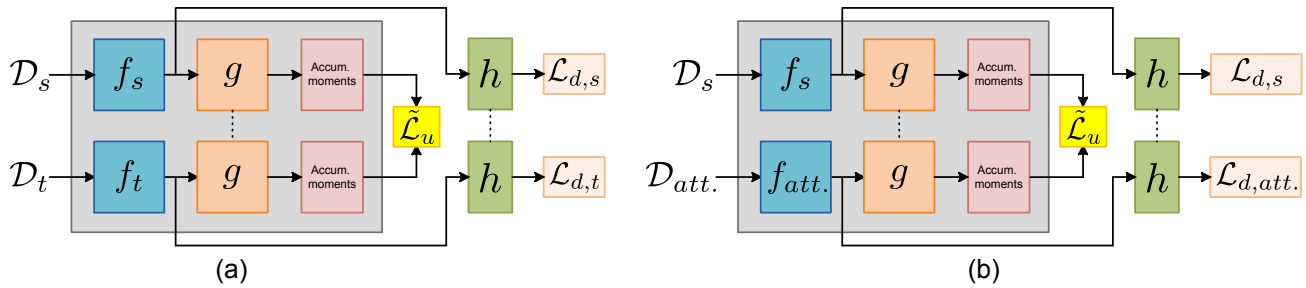

Figure 3. **Schematic diagram for UDA and ZSL models with the proposed statistical alignment** (a) The UDA model is trained on the softmax cross entropy loss, $\mathcal{L}_{d,s}$ on labeled source data, the discriminative cross-entropy loss, $\mathcal{L}_{d,t}$ on unlabeled target domain data and the statistical alignment loss, $\tilde{\mathcal{L}}_u$. (b) The generator training stage of the proposed two staged ZSL model. The model is trained on discriminative losses, $\mathcal{L}_s$, $\mathcal{L}_{att.}$ and the statistical alignment loss, $\tilde{\mathcal{L}}_u$.

Here, $\mathcal{L}_{d,s}$ and $\mathcal{L}_{d,att.}$ are softmax cross-entropy loss functions computed on predictions from networks h_s and $h_{att.}$, respectively. The scalar constant λ_u is a training hyper-parameter. The proposed statistical alignment losses between the domains $\mathcal{D}_s$ and $\mathcal{D}_{att.}$ is given by $\tilde{\mathcal{L}}_u$ (*see* § 2).

3.2.1 ZSL Classifier Training with Generated Samples

In this second stage, we feed the trained conditional generator, $f_{att.}$ with unseen class attributes (*i.e.*, inputs from $\tilde{\mathcal{D}}_{att.}$). The idea here is to generate features that represent unseen class instances, assuming that the learned conditional generator, $f_{att.}$ is able to generalize well to unseen classes. Note that such a generalization assumption is extensively used in ZSL literature [7, 1, 22, 30]. Thereafter, we use the generated features together with real seen class features (*i.e.*, outputs from f_s for real seen class instances) to train a new feature classifier, $h^* : \mathbb{R}^d \rightarrow \mathbb{R}^{|\mathcal{C}|+|\tilde{\mathcal{C}}|}$[3]. For evaluation, we use the model $h^* \circ f_s$ to classify test instances.

4. Related Work

In this section, we first discuss related UDA and ZSL solutions. Thereafter, we discuss deep algorithms that are based on the min-max optimization framework.

Unsupervised Domain Adaptation: Our proposal can be employed to address UDA by statistical alignment. Statistical alignment of domains is a fundamental solution for UDA [25, 14, 26]. Focusing on UDA methods, the closest work to ours is the Deep Correlation Alignment solution (D-CORAL) of Sun *et al*. [26]. Apart from the fundamental difference in the formulation, *i.e.*, min-max in our case in comparison to min in D-CORAL, the statistical disparity measure is different between the two solutions. Here, we use a symmetric KL-divergence while D-CORAL measures the disparity using the Frobenius norm.

Domain adversarial learning [9, 28, 24] uses GAN [11] principles for learning domain invariant features. Here, a feature extractor acts as the generator network of the GAN. Its objective is to outperform the ability of the domain discriminator to distinguish the domain of a given instance. As explained, our confusion network is somewhat similar in spirit to the discriminator network of domain adversarial solutions. However, the purpose of the confusion network is to maximize the statistical disparity.

Zero-Shot Learning: Learning a relationship between semantic descriptors and instance features is the core concept behind ZSL solutions. For instance, [7, 1, 22] learn a bilinear relationship between semantic attributes and instance features. Furthermore, Deep Auto-Encoders [17, 16], synthesizing classifiers [4], Kernel methods [32] are also among the machine learning tools that have been recently proposed for learning complex relationships between semantic descriptors and instance features.

In a different direction, Xian *et al*. [30] propose feature generation GANs for ZSL. Their objective is to first train a model that can generate features for seen classes given attribute vectors as inputs. Thereafter, it is assumed that this generator is generalizable for generating features for unseen classes. Our two-stage ZSL framework is inspired by this solution. However, in contrast we propose to use statistical alignment to learn the generative model. This is an unexplored path for ZSL. Furthermore, we explore the capacity of min-max alignment in this context.

Min-Max Learning: The Maximum-Mean Discrepancy [12] has been widely used to learn from distributions. Naturally, minimization of MMD will create a min-max problem. As such, Dziugaite *et al*. [6] propose to train a generative model by minimizing an MMD loss. However, the kernel used in MMD enabled the authors to avoid an ex-

[3]Note that we evaluate our ZSL on the Generalized ZSL protocol proposed in [31]

plicit maximization step. The MMD-GAN formulation of Li *et al*. [20] takes the idea further by learning the kernel for MMD along the way. Our proposal is different from the aforementioned MMD solutions in the sense that the confusion network by itself is a feature mapping. Furthermore and in contrast to MMD, our maximization is not a component of the statistical disparity computations.

We conclude this part by acknowledging the recent work of Shalev-Shwartz and Wexler [23]. There, the authors study the min-max framework in optimizing various forms of loss functions with a stress on classification problems. For example, the authors show that the min-max solution for binary classification problems with 0-1 loss enjoys a strong form of guarantee while the min counterpart does not. We believe that the theoretical insights provided in [23] strengthen our idea of developing the min-max alignment.

5. Experiments

In this section, we empirically contrast our min-max solution against various baselines on UDA and ZSL. Our Deep Neural Network (DNN) models are trained end-to-end with RMSProps optimizer with a batch size of 256. Randomly initialized models are trained with a learning rate of 0.001 and pre-trained AlexNet [18] models with a learning rate of 0.0001. To obtain stable covariance matrices, we use an additional dimensionality reduction layer between $fc7$ and $fc8$ layers of the AlexNet where the dimensionality is reduced to 256. Similar dimensionality reduction layers are used in prior work [9]. We fix the value of λ_u to 0.001 (*see* Eq. (8)) unless stated otherwise. Furthermore, we use an accumulation momentum of 0.5 (*i.e*., $m = 0.5$ in Eq. (6)). Our confusion model consists of a single convolution (if the input is a 2D-feature map) or fully connected layer with a residual skip connection. We also use a leaky-ReLU activation layer at the output of confusion network. More details of our DNN structures, training hyper-parameters, and impact of various confusion structures can be found in the supplementary material.

5.1. Experiments on UDA

We evaluate and assess our min-max solution on UDA with two sets of experiments. The first experiment is done with Office31 dataset using pre-trained AlexNet architecture. The second set of experiments is performed on MNIST, SVHN, SYN. DIGITS, GTSRB, SYN. SIGNS, STL and CIFAR datasets , where the CNN model is trained from scratch. We share model parameters between source and target domain networks during training (*see* Fig. 3(a)).

Focusing on the baselines, we denote the model trained from the source data (*i.e*., no adaptation is considered) as **CNN**. Other baselines include several state-of-the-art UDA methods such as **D-CORAL** [26], **DANN** [9] and **VADA** [24][4]. These baselines are the most relevant ones, idea-wise to our work. We also report results for the model that only minimizes the KL divergence between source and target domains (denoted by **Min**). We denote the proposed min-max solution by **Min-Max** when $\lambda_t = 0.0$ in Eq. (8). We also denote our min-max solution as **Min-Max+** when $\lambda_t = 0.1$.

In addition to Office31, we also experimented with six more domain adaptation tasks, namely, SVHN $\longleftrightarrow$ MNIST, SYN. DIGITS $\longrightarrow$ SVHN, SYN. SIGNS $\longrightarrow$ GTSRB and STL $\longleftrightarrow$ CIFAR[5], where we used the same protocol, data setup and network architecture as in [24] (see supplemantary of this paper for the details).

For the two experiment sets STL $\longleftrightarrow$ CIFAR, we used $\lambda_u = 0.0001$. The value of λ_u is 0.01 for all the remaining experiments (*i.e*., experiments with MNIST, SVHN, SYN. DIGITS, SYN. SIGNS, GTSRB).

In Table 1, we report the performance of both **Min-Max** and **Min.**. We observe that in all cases, the **Min-Max** method outperforms the **Min.** method. For instance, in MNIST$\rightarrow$SVHN, the **Min-Max** outperforms the **Min.** alignment by 36.1%. Interestingly, in some cases the **Min.** alignment is not able to improve upon the **CNN** baseline that does not benefit from any adaptations (*e.g*., $D \rightarrow W$ of Office31 dataset). However, our **Min-Max** method improves the results in all cases. Our method obtains an average improvement of 12.2% over baseline **CNN** and 5.4% over the baseline **Min.**. The improvement is a clear indication of a better statistical alignment, reinforcing our claim that the min-max solution (**Min-Max**) is a better alternative compared to aligning by minimizing the statistical disparities.

In Table 2, we compare our min-max solution with D-CORAL [26] and the adversarial methods DANN [9] and VADA [24]). For the Office31 experiments, we use reported results in [9], [10] and [26]. Results for remaining domain sets (*i.e*., MNIST, SVHN, SYN. DIGITS, SYN. SIGNS, GTSRB, STL, CIFAR) are obtained from Shu *et al*. [24] for DANN and VADA. For Office31 dataset, we do not observe a significant performance difference between **Min-Max** and **Min-Max+**. Hence, we use only **Min-Max** (*i.e*., $\lambda_t = 0.0$). However, for the remaining domain adaptation tasks, the discriminative loss on the target domain is helpful and **Min-Max+** method outperforms **Min-Max**.

In Office31 dataset, our solution is ranked among the top two performers in four instances out of six and the top performer in two instances. Furthermore, our solution outper-

[4]We acknowledge that Shu *et al*. [24] also propose a more involved algorithm, **dirt-t** as an improved version of **VADA**. Since our goal is to mainly contrast min-max alignment against the min solution, we did not benefit from advanced learning techniques for UDA. Hence, we believe VADA is a more appropriate baseline here.

[5]For the CIFAR and STL data, we consider the 9 shared classes (omiting "monkey" and "frog") as in [24].

Sol.	s	A	A	D	D	W	W	MNIST	SVHN	DIGITS	SIGNS	STL	CIFAR
	t	D	W	A	W	A	D	SVHN	MNIST	SVHN	GTSRB	CIFAR	STL
CNN		60.8	58.5	42.6	94.1	38.6	98.1	37.5	63.1	84.0	77.8	59.1	75.9
Min		66.1	70.5	47.2	93.5	47.6	98.6	49.8	70.6	85.4	78.7	57.7	74.4
Min-Max		**69.1**	**71.7**	**51.3**	**95.0**	**52.1**	**99.3**	**67.8**	**72.0**	**88.5**	**83.2**	**60.6**	**76.1**

Table 1. **Comparison of the Min-Max UDA solution with CNN and Min baselines.** For Office31 we use the "fully transductive protocol" as in [9, 26]. We use the acronyms A:Amazon, W:Webcam, and D: DSLR. For the remain domain sets (MNIST, SVHN, SYN. DIGTS, GTSRB, SYN. SIGNS, CIFAR, STL) we follow the training protocol and network models used in [24].The best performance is in **Bold**.

Sol.	s	A	A	D	D	W	W	MNIST	SVHN	DIGITS	SIGNS	STL	CIFAR
	t	D	W	A	W	A	D	SVHN	MNIST	SVHN	GTSRB	CIFAR	STL
DANN [9]		67.1	**73.0**	**54.5**	**96.4**	**52.7**	99.2	60.6	68.3	90.1	97.5	62.7	78.1
VADA [24]		-	-	-	-	-	-	73.3	94.5	**94.9**	**99.2**	**71.4**	78.3
D-CORAL [26]		66.8	66.4	52.8	95.7	51.5	99.2	72.7	87.8	71.8	59.9	60.5	76.2
Min-Max+		**69.1**	71.7	51.3	95.0	52.1	**99.3**	**79.3**	**97.0**	94.6	97.3	67.7	**79.9**

Table 2. **Comparison of the Min-Max+ UDA solution with related UDA solutions.** For Office31 we use the "fully transductive protocol" as in [9, 26]. We use the acronyms A:Amazon, W:Webcam, and D: DSLR. For the remain domain sets (MNIST, SVHN, SYN. DIGTS, GTSRB, SYN. SIGNS, CIFAR, STL) we follow the training protocol and network models used in [24]. The best performance is in **Bold** and second best is in Blue.

forms D-CORAL [26] in four instances on Office31 dataset. On the remaining domain adaptation tasks, our method outperforms D-CORAL by a considerable margin. Our **Min-Max+** method performs better than state-of-the-art adversarial methods such as DANN and VADA in three instances out of six (*i.e.*, MNIST↔SVHN and CIFAR→ STL).

Out of all the domain transformation tasks, the MNIST→SVHN task is the most difficult one to adapt. This is evident by the low performance in our baseline evaluations. However, our proposed solution shows a significant improvement for this particular UDA task.

5.2. Experiments on Zero-Shot Learning

We compare our method with recent ZSL methods on two fine-grained image classification datasets, namely Caltech-UCSD-Birds [29] (CUB) and SUN dataset [21] and two coarse grained datasets (Awa1 and Awa2 [19]) following the Generalized ZSL (GZSL) protocol of [31]. We compute average per-class accuracy for the seen and unseen classes on the test instances which are denoted by S and U, respectively. The model performance is obtained through harmonic mean (denoted by HM) between U and S (*i.e.* $HM = 2 \times (S \times U)/(S + U)$). The GZSL protocol evaluates the performance on both seen and unseen samples with the same classifier (*i.e.*, the search space includes both seen and unseen classes). However, low performance either in seen classes or the unseen classes will eventually tend to give a low value for HM. We use the Resnet-101 [13] network, pre-trained on ImageNet dataset [5], to extract features. We use the train-test splits provided by [31] for all these datasets. These splits are suitable for this experiment as all unseen classes do not contain any overlap with the ImageNet classes.

We implement our ZSL model components using fully-connected neural networks, f_s : in $\rightarrow$ $fc(n)$ $\rightarrow$ out, $f_{att.}$: in $\rightarrow$ $fc(n)$ $\rightarrow$ noise $\rightarrow$ $fc(512)$ $\rightarrow$ noise $\rightarrow$ $fc(n)$ $\rightarrow$ noise $\rightarrow$ $fc(n)$ $\rightarrow$ out. The confusion network, g is a single layer network, $fc(n)$ with a residual skip connection from the input to the output. Here, $fc(n)$ represents a fully-connected layer with n outputs and "noise" represents a dropout layer followed by an additive Gaussian noise layer. We use $n = 1024$ for the Sun dataset experiments and $n = 512$ for the remaining sets[6]. We set the value of λ_u and m to 0.1. Furthermore, we start training the models f_s and h earlier than $f_{att.}$. This is to make sure that $f_{att.}$ receives an informative gradient from the classifier, h.

Mini-batch creation for GZSL. We trained the classifier (*see* section 3.2.1) with mini-batches containing a mixture of real seen classes and generated unseen classes. To improve our classifier's robustness to unseen classes, every mini-batch used 2× more generated samples per-class as that of the real instances except for the case Awa1 for which we report on 20×. For Awa1 we observe a significant improvement can be achieved by using higher generated sample proportions. Further analysis on this can be found in our supplementary.

Results for zero-shot-learning using our method in shown in Table. 3. As a baseline, we also report results for statistical alignment by minimization (**Min.**) All the baseline methods use Resnet-101 features provided in [31]. We observe that our proposed min-max alignment outperforms the Min. baseline in all four experiments. For completeness, we also performed an experiment where a correla-

[6]More details about model structure can be found in supplementary material

Data.	Awa1			Awa2			Cubs			Sun		
Sol.	U	S	HM	U	S	HM	U	S	HM	U	S	HM
SAE [17]	1.8	77.1	3.5	1.1	82.2	2.2	7.8	54.0	13.6	8.8	18.0	11.8
ZKL [32]	18.3	79.3	29.7	18.9	82.7	30.8	24.2	63.9	35.1	21.0	31.0	25.1
Cls. Prot. [15]	28.1	73.5	40.6	-	-	-	23.5	55.2	32.9	21.5	34.7	26.5
CLSW [30]	57.9	61.4	59.6	-	-	-	43.7	57.7	49.7	42.6	36.6	**39.4**
Min	46.0	83.3	59.3	32.9	89.7	48.1	46.1	50.8	48.3	38.8	35.0	36.8
Min-Max	46.6	84.2	**60.0**	37.8	88.8	**53.0**	47.1	53.8	**50.2**	37.9	36.5	37.2

Table 3. **Comparison of the proposed ZSL solution (Min-Max) on GZSL protocol [31].** We report the average per-class accuracy on seen class test instance as "S". The unseen class performance is reported as "U". The harmonic mean of "U" and "S" are reported as "HM". The best performance is in **Bold** and second best is in Blue.

tion loss [26] is used instead of the KL divergence in Min. However, this model did not show competitive performance (*e.g.*, the HM results for Awa1:26.1%, Awa2:19.6%).

We also compare our solution with the reported results on the recent ZSL solutions. The proposed min-max alignment outperforms the CLSW [30], which uses a powerful Wasserstein GAN [2]) in the Cubs dataset by a large margin. Overall, the proposed min-max alignment method reaches competitive performances with state-of-the-art ZSL methods, yet again indicating the effectiveness of proposed alignment method.

Further Improvement by End-to-end Fine-tuning. We followed the two-staged framework of CLSW [30] closely but our model can be fine-tuned in an end-to-end manner. With end-to-end fine-tuning, we observed a consistent improvement $< 1\%$ over the results reported in Table 3. However and interestingly for Awa1 we observe our min-max solution achieves a **HM of** 61.3% through this.

5.3. Effect of Moment Accumulation

Lastly, we discuss the effect of momentum accumulation (*see* § 2.2) for our solution. In Fig. 4, we report the performance of the proposed min-max (denoted as **Min-Max**) and minimization (**Min**) methods for various values of the momentum, m (*see* Eq. (6)). Here, we consider one UDA (Webcam $\rightarrow$ Amazon) dataset and one ZSL dataset (Cub). Studying Fig. 4 shows that the accumulation helps in both cases. Overall and inline with previous results, we observe that the min-max solution performs better than the min one. For the ZSL experiment, we find out that the accumulation is essential. We conjecture that this is due to the stability, accumulation can bring in to the min-max learning.

6. Conclusion

In this paper, we proposed min-max statistical alignment for learning domain invariant features. To realize our min-max solution as an end-to-end trainable deep model, we proposed a novel structure, "the confusion network". Our confusion network behaves similarly, in spirit, to the discriminator of GANs. However, the proposed confusion network attempts to learn a function that maximizes the statistical disparity between the two domains. We showed that the performance of the proposed min-max statistical alignment can be improved by accumulation of mini-batch statistics. Furthermore, an important theoretical property of the min-max solution is established in Theorem 1.

We evaluated our proposed min-max solution on two transfer learning case- studies, namely, Unsupervised Domain Adaptation (UDA) and Zero-Shot Learning (ZSL). Our UDA model used statistical alignment to train an invariant feature extractor for source and target domains. For ZSL, we used the statistical alignment to train a feature generator for unseen classes. In our evaluations, the min-max solution consistently outperformed statistical alignment by minimization. Interestingly, we also showed that with the proposed min-max solution we could even reach comparable results with the state-of-the-art solutions using GAN's principles.

Extending the proposed min-max learning to other statistical disparities (*e.g.*, Wasserstein alignment) is our future plan. Furthermore, we intend to explore various forms of confusion networks to design generative models for complex data (*e.g.*, images) and present a dedicated theoretical study about confusion network properties in future.

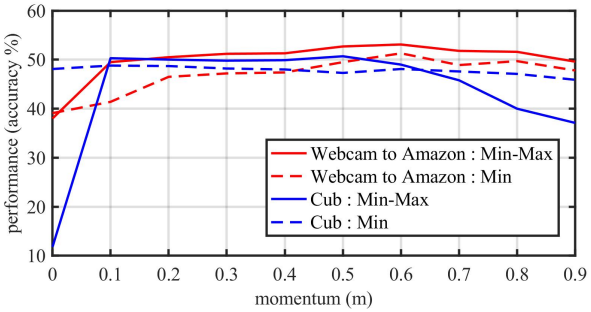

Figure 4. Effect of moment accumulation (*see* § 2.2) for the proposed statistical alignment.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, 2016. 5

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 8

[3] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992. 1

[4] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5327–5336, 2016. 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 7

[6] GK Dziugaite, DM Roy, and Z Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Uncertainty in Artificial Intelligence-Proceedings of the 31st Conference, UAI 2015*, pages 258–267, 2015. 5

[7] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 2121–2129, 2013. 5

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 1, 3, 4

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 5, 6, 7

[10] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 6

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. 3, 5

[12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 5

[13] 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. 7

[14] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Learning an invariant hilbert space for domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 5

[15] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 8

[16] Elyor Kodirov. *Cross-class Transfer Learning for Visual Data*. PhD thesis, Queen Mary University of London, 2017. 5

[17] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3174–3183, 2017. 5, 8

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 6

[19] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 7

[20] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 2203–2213, 2017. 1, 6

[21] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2751–2758. IEEE, 2012. 7

[22] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *Proc. Int. Conference on Machine Learning (ICML)*, pages 2152–2161, 2015. 5

[23] Shai Shalev-Shwartz and Yonatan Wexler. Minimizing the maximal loss: how and why. In *Proc. Int. Conference on Machine Learning (ICML)*, pages 793–801, 2016. 1, 6

[24] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 1, 4, 5, 6, 7

[25] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1, 5

[26] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer, 2017. 1, 5, 6, 7, 8

[27] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops*, pages 443–450. Springer, 2016. 1, 2, 4

[28] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971. IEEE, 2017. 5

[29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 7

[30] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 4, 5, 8

[31] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3077–3086, 2017. 1, 4, 5, 7, 8

[32] Hongguang Zhang and Piotr Koniusz. Zero-shot kernel learning. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 8

Supplementary Material : Min-Max Statistical Alignment for Transfer Learning

Samitha Herath[1,4], Mehrtash Harandi[2,4], Basura Fernando[1,5], and Richard Nock[1,3,4]
[1]The Australian National University, [2]Monash University, [3]The University of Sydney
[4]DATA61-CSIRO, Australia
[5]Human-Centric AI Programme, A*STAR, Singapore
Samitha.Herath@data61.csiro.au, Mehrtash.Harandi@monash.edu,
Fernando_Basura@scei.a-star.edu.sg, Richard.Nock@data61.csiro.au

We start this supplementary by providing the proof of **Theorem 1** from the main text. We repeat the theorem below for the convenience of the reader.

Theorem 1. *If the confusion function, g is a linear invertible transformation, $\boldsymbol{Q}$ with $\boldsymbol{Q}\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{-1}\boldsymbol{Q} = \boldsymbol{I} \in \mathbb{R}^{d\times d}$ then the proposed min-max statistical alignment by confusion is equivalent to statistical alignment by minimization. Here, d is the dimensionality of both the domain input features.*

Proof. Let, $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ be the covariance matrices for the domains $\mathcal{D}_0$ and $\mathcal{D}_1$,respectively. Similarly, consider the domain means to be $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$. Furthermore, let the feature confusion function, g be a linear transformation given by a square invertible matrix, $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$. Hence, the confused feature means $\tilde{\boldsymbol{\mu}}_0$ and $\tilde{\boldsymbol{\mu}}_1$ for the two domains can be written as,

$$\tilde{\boldsymbol{\mu}}_0 = \boldsymbol{Q}^T\boldsymbol{\mu}_0, \text{and } \tilde{\boldsymbol{\mu}}_1 = \boldsymbol{Q}^T\boldsymbol{\mu}_1 \tag{1}$$

Furthermore, the confused feature covariances $\tilde{\boldsymbol{\Sigma}}_0$ and $\tilde{\boldsymbol{\Sigma}}_1$ are given by,

$$\tilde{\boldsymbol{\Sigma}}_0 = \boldsymbol{Q}^T\boldsymbol{\Sigma}_0\boldsymbol{Q}, \text{ and } \tilde{\boldsymbol{\Sigma}}_1 = \boldsymbol{Q}^T\boldsymbol{\Sigma}_1\boldsymbol{Q}. \tag{2}$$

From this, the confused feature KL divergence , $D_{KL}(\tilde{P}_0\|\tilde{P}_0)$ can be computed as,

$$\begin{aligned}
&D_{KL}(\tilde{P}_0\|\tilde{P}_0) = \\
&\quad\frac{1}{2}\big(\mathrm{tr}\big(\tilde{\boldsymbol{\Sigma}}_1^{-1}\tilde{\boldsymbol{\Sigma}}_0\big) + \big(\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_0\big)^T\tilde{\boldsymbol{\Sigma}}_1^{-1}\big(\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_0\big) \\
&\quad + \log\left(\frac{\det\tilde{\boldsymbol{\Sigma}}_1}{\det\tilde{\boldsymbol{\Sigma}}_0}\right) - d\big) \\
&=\frac{1}{2}\big(\mathrm{tr}\big((\boldsymbol{Q}^T\boldsymbol{\Sigma}_1\boldsymbol{Q})^{-1}\boldsymbol{Q}^T\boldsymbol{\Sigma}_0\boldsymbol{Q}\big) \\
&\quad + \big(\boldsymbol{Q}^T\tilde{\boldsymbol{\mu}}_1 - \boldsymbol{Q}^T\tilde{\boldsymbol{\mu}}_0\big)^T\big(\boldsymbol{Q}^T\tilde{\boldsymbol{\Sigma}}_1\boldsymbol{Q}\big)^{-1}\big(\boldsymbol{Q}^T\tilde{\boldsymbol{\mu}}_1 - \boldsymbol{Q}^T\tilde{\boldsymbol{\mu}}_0\big) \\
&\quad + \log\left(\frac{\det\big(\boldsymbol{Q}^T\boldsymbol{\Sigma}_1\boldsymbol{Q}\big)}{\det\big(\boldsymbol{Q}^T\boldsymbol{\Sigma}_0\boldsymbol{Q}\big)}\right) - d\big) \\
&=\frac{1}{2}\big(\mathrm{tr}\big(\boldsymbol{Q}^{-1}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{Q}^{-T}\boldsymbol{Q}^T\boldsymbol{\Sigma}_0\boldsymbol{Q}\big) \\
&\quad + \big(\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_0\big)^T\boldsymbol{Q}\boldsymbol{Q}^{-1}\tilde{\boldsymbol{\Sigma}}_1^{-1}\boldsymbol{Q}^{-T}\boldsymbol{Q}^T\big(\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_0\big) \\
&\quad + \log\left(\frac{\det\big(\boldsymbol{Q}^T\big)\det\big(\boldsymbol{\Sigma}_1\big)\det\big(\boldsymbol{Q}\big)}{\det\big(\boldsymbol{Q}^T\big)\det\big(\boldsymbol{\Sigma}_0\big)\det\big(\boldsymbol{Q}\big)}\right) - d\big) \\
&=\frac{1}{2}\big(\mathrm{tr}\big(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\big) + \big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\big)^T\boldsymbol{\Sigma}_1^{-1}\big(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\big) \\
&\quad + \log\left(\frac{\det\big(\boldsymbol{\Sigma}_1\big)}{\det\big(\boldsymbol{\Sigma}_0\big)}\right) - d\big) \\
&=D_{KL}(P_0\|P_0)
\end{aligned} \tag{3}$$

■

1. Network structures

Here we provide details of our network models. For our UDA experiments on Office31 dataset, we use the pretrained AlexNet [7] on ImageNet [1] and fine-tuned it accordingly. To obtain stable covariance matrices, we used an additional dimensionality reduction layer between $fc7$ and $fc8$ layers of the AlexNet where the dimensionality is reduced to 256. A similar modification to the AlexNet is used in prior work[2] for Office31 experiments. Table 1 provides details of the structure of the network used in UDA experiments on MNIST, SVHN, SYN. DIGITS, SYN. SIGNS, GTSRB, STL and CIFAR datasets. In Table 2 the details of the network models (f_1 and g) used in our toy data experiment (*see* Fig.1 in the main paper) are listed. Note that the feature extractor f_0 is an identity mapping for this experiment. Lastly, in Table 3, we report the details of the deep network models used in ZSL experiments.

Feature extractor (f_s, f_t)	
1	32 x 32 x 3 Image
2	Instance Normalization
3	3 x 3 conv, 64 (96) filters, leaky-ReLU
4	3 x 3 conv, 64 (96) filters, leaky-ReLU
5	3 x 3 conv, 64 (96) filters, leaky-ReLU
6	2 x 2 max-pool, stride 2
7	dropout, keep prob. = 0.5
8	Gaussian noise
9	3 x 3 conv, 64 (192) filters, leaky-ReLU
10	3 x 3 conv, 64 (192) filters, leaky-ReLU
11	3 x 3 conv, 64 (192) filters, leaky-ReLU
12	2 x 2 max-pool, stride 2
13	dropout, keep prob. = 0.5
14	Gaussian noise
Classifier Network (h)	
1	Input features
2	3 x 3 conv, 64 (192) filters, leaky-ReLU
3	3 x 3 conv, 64 (192) filters, leaky-ReLU
4	3 x 3 conv, 64 (192) filters, leaky-ReLU
5	global average pooling
6	10 (9) - fully connected
Confusion Network (g) - Residual Model	
1	Input features
2	3 x 3 conv, 64 (192) filters
3	leaky-ReLU
4	global average pooling

Table 1. The network structures for the UDA experiments on MNSIT, SVHN, SYN. DIGITS, SYN. SIGNS, GTSRB, STL and CIFAR. The values within parenthesis are used for the domain transformations STL↔CIFAR. All leaky-ReLU layers use 0.1 as the negative scale factor. The additive Gaussian noise layers use zero mean and a standard deviation of 0.0001. We use Batch-normalization for all convolutional layers. As in Shu *et al*. [8], we use Instance Normalization [10] at the input. Note that unlike the Office31 experiments, the outputs of f_s and f_t are 8×8 feature maps. Therefore, we use a convolutional network as the confusion network, g.

Feature extractor (f_1)	
1	2 - dimensional input features
2	2 - fully connected, Tanh
3	2 - fully connected, Tanh
Confusion Network (g)	
1	2 - dimensional input features
2	16 - fully connected, leaky - ReLU

Table 2. The network structures for the toy data experiment. Note that f_0 is an identity mapping. All leaky-ReLU layers use 0.1 as the negative scale factor.

Feature extractor (f_s)	
1	2048 dimensional features
2	512 (1024) - fully connected, leaky-ReLU
Feature Generator Network ($f_{att.}$)	
1	class attribute vectors
2	512 (1024) - fully connected, leaky-ReLU
3	dropout, keep prob. = 0.5
4	Gaussian noise
5	512 - fully connected, leaky-ReLU
6	dropout, keep prob. = 0.5
7	Gaussian noise
8	512 (1024) - fully connected, leaky-ReLU
9	dropout, keep prob. = 0.5
10	Gaussian noise
11	512 (1024) - fully connected, leaky-ReLU
Feature Classifier Network (h)	
1	512 (1024) dimensional features
2	number of classes - fully connected
Confusion Network (g) - Residual Model	
1	512 (1024) dimensional features
2	512 (1024) - fully connected, leaky-ReLU

Table 3. The network structures for the ZSL experiments. The values within parenthesis are used for the Sun dataset. All leaky-ReLU layers use 0.1 as the negative scale factor. The additive Gaussian noise layers use zero mean and a standard deviation of 1.0. We use Batch-normalization for all layers of the generator network (*i.e.*, $f_{att.}$).

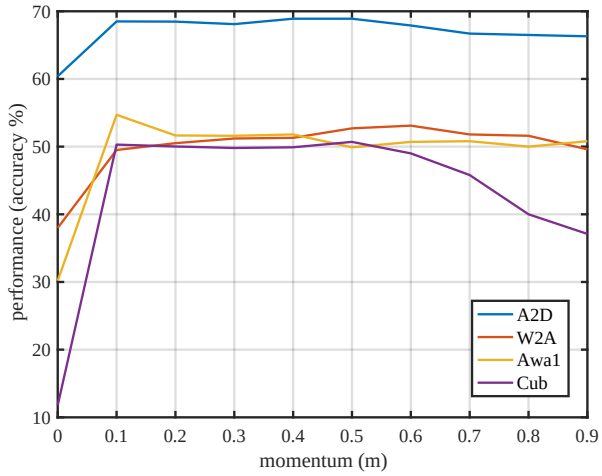

Figure 1. The robustness of the proposed min-max solution with respect to variations of the statistical accumulation momentum, m.

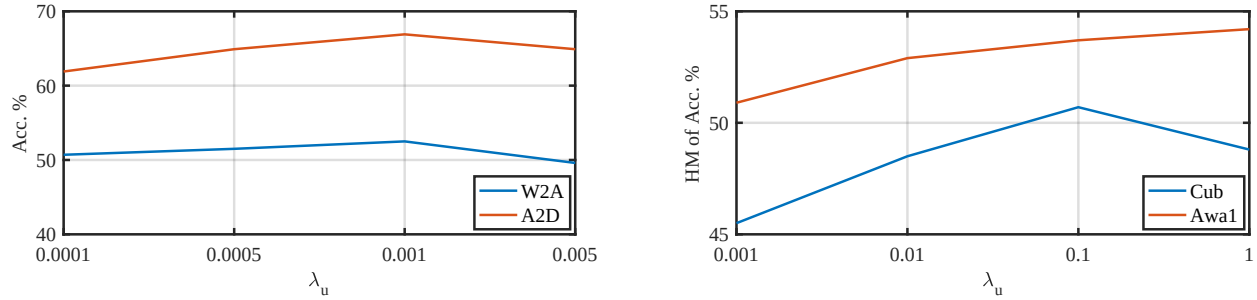

Figure 2. The robustness of the proposed min-max solution for the λ_u parameter in the loss function (*see* equations 8 and 11 in the main paper) for two UDA sets (Left) and two ZSL sets (Right).

2. Moment accumulation

In our experiments, we used statistical accumulation with $m = 0.5$ for UDA and $m = 0.1$ for ZSL. In Fig. 1, we report the robustness of the proposed min-max solution. For this experiment, we select two UDA experiment sets and two ZSL experiment sets. We observe that in all cases, moment accumulation is beneficial for statistical alignment. Furthermore, the performance of the solutions stays consistent for a wide range of momentum values (*i.e.*, $0.1 \leq m \leq 0.6$). We observe that in the case of ZSL, the moment accumulation is essential. We conjecture that this is due to the stability accumulation can bring in to the min-max learning.

3. Statistical loss weight λ_u

In Fig. 2, we report effects of changing the statistical loss weight, *i.e.*, λ_u in Eq. 8 and Eq. 11 of the main text. For this experiment, we select two UDA sets with $\lambda_u \in \{0.0001, 0.0005, 0.001, 0.005\}$ and two ZSL experiment sets with $\lambda_u \in \{0.001, 0.01, 0.1, 1\}$. In both experiments, consistent performance can be attained in a wide range of values for λ_u, suggesting robustness and easy-tuning. We observe that relatively smaller values of λ_u are useful for the UDA sets in comparison to the ZSL sets.

4. Mini-batch creation for GZSL

We train the final classifier (*see* § 3.2.1 of the main paper) with mini-batches containing a mixture of real seen classes instances and generated unseen class instances. To improve our classifier's robustness to unseen classes, we include generated

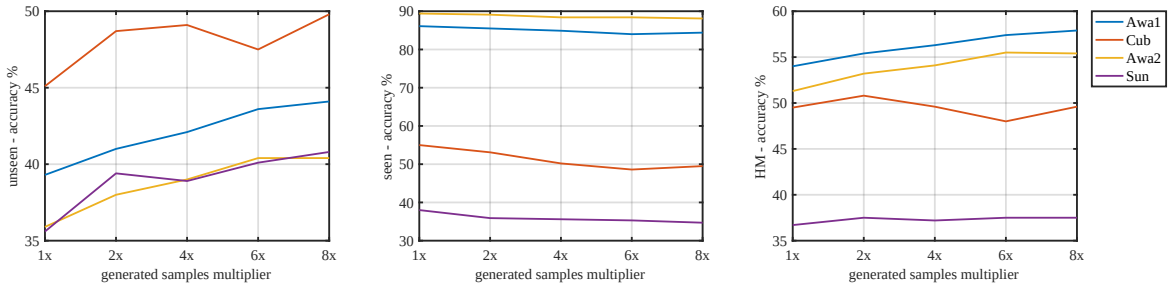

Figure 3. The performance of the proposed ZSL solution with different proportions of generated class instances.

class instances as 2x the number of real instance for each mini-batch in our reported experiments (*see* Table 3 in the main paper). In Fig. 3, we report the performance of our solution when the number of unseen class instances are increased by 2x, 4x, 6x and 8x. Here, we observe a consistent improvement in performance, except for Cubs experiment, by increasing the number of generated samples in mini-batches. Furthermore, we observe that the unseen class performance can be significantly improved by increasing the proportion of generated unseen class training samples.

5. Confusion network design

In all our reported experiments, we used confusion networks with a residual skip connection. In this section we provide further insights into our proposed confusion network design by comparing the performance of several network structures (*see* Fig. 4). In Table 4 we report the performance of different confusion networks for two domain sets from Office31 dataset and two ZSL datasets. For comparison, we also included the performance of the minimization solution (*i.e.*, Min).

Change in dimensionality. We test three input/output configurations of the confusion network. Namely, **1.** Equal-dim-confusion(EDC, *see* Fig. 4(a)), where the input and output dimensions are equal, **2.** Low-dim-confusion (LDC, *see* Fig. 4(b)), where the output dimension is lower than the input , and **3.** High-dim-confusion (HDC, *see* Fig. 4(c)) where the output dimension is higher than the input. For the two UDA sets, the output dimensions of the LDC, EDC and HDC models are 128, 256 and 512, respectively. For the two ZSL sets they are 384, 512 and 768, respectively. We observe that the EDC and HDC structures perform better than the LDC structure. This is intuitive as going down with the dimension might lead to loss of information. Overall, we observe that the EDC structure is marginally better than the HDC network.

Use of Batch Normalization. For the network Confusion-with-Batch-Norm (BNC, *see* Fig. 4(d)), we use an additional Batch Normalization layer after the EDC model. In our preliminary experiments without statistical accumulation, we observed that confusion networks performing marginally better with Batch-Normalization. By design, the Batch Normalization layers attempt to keep a stable output feature distribution. This stability can be a favorable condition for statistical alignment. However, when statistical accumulation is used we observe that the proposed EDC model outperforms the BNC network (*see* Table 4). We consider this to be an indication that the proposed statistical accumulation is a better regularizer for the proposed statistical alignment.

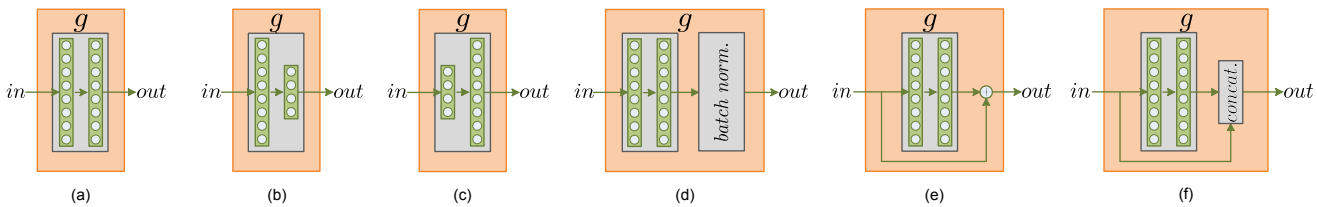

Figure 4. Schematic diagrams of the confusion networks.

Experiment	Min	EDC	LDC	HDC	BNC	Re.C	De.C
A2D	65.5	69.3	64.1	67.3	66.5	68.3	69.7
W2A	48.4	53.9	49.5	52.0	43.3	52.7	50.0
Awa1	53.1	51.9	47.8	52.4	23.4	54.5	40.8
Cub	48.8	50.0	44.8	50.3	22.5	50.6	17.4

Table 4. The performance for various confusion networks structures (Equal-dim-confusion (EDC), Low-dim-confusion (LDC), High-dim-confusion (HDC), Confusion-with-Batch-Norm. (BNC), Residual-confusion (Re.C), and Dense-confusion (De.C)) given in Fig. 4. For comparison, we also include the performance of the KL-minimization.

Solution	Awa1	Awa2	Cubs	Sun
SAE [6]	53.0	54.1	33.3	40.3
ZKL [14]	70.1	70.5	51.7	61.7
Cls. Prot. [5]	69.9	-	54.3	63.3
CLSW [12]	68.2	-	57.3	60.8
Min	61.7	62.3	52.8	54.2
Min-Max	64.1	69.5	56.6	58.8

Table 5. Comparison of the proposed ZSL solution (Min-Max) on conventional ZSL for the proposed splits in [13]. Although our solution does not outperform state-of-the-art solutions in this conventional protocol, we observe that the proposed min-max consistently outperforms the minimization solution.

Use of Skip Connections. Here we explore two skip connection based confusion network structures. Namely, **1.** Residual-confusion (Re.C) (*see* Fig. 4(e)) and **2.** Dense-confusion (De.C) (*see* Fig. 4(f)). Similar to ResNet [3], we perform an addition of the skip connection for the Re.C model. We use a concatenation of the skip connection for De.C in the spirit of the DenseNet [4] and the Inception module [9]. In comparison to our EDC and De.C networks, we observe a stable performance improvement from the Re.C model. Therefore, for all our reported results we use Re.C confusion network.

6. Convergence of the min-max and stopping criteria

We report the performance of min and min-max solutions after a fixed number of training iterations. In Fig. 5, we show the min-max alignment training loss for a UDA and a ZSL set. Similar convergence curves are observed in other experiments.

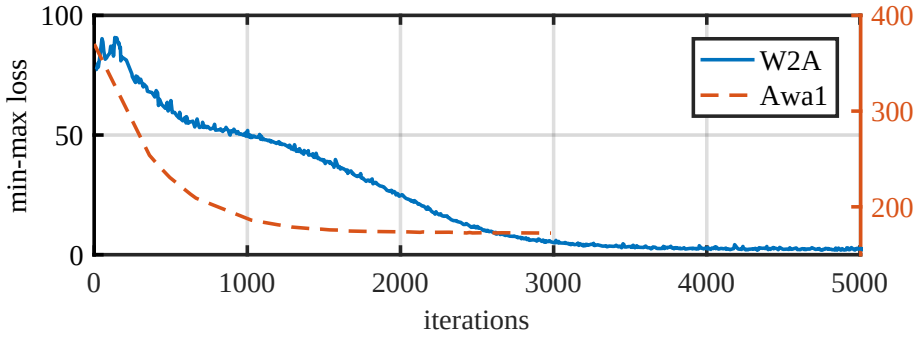

Figure 5. Convergence of min-max training for the W2A and Awa1 experiment sets.

7. Conventional ZSL Performance

It is our understanding that the generalized protocol (*i.e.*, GZSL [13, 11]) is more recent and represents a more challenging problem than the conventional ZSL. That said, we evaluated the min-max and min solution on conventional ZSL(*see* Table 5). We achieve competitive result on the CUBs and Awa2 datasets (56.9% and 69.5%, respectively). As a pointer, the min solution achieves 52.8% and 62.3% for these datasets, respectively. On the Awa1, the min-max solution, outperforms the min solution by a large gap but comes short in comparison to the SOTA ZSL solutions (our result reads as 64.1%).

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 2

[2] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 2

[3] 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. 5

[4] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2017. 5

[5] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 5

[6] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3174–3183, 2017. 5

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 2

[8] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 2

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 5

[10] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2

[11] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2018. 5

[12] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[13] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3077–3086, 2017. 5

[14] Hongguang Zhang and Piotr Koniusz. Zero-shot kernel learning. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5