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 minimizing 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 Kullback-Leibler (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.
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., $X$) and bold lower case letters show column vectors (e.g., $x$). We represent sets with {...} and their cardinality with $|\cdot|$. 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 $D$ to represent a domain (e.g., $D_0$ for domain 0).

For simplicity, we also use this notation $D$ when referring to the data samples from a domain (e.g., $x_i^{(0)} \in D_0$, for $i = 1, 2, 3, \ldots, N_0$ to denote the samples with $y_i^{(0)} \in D_0$ being their labels from domain 0.). We consider the dimensionality of samples from a domain $D_k$ to be $n_k$ (e.g., $x_i^{(0)} \in \mathbb{R}^{n_0}$).

Our objective is to learn two non-linear mappings, $f_k(\cdot, \theta_k) : \mathbb{R}^{n_k} \rightarrow \mathbb{R}^d$, $k \in \{0, 1\}$ to embed samples from domains $D_k$, $k \in \{0, 1\}$ to a shared feature space such that they are statistically aligned. The non-linear mappings are parametrized by $\theta_0$ and $\theta_1$ 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, $\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} f_k(x_i^{(k)})$ and the covariance, $\Sigma_k = \frac{1}{N_k-1} \sum_{i=1}^{N_k} (f_k(x_i^{(k)}) - \mu_k)(f_k(x_i^{(k)}) - \mu_k)^T$ the statistical misalignment loss, $L_u$ between $D_0$ and $D_1$ can be expressed using symmetric KL-divergence as,

$$L_u = \frac{1}{2} (D_{KL}(P_0||P_1) + D_{KL}(P_1||P_0)).$$

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

$$D_{KL}(P_0||P_1) = \frac{1}{2} \left( \text{tr}(\Sigma_1^{-1}\Sigma_0) + \log \left( \frac{\det \Sigma_1}{\det \Sigma_0} \right) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - d \right).$$

In realizing min-max alignment we propose to make use of an additional mapping (i.e., the confusion network), $g(\cdot, \theta_g) : \mathbb{R}^d \rightarrow \mathbb{R}^p$. The inputs to the confusion network are the domain features from the functions $f_0(\cdot, \theta_0)$ and $f_1(\cdot, \theta_1)$. We refer to the output features of $g(\cdot, \theta_g)$ as the confused features. We implement the confusion network, $g(\cdot, \theta_g)$ using a neural network parameterized by $\theta_g$. Thereafter, we propose to perform the min-max alignment by optimizing,

$$\min_{\theta_0, \theta_1, \theta_g} \max \tilde{L}_u,$$

with

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

Figure 1. Toy data demonstration (Should be viewed in color). A comparison of aligning the input data, $D_1$ (i.e., Gaussian noise) to synthetic target data, $D_0$ (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 $D_0$ 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.
However, distinct to the defined statistical loss in equation (1), the feature distributions, \( \hat{P}_k, k \in \{0, 1\} \) are parameterized with,

\[
\hat{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} g \circ f_k(x_i^{(k)}), \tag{5}
\]

\[
\hat{\Sigma}_k = \frac{1}{N_k - 1} \sum_{i=1}^{N_k} \left( g \circ f_k(x_i^{(k)}) - \hat{\mu}_k \right) \left( g \circ f_k(x_i^{(k)}) - \hat{\mu}_k \right)^\top.
\]

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, \( Q \) with \( QQ^{-1} = Q^{-1}Q = I \in \mathbb{R}^{d \times d} \) 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. \( \blacksquare \)

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-
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, $D_s$, are used to train a classifier to classify unlabeled samples from the target domain, $D_t$. 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 source and target domains respectively.

It is typical to use a two-stream network in deep UDA where each stream corresponds to a specific domain (i.e., either $D_s$ or $D_t$). 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, $h(\cdot, \theta_h) : \mathbb{R}^d \rightarrow \mathbb{R}^c$ represents a shared classifier with parameters, $\theta_h$. The source and target domain feature extraction models, $f_s$ and $f_t$ are parameterized with $\theta_s$ and $\theta_t$, respectively (see Fig. 3(a) for a schematic).

For UDA, our objective is to jointly learn the shared feature space and the classifier $h(\cdot, \theta_h)$. 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

$$\min_{\theta_s, \theta_t} \max_{\theta_h} L_{d,s} + \lambda_t L_{d,t} + \lambda_u \tilde{L}_u.$$  (8)

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

$$L_{d,t} = -\mathbb{E}_{x \sim D_t}[h_t(x)^T \log h_t(x)],$$  (9)

computed from unlabeled target domain samples as in [24]. We denote the proposed statistical alignment loss with $\tilde{L}_u$. The parameter $\lambda_u$ and $\lambda_t$ 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\(^1\). We define the domain, $D_s$, to contain the labeled training instances from classes in $\mathcal{C}$. For training, we are provided with semantic descriptions for all the classes, $\mathcal{C} \cup \tilde{\mathcal{C}}$. 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 $D_{\text{att.}}$ and $D_{\text{att.}}$ 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, $f_{\text{att.}}(\cdot, \theta_{\text{att.}}) : \mathbb{R}^{n_{\text{att.}}} \rightarrow \mathbb{R}^d$ with $n_{\text{att.}}$ 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 $D_0 \sim D_s$ and $D_1 \sim D_{\text{att.}}$, per notations used in § 2. Later, we use $f_{\text{att.}}$ 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 $h_{\text{att.}} = \text{softmax} \circ h \circ f_{\text{att.}}$ (see Fig. 3 for a schematic). Here, $f_s(\cdot, \theta_s) : \mathbb{R}^c \rightarrow \mathbb{R}^d$ is a feature extraction network for real seen class instances. The function $h(\cdot, \theta_h) : \mathbb{R}^d \rightarrow \mathbb{R}^{c_{\text{att.}}}$ is a shared classifier for real and generated seen class features. We parameterize $h$, $f_{\text{att.}}$, and $f_s$ with $\theta_h$, $\theta_{\text{att.}}$, and $\theta_s$, respectively. Thereafter, we learn our generator network by following the two optimizations,

$$\min_{\theta_s, \theta_t} L_{d,s},$$  (10)

$$\min_{\theta_{\text{att.}}} \max_{\theta_u} L_{d,\text{att.}} + \lambda_u \tilde{L}_u.$$  (11)

\(^1\)Note the domain equivalences $D_0 \sim D_s$ and $D_1 \sim D_t$ with the discussion in § 2.

\(^2\)Note $\mathcal{C} \cap \tilde{\mathcal{C}} = \emptyset$
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 $\lambda_u$ is a training hyperparameter. The proposed statistical alignment losses between the domains $\mathcal{D}_s$ and $\mathcal{D}_{att.}$ is given by $\hat{\mathcal{L}}_u$ (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 $\mathcal{D}_{att.}$). 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, $h^*: \mathbb{R}^d \rightarrow \mathbb{R}^{|C|+|\bar{C}|}$

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-
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 $\lambda_u$ 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]. 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 $\leftrightarrow$ MNIST, SYN. DIGITS $\rightarrow$ SVHN, SYN. SIGNS $\rightarrow$ GTSRB and STL $\leftrightarrow$ CIFAR, where we used the same protocol, data setup and network architecture as in [24] (see supplementary of this paper for the details).

For the two experiment sets STL $\leftrightarrow$ CIFAR, we used $\lambda_u = 0.0001$. The value of $\lambda_u$ 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-
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. We set the value of $\lambda_s$ 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-

---

More details about model structure can be found in supplementary material
Data. & Awa1 & Awa2 & Cubs & Sun 
\hline
\hline
SAE [17] & 1.8 & 77.1 & 3.5 & 1.1 & 82.2 & 2.2 & 7.8 & 54.0 & 13.6 
ZKL [32] & 18.3 & 79.3 & 29.7 & 18.9 & 82.7 & 30.8 & 24.2 & 63.9 & 35.1 
Cls. Prot. [15] & 28.1 & 73.5 & 40.6 & - & - & - & 23.5 & 55.2 & 32.9 
CLSW [30] & 57.9 & 61.4 & 59.6 & - & - & - & 43.7 & 57.7 & 49.7 
Min & 46.0 & 83.3 & 59.3 & 32.9 & 89.7 & 48.1 & 46.1 & 50.8 & 48.3 
Min-Max & 46.6 & 84.2 & 60.0 & 37.8 & 88.8 & 53.0 & 47.1 & 53.8 & 50.2 
\hline
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.
References


