This ACCV 2020 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2020
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

Bridging Adversarial and Statistical Domain Transfer via Spectral Adaptation Networks

Christoph Raab[0000−0001−9921−2668], Philipp Väth, Peter Meier, and Frank-Michael Schleif[0000−0002−7539−1283]

University of Applied Sciences Würzburg-Schweinfurt, Würzburg, Germany
{christoph.raab,frank-michael.schleif}@fhws.de

Abstract. Statistical and adversarial adaptation are currently two extensive categories of neural network architectures in unsupervised deep domain adaptation. The latter has become the new standard due to its good theoretical foundation and empirical performance. However, there are two shortcomings. First, recent studies show that these approaches focus too much on easily transferable features and thus neglect important discriminative information. Second, adversarial networks are challenging to train. We addressed the first issue by the alignment of transferable spectral properties within an adversarial model to balance the focus between the easily transferable features and the necessary discriminatory features, while at the same time limiting the learning of domain-specific semantics by relevance considerations. Second, we stabilized the discriminator networks training procedure by Spectral Normalization employing the Lipschitz continuous gradients. We provide a theoretical and empirical evaluation of our improved approach and show its effectiveness in a performance study on standard benchmark data sets against various other state of the art methods.

1 Introduction

The ability to learn sophisticated functions and non-trivial data distributions are some of the main advantages of deep learning networks. In recent years, this capability has led to a drastic increase in classification accuracy in computer vision [1] and natural language processing [2], making them state of the art models in these fields. These flexible network architectures tend to overfit on the given training distribution while showing poor generalization on related distributions. Especially in real application scenarios, the training and test domains are different, and the networks cannot generalize well to the test distribution [3].

Unsupervised deep domain adaptation is a commonly utilized technique where fine-tuning of networks [4] is insufficient, due to missing test labels or significant differences between related domains [5]. During the training process, the networks learn discriminative features for the classification task and simultaneously learn an invariant representation by minimizing a statistical discrepancy between two or more domains [5, 3, 6]. Statistical adaptation [7] is usually integrated as a regularization term into the network. To some extent, these methods can be

interpreted as minimizing the discrepancy between one or more (higher) central moments of the domains [6]. The obtained representation should neglect source-specific domain characteristics such as light and camera settings in an image classification task.

However, statistical adaptation networks are naturally restricted in creating invariant features concerning the chosen discrepancy measure. In contrast, domain adversarial neural networks (DANN) [8] consist of a classifier network and a domain classifier on top of the feature extractor (bottleneck output). The learning process is a min-max game related to GANs [9]. The network feature extractor tries to fool the domain classifier (discriminator) by learning an adversarial representation expected to be invariant to the source and target domain. Supported by the domain adaptation theory [10], minimizing the domain classifier loss and reverse propagating the resulting gradient to the feature extractor facilitates learning a transferable representation.

Recent work [7] revealed that DANN type networks focus too much on easily transferable features associated with large singular values, neglecting discriminative features assigned to mid-size singular values. We derive the spectral alignment of both domains during learning, reducing the influence of large singular values while balancing the relevance of source and target domain. Therefore, we consider transferable and discriminative features of the source and target domain as sufficiently similar after learning. Hence, it is not necessary to reduce the influence of high singular values. Additionally, when striving for domain invariant representations, a minimization of domain-specific influences [11] of the primary learning domain, i. e., small source singular values, should take place.

To bridge the gap between statistical adaptation and adversarial enhancement in a single loss function, we propose the *Relevance Spectral Loss* (RSL) within our *Adversarial Spectral Adaptation Network* (ASAN). It aligns the spectrum of the source and target domain in the learning and adaptation process and simultaneously minimizes the influence of domain-specific features relative to the overall spectrum. The proposed RSL is related to moment-matching networks [3, 6, 5], due to the relationship of (squared) singular values to the variance and the second central momentum. Hence, minimizing RSL aligns not only discrepancies between spectral properties [7], i. e., transferability and discriminability, of the adversarial features, but also the statistical properties, i. e., second-order momentum. To obtain better control of the gradients from the domain classifier and to lower training difficulties of adversarial networks, we utilize the process by Spectral Normalization [12] of the discriminator weights. The ASAN model shows superior classification performance compared to state of the art methods.

The contributions of this paper are summarized in the following:

- Proposing the *Relevance Spectral Loss* (RSL), underlying reasoning of RSL and integration of Spectral Normalization [12] into our *ASAN* (Sec. 3 - 3.4).
- Theoretical evaluation of the gradient and the learning properties of the proposed *Adversarial Spectral Adaptation Network* (Sec. 3.5 - 3.6).
- Empirical evaluation on benchmark datasets against competitive networks and an analysis of its properties showing the efficiency of ASAN (Sec. 4).

2 Background and Related Work

In unsupervised deep domain adaptation [6, 8, 13, 3], we consider a labeled source dataset $D_s = \{\mathbf{X}_s, Y_s\} = \{\mathbf{x}_i, y_i\}_{i=1}^{n} \overset{i.i.d.}{\sim} p(\mathcal{S})$ in the source domain $\mathcal{S}$ and an unlabeled target dataset $D_t = \{\mathbf{X}_t\} = \{\mathbf{x}_j\}_{j=1}^{m} \overset{i.i.d.}{\sim} p(\mathcal{T})$ in the target domain $\mathcal{T}$ with same label space $\forall i, j : y_i, y_j \in \mathcal{Y}$ but different distributions $p(\mathcal{S}) \neq p(\mathcal{T})$. The overall goal is (still) to learn a classifier model, but additionally, it should generalize to a related target domain. The input feature space $\mathcal{X}$ is the initial representation of the source and target, i. e., $\mathbf{X}_s, \mathbf{X}_t \in \mathcal{X}$.

Initially, we consider a neural network $g : \mathcal{X} \to \mathcal{Y}$ with the parameters θ and given D_s, minimizing a classification loss - most often the cross-entropy $\mathcal{L}(g(\mathbf{x}; \theta), y) = -\sum_{i \in \mathcal{Y}} y_i log(g(\mathbf{x}_s; \theta)_i)$. The expected loss or risk of the network is $\mathcal{R}[\mathcal{L}(g(x; \theta), y)]$ and during learning the empirical risk approximates the risk by

$$\min_{\theta} \mathbb{E}[\mathcal{L}(g(\mathbf{X}_s; \theta), Y_s)]. \tag{1}$$

The network architecture is composed of multiple hidden layers followed by an output or classification layer. Consider $g(\mathbf{X}_s; \theta)_l = a(f(\mathbf{X}_s; \theta)_l)_l$ as the layer l with an activation function $a(\cdot)_l$ and parameter layer $f(\cdot)_l$ for the given source data and $g(\mathbf{X}_t; \theta)_l$ for the target data analogously. Recent network architectures roughly distinguish between the categories of statistical adaptation [13, 3, 6, 5] and adversarial adaptation [7, 14–17].

In *statistical* adaptation, approaches use one or more higher layers, i. e., the fully connected layers of the network, to adapt the output distributions of the (hidden) layers $g(\mathbf{X}_s; \theta)_l$ and $g(\mathbf{X}_t; \theta)_l$[6]. This leads to very individualized approaches. To measure the difference between the output distributions of the network, a divergence measure $dist : g(\mathbf{X}_s; \theta)_l \times g(\mathbf{X}_t; \theta)_l \to \mathbb{R}_0^+$ is employed and added to the objective function:

$$\min_{\theta} \mathbb{E}[\mathcal{L}(g(\mathbf{X}_s; \theta), Y_s)] + \eta \cdot dist(g(\mathbf{X}_s; \theta)_l, g(\mathbf{X}_t; \theta)_l). \tag{2}$$

The dissimilarity measure is used as a regularization. The parameter $\eta \in [0, \infty)$ controls the trade-off between aligning the statistical divergence and minimizing the classification objective. A commonly used dissimilarity measure is the Maximum Mean Discrepancy (MMD) [18], which is the difference in mean of the domain data matrices in a reproducing kernel Hilbert space (RKHS). The minimization of MMD in the proposed networks can be seen as an alignment of statistical moments given a particular kernel, e. g., RBF-Kernel, of the two domains [19].

The authors of [6] proposed the Central Moment Discrepancy for domain adaptation, which explicitly minimizes higher central moments. The CORAL loss [3], minimizing the difference of the full covariance matrices between two domains. Our proposal is a particular case of CORAL. By aligning *only* the singular spectra of the domains, we align the covariances as a side effect. However, our ASAN minimizes a diagonal matrix, which is easier to learn, making it favorable over CORAL. An interpretation of our loss is the minimization of the

second central moment between domains. Further, we do not rely on a particular kernel matrix nor kernel function, but any positive semi-definite (psd) kernel can be used. Due to this flexibility, it is also relatively easy to extend, e. g., relevance weighting as proposed in Sec. 3.3.

Since our ASAN combines statistical and *adversarial* adaptation, we now introduce adversarial learning: let the b_{th} layer of the network be the bottleneck layer and consider the network from the first to the b_{th} layer as the feature extractor $f : \mathcal{X} \to \mathcal{F}$ with parameters θ_f. From the $b + 1_{th}$ layer to the output, let $g : \mathcal{F} \to \mathcal{Y}$ be the classifier network with parameters θ_g, where $\mathcal{F}$ is a latent feature space and $\mathcal{Y}$ is the label space. Usually $b \leq l$ for the domain regularization layer. Additionally, let $d : \mathcal{F} \to \mathcal{C} = \{-1, 1\}$ be a domain classifier with parameters θ_d, predicting the domain of samples. Adversarial domain adaptation yields to minimize the loss of $d(\cdot)$, by propagating the reversed gradients from $d(\cdot)$ to $f(\cdot)$ and trying to confuse $d(\cdot)$ [8]. The Gradient-Reversal-Layer is defined as $R(x) = x$ and $\frac{\partial R}{\partial x} = -\lambda \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The invariant representation is achieved at the saddle point of

$$\min_{\theta_f, \theta_g, \theta_d} \mathbb{E}[\mathcal{L}(g(f(\mathbf{X}_s; \theta_f); \theta_g), Y_s))] + \lambda \mathbb{E}[\mathcal{L}_d(d(R[f(\mathbf{X}; \theta_f)]; \theta_d), Y_d)] \tag{3}$$

where $Y_d = [\mathbf{1}n, -\mathbf{1}m]$ are the domain labels of source size n and target size m and $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t]$. The vanilla DANN [8] implements the cross-entropy for $\mathcal{L}_d$. Other authors [15] used the Wasserstein distance in $\mathcal{L}_d$ because of the intuitive expression of distribution differences [20]. DANN-type networks have also been extended to *normalized* Wasserstein distances [21].

The Conditional Domain Adaptation Network (CDAN) [16] enriches bottleneck features with class conditional confidences via multi-linear mapping $T : \mathcal{F} \times \mathcal{Y} \to \mathcal{F}_c$, which is fed into the discriminator, i. e., $d(T(\cdot))$. CDAN is the baseline in related networks and is extended in this work due to its good performance. The SDAN [22] integrates the Spectral Normalization (SN) [12] to obtain 1-Lipschitz continuous gradients. SN is also a building block in our network, but SDAN does no statistical alignment during learning.

None of the adversarial networks above explicitly define $dist(\cdot)$ in Eq. (3) in the same way as statistical deep learning does in Eq. (2). In this sense, the Batch-Spectral-Penalization (BSP) network is related to us in terms of shrinking the first k singular values, given features from $f(\cdot)$, to lower the influence of easily transferable features. However, our ASAN network explicitly defines $dist(\cdot)$ to align the spectra, which is crucial to be less dependent on the source feature spectrum. In [23], an element-wise comparison of distributions in the label space is implemented and [14] where $l < b$, aligning distributions in low-level filters. Our proposed loss directly modifies adversarial and statistical characteristics in one loss, making it superior to discussed approaches.

3 Model

This section presents the main contribution: The reasoning behind the Relevance Spectral Loss (RSL) in Sec. 3.1 and 3.2. Afterward, the loss itself in Sec. 3.3 and

the combination with SN in Sec. 3.4. Further, we analyze the learning and theoretical properties in Sec. 3.5 and 3.6, respectively. We present the architecture of the network in Fig. 1.

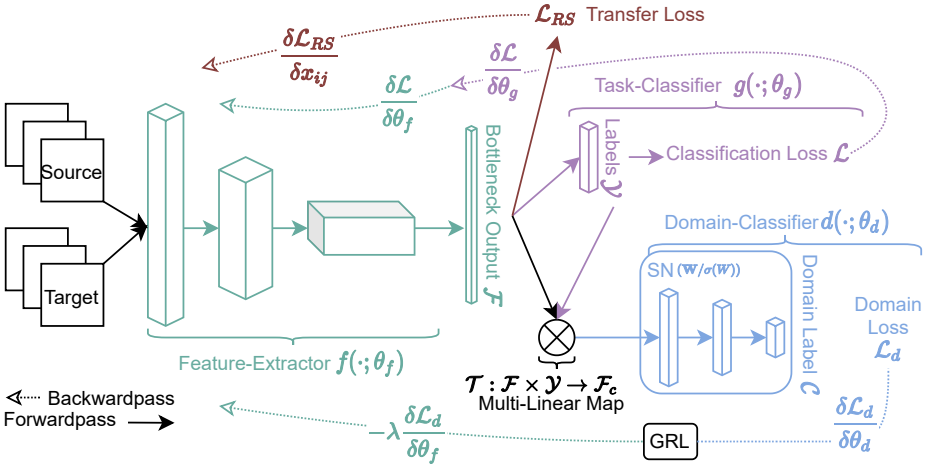


Fig. 1: Architectural overview of our proposed ASAN model, extending the CDAN [16] network by our $\mathcal{L}_{RS}$ and Spectral Normalization [12] (SN). GRL is the Gradient-Reversal-Layer [8].

3.1 Statistical Properties of Singular Values

Given the background in Sec 2, let $\mathbf{X}_s^l$ and $\mathbf{X}_t^l$ be the output of the source and target from layer b or more precise the output of $f(\cdot)$ called bottleneck features. The Singular Value Decomposition (SVD) of these outputs is given with $\mathbf{X}_s^l = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ and $\mathbf{X}_t^l = \mathbf{LTR}^T$. Here $\mathbf{R}, \mathbf{V} \in \mathbb{R}^{d \times d}$, $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{L} \in \mathbb{R}^{m \times m}$ are matrices. Further, $\mathbf{U}, \mathbf{L}, \mathbf{V}$ and $\mathbf{R}$ are column-orthogonal. $\boldsymbol{\Sigma}$ is a $n \times d$ matrix wherein all entries $\sigma_{ij} = 0$ iff $i \neq j$ and $\mathbf{T}$ is a $m \times d$ matrix wherein all entries $t_{ij} = 0$ iff $i \neq j$. Furthermore, by σ_k we denote the singular value in the k-th column of $\boldsymbol{\Sigma}$. For a linear covariance function, we can decompose the respective kernel with the eigenvalue decomposition and the SVD into

$$\mathbf{K} = \mathbf{CDC}^{-1} = \mathbf{X}^T\mathbf{X} = (\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T)(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) = \mathbf{C}\boldsymbol{\Sigma}^2\mathbf{C}^{-1}, \tag{4}$$

where $\mathbf{C}$ are the eigenvectors (right singular vectors of $\mathbf{X}$) and $\mathbf{D}$ are the eigenvalues of $\mathbf{K}$. The singular values $\boldsymbol{\Sigma}$ of $\mathbf{X}$ are the square root eigenvalues of $\mathbf{K}$. Accordingly, the entries of the diagonal of $\mathbf{D}, \boldsymbol{\Sigma}^2$ give the variance of the columns of $\mathbf{K}$. Assuming that the expected values $\mathbb{E}_s(\mathbf{X}_s^l) = 0, \mathbb{E}_t(\mathbf{X}_t^l) = 0$, we got $tr(\boldsymbol{\Sigma}^2) = tr(\mathbf{X}^t\mathbf{X})$. Hence, minimizing the difference between $\boldsymbol{\Sigma}$ and $\mathbf{T}$ is the same as minimizing covariance matrices differences on the diagonal. Subsequently, we assume the same batch sizes for both domains during training.

3.2 Relationship of Feature Characteristics and Singular Values

Given $\boldsymbol{\Sigma}$ and $\mathbf{T}$ produced by a DANN [8] network, each spectrum associated with their features is separable into three areas. Large singular values represent features with high transferability due to high principal angels between their associated subspaces. Features with mid-size singular values contain class discriminative information for learning classification tasks [7]. Small singular values interfere with the generalization ability of a network because of domain-specific features, based on the observation of shrinking small source singular values in the fine-tuning process of neural networks and low principal angels between the corresponding source and target singular vectors [11].

Combining [7, 11] from above and the statistical perspective (Sec. 3.1), singular values express the variance of the feature covariance matrix and are associated with the variance of filter outputs. Therefore, large singular values correspond to high variance, resulting in transferability due to uniform filter activations in both domains. Mid-size singular values are associated with filters producing class-discriminative information with more class-specific activations. Small singular values express domain-specific features due to the low expressiveness of filters over domain borders. By aligning the spectra, overly emphasized features, given large singular values, are neglected due to the similarity of the singular values, while the discriminative features are aligned. Aligning the statistics (Sec. 3.1) and shrink domain-specific signals [11] enables rich adaptation without the need for devastating transferable features [7], which is counter-intuitive in adaptation tasks.

3.3 Relevance Spectral Alignment

The approach [7] of shrinking the k highest or smallest k singular values from both domains for domain adaptation comes with the drawback that they are still related to the source spectrum. In some sense, it is counter-intuitive to shrink the influence of highly transferable features in the adaptation task. However, by aligning the most significant singular values, the network does not rely on one spectrum. The expressed variances of the features are the result of two domains. Following the same reasoning, the classification task is enhanced by not relying on the description of one but the alignment of two spectra. Only the domain-specific contents should be shrunk due to low expressiveness over domain borders [11], which we consider as relevance weighing of domain-specific features. Finally, we define our proposed Relevance Spectral loss as

$$\mathcal{L}_{RS} = ||(\boldsymbol{\Sigma}_n - \mathbf{T}_n) + (\boldsymbol{\Sigma}_k^2 - \mathbf{T}_k)||_F^2 \tag{5}$$

where $\boldsymbol{\Sigma}_n$ and $\mathbf{T}_n$ are the largest $n-k$ singular values and $\boldsymbol{\Sigma}_k$ and $\mathbf{T}_k$ are the smallest k singular values respectively. Further, $||\cdot||_F^2$ is the squared Frobenius norm. We follow the fine-tuning perspective of [11] and actively shrink only the k smallest source singular values. However, we show in Sec. 3.5 that the respective target singular values are also minimized during learning. The loss

can be integrated into any layer as a regularization term or simultaneously used in multiple layers, as suggested by [13]. Here we use our proposed approach in the bottleneck layer.

$\mathcal{L}_{RS}$ has three main benefits. The network aligns the source and target spectrum during learning and depends not only on the source spectrum in $\mathcal{F}$. Assuming the properties as in Sec. 3.1, the network minimizes the difference of the diagonal entries of covariance matrices, making it related to [3, 6], which adapts the second central moment. Domain-specific information [11] in the last k source singular values is minimized during the adversarial adaptation process. Simultaneously, discriminative and transferable features [7] in the largest $n-k$ are encouraged to be large and similar over domain borders. Summarizing, $\mathcal{L}_{RS}$ bridges statistical and enhances adversarial adaptation formulated in one loss, making ASAN favorable in simplicity and theoretical understanding.

3.4 Stabilize Discriminator Gradients

The domain classifier $d(\cdot)$ is related to the discriminator in GANs [22]. The GAN discriminator suffers from unstable gradients and saturation of training due to perfect predictions [24]. The occurrence of these problems is also possible in training the domain classifier. We integrate Spectral Normalization [12] in the domain classifier network to tackle this problem by regularizing the gradients to be 1-Lipschitz [12] and avoid the situation where the gradients coming from $d(\cdot)$ are devastating gradients from $\mathcal{L}_{RS}$. Let $\mathbf{W}$ be parameters in an arbitrary layer of $d(\cdot)$, then SN [12] has the form

$$\mathbf{W}_{sn} = \frac{\mathbf{W}}{\sigma(\mathbf{W})}, \tag{6}$$

where $\sigma(\mathbf{W})$ is the largest singular value of the parameters $\mathbf{W}$ and $||\mathbf{W}_{sn}||_{lip} \leq 1$. We implement SN [12] into every layer of the domain classifier, and SN takes place after the forward pass and before gradient propagation. In the following, we call the combination of $\mathcal{L}_{RS}$, SN and CDAN-baseline the *Adversarial Spectral Adaptation Network*. See Fig. 1 for an architectural overview.

3.5 Learning Procedure

From the perspective of statistical adaptation, data-driven gradients are common [3, 13, 6]. Therefore, we follow the suggestions and learn the RSL given the source $\frac{\partial \mathcal{L}_{RS}}{\partial \mathbf{x}_s}$ and target data $\frac{\partial \mathcal{L}_{RS}}{\partial \mathbf{x}_t}$. Given the definition of the covariance-matrix in Eq. (4) and following [25, 26], the derivative of a singular value σ and a squared singular value σ^2 w.r.t a sample point is

$$\frac{\partial \sigma_e}{\partial x_{ij}} = u_{ie} v_{je}, \qquad \frac{\partial \sigma_e^2}{\partial x_{ij}} = v_{ie} v_{je}. \tag{7}$$

where u_{ie} and v_{je} are the components of the left and right singular vectors of $\mathbf{X}$. The derivative of the e_{th} singular value is given by the j_{th} feature of the i_{th}

data-sample and is defined for any $\sigma_i \neq \sigma_j$. For a detailed description of the derivative of singular and eigen-values, see [25, 26].

Looking once more at Eq. (5), the loss needs a derivative for each domain. The source-based derivative of $\mathcal{L}_{RS}$ is given by the derivative of σ_k w.r.t. the source data x^s_{ij}, therefore

$$\frac{\partial \mathcal{L}_{RSL}}{\partial x^s_{ij}} = \frac{\partial \sigma_e}{\partial x^s_{ij}} \sqrt{\Sigma_{e=1}^{n-k}(\sigma_e - t_e)^2 + \Sigma_{i=k+1}^{n}(\sigma_e^2 - t_e)^2}^2 \tag{8}$$

$$= 2\Sigma_{e=1}^{n-k}(\sigma_e - t_e)\frac{\partial \sigma_e}{\partial x^s_{ij}} + 2\Sigma_{e=k+1}^{n}(\sigma_e^2 - t_e)\frac{\partial \sigma_e}{\partial x^s_{ij}} \tag{9}$$

$$= 2\big(\Sigma_{e=1}^{n-k}(\sigma_e - t_e) \cdot |u_{ie}v_{je}| + \Sigma_{e=k+1}^{n}(\sigma_e^2 - t_e) \cdot |v_{ie}v_{je}|\big). \tag{10}$$

Let l_{ie} and r_{je} be the singular vector components of the target data. The target-based derivative of $\mathcal{L}_{RS}$ is analogously given by t_k w.r.t target data x^t_{ij} as

$$\frac{\partial \mathcal{L}_{RSL}}{\partial x^t_{ij}} = -2\big(\Sigma_{e=1}^{n-k}(\sigma_e - t_e) \cdot |l_{ie}r_{je}| + \Sigma_{e=k+1}^{n}(\sigma_e^2 - t_e) \cdot |l_{ie}r_{je}|\big). \tag{11}$$

The derivatives give some interesting insights to discuss. The absolute value of the singular-vector components is not a derivative product but was added afterward to avoid sign flipping of singular-vectors [27]. Sign flipping leads to wrongly directed gradients. Both derivations point in opposite directions leading to equilibrium at $\mathcal{L}_{RS} = 0$ because every iteration makes a step towards the respective other domain.

The difference between the $n-k$ largest singular values from the source and target is regularized by the component-wise correlation of the row and column spaces given a source or target data point. This prevents a too drastic focus on one of the spaces. The smallest k source singular values are regularized by the component-wise correlation of right singular vectors given feature x_{ij}. Consequently, the smallest k target singular values are mitigated while the target spectrum adapts to the source spectrum.

3.6 Theoretical Properties

The analysis of ASAN relies on the work of Zhao et al. [28], which extends the domain adaptation theory of Ben-David et al. [10]. To improve the readability, we adapt the former notation as follows: let $\varepsilon(s)$ and $\varepsilon(t)$ be the risk of classifier $g(\cdot)$ on the source and target domain. $\varepsilon(d)$ is the risk of the trained domain classifier $d(\cdot)$. Further, let $min\{\mathbb{E}_{p(\mathcal{S})}[|f_s - f_t|], \mathbb{E}_{p(\mathcal{T})}[|f_s - f_t|]\}$ be the expected divergence between the optimal labeling function of source f_s and target f_t w.r.t the source and target marginal distributions. Define $d_{\tilde{\mathcal{H}}}(p(\mathcal{S}), p(\mathcal{T})) = \sup_{\tilde{h} \in \tilde{\mathcal{H}}} |Pr_{p(\mathcal{S})}(\tilde{h} = 1) - Pr_{p(\mathcal{T})}(\tilde{h} = 1)|$ as the disagreement of hypothesis $\tilde{h}$ on the source and target distribution given $\tilde{\mathcal{H}} = \{sgn(|h(\mathbf{x}) - h'(\mathbf{x})| - z) \mid h, h' \in \mathcal{H}, 0 \leq z \leq 1\}$, where $\mathcal{H}$ is a hypothesis class with finite VC dimension. Conveniently, this is referred

to as the difference in marginal distribution [28]. The domain adaptation theory by [28] states that

$$\varepsilon(t) \leq \varepsilon(s) + d_{\tilde{\mathcal{H}}}(p(\mathcal{S}), p(\mathcal{T})) + min\{\mathbb{E}_{p(\mathcal{S})}[|f_s - f_t|], \mathbb{E}_{p(\mathcal{T})}[|f_s - f_t|]\}. \quad (12)$$

Assuming a fixed representation obtained from $f(\cdot)$, it is shown for CDAN that learning $d(\cdot)$ yields an upper bound, i. e., $d_{\tilde{\mathcal{H}}}(p(\mathcal{S}), p(\mathcal{T})) \leq \sup|\varepsilon(d)|$ where $z = 0$ [16]. This is done under the assumption that the hypothesis class $\mathcal{H}_d$ of $d(\cdot)$ is rich enough to contain $d_{\tilde{\mathcal{H}}}$, i. e., $d_{\tilde{\mathcal{H}}} \subset \mathcal{H}_d$ for $z = 0$. This is not an unrealistic scenario, since we are able to choose $d(\cdot)$ as a multi-layer perceptron approximating any function [16, 8]. Following this reasoning, we assume that $\mathcal{H}_d$ is also rich enough that $d_{\tilde{\mathcal{H}}} \subset \mathcal{H}_d$ for $0 \leq z \leq 1$ and bound Eq. (12) by

$$\varepsilon(t) \leq \varepsilon(s) + \sup|\varepsilon(d)| + min\{\mathbb{E}_{p(\mathcal{S})}[|f_s - f_t|], \mathbb{E}_{p(\mathcal{T})}[|f_s - f_t|]\}. \quad (13)$$

See [16, 8, 28] for technical details about the proof. Hence, minimizing $\sup|\varepsilon(d)|$ by learning $d(\cdot;\theta_d)$, influenced by learning $\mathcal{L}_{RS}$, yields an upper bound for the risk of $g(\cdot)$ on the target domain, i. e., $f(\cdot)$ learns an invariant representation. In particular, the last term of Eq. 13, i. e., $min\{\cdot\}$, is the limitation of ASAN, since it does not approximate the labeling functions. Therefore, the performance of DANN-type networks is limited to differences in ground truth labeling of source and target, see Fig. 1 in [28].

Time Complexity. The SVD required for $\mathcal{L}_{RS}$ is $\mathcal{O}(min(p, d_f)^2)$ where p is the batch size and d_f is the dimension of the bottleneck space $\mathcal{F}$. The input space is usually high dimensional due to RGB image data, e. g., $d_x = 3 \times 224 \times 224 = 150.528$ in Resnet50 [1]. Given the computational complexity of convolution [29] and $d_f = 256 \ll d_x$ in ASAN, the computation of SVD at the bottleneck layer does not increase the complexity class of the network. Further, SN uses a *modified* power iteration converging in one iteration [12]. In practice, the time requirements of our extensions are neglectable in comparison to the Resnet50 training time.

4 Experiments

We provide the experimental validation of the ASAN architecture against state-of-the-art domain adaptation networks on standard benchmark datasets. The PyTorch code is published at `https://github.com/ChristophRaab/ASAN`.

4.1 Datasets

Office-31 [30] is an image data set with 4652 photographs, each assigned to one of the 31 classes. The dataset is divided into the three domains Amazon (**A**), **D**igital **S**ingle-**L**ens **R**eflex camera (**D**) and Webcam (**W**). The domain adaptation task is to learn on one domain and test on another. The shift between the domains results from differences in surroundings and camera characteristics. The tasks are **A**→**W**, **D**→**W**, **W**→**D**, **A**→**D**, **D**→**A** and **W**→**A**.

Image-Clef is another image dataset and was released in 2014 as part of the ImageClef domain adaptation challenge. It contains 12 common classes from domains Caltech-256 (**C**), ImageNet ILSVRC2012 (**I**), and PAS-CALVOC2012 (**P**) with an total of 1.800 photos. The test setting, again similar to Office-31, is **I**→**P**, **P**→**I**, **I**→**C**, **C**→**I**, **C**→**P** and **P**→**C**.

Office-Home [31] is more comprehensive and more difficult than Office-31 with 65 classes and 15.500 images in total. The dataset domains are Art (**A**), containing painting and sketches, Clipart (**C**), Product (**P**), containing product images without background, and real-world (**R**), containing objects from regular cameras. The test setting for domain adaptation is **A**→**C**, **A**→**P**, **A**→**R**, **C**→**A**, **C**→**P**, **C**→**R**, **P**→**A**, **P**→**C**, **P**→**R**, **R**→**A**, **R**→**C** and **R**→**P**.

4.2 Implementation Details

Architecture. Following the architectural style of CDAN+E (+E : Entropy reweighting)[16], the Resnet50 bottleneck network [1] pre-trained on Imagnet represents the feature extractor $f(\cdot)$. The classifier $g(\cdot)$ on top of $f(\cdot)$ is a fully connected network matching $\mathcal{F}$ to the task depended label space $\mathcal{Y}$, e. g., with 31 dimensions for Office-31. The classifier loss is cross-entropy. The domain classifier $d(\cdot)$ has three fully connected layers, while the first two have RELU activations and dropout, and the last has sigmoid activation. The input of the discriminator $d(\cdot)$ is the result of the multi-linear map $T(f, g) = f(\mathbf{X}) \otimes g(\mathbf{X})$ [16]. The loss of $d(\cdot)$ is binary-cross-entropy. In the adaptation process, the whole network is fine-tuned on the domain adaptation task. Beyond that, we extend the following: all discriminator layers are regularized with SN [12]. The proposed RSL is computed from the source and target bottleneck features and propagated to the feature extractor.

Competitive Methods. We compare our network against the following recent adversarial networks: Domain Adversarial Neural Network (**DANN**) [8], Conditional Domain Adversarial Network (**CDAN**) [16], Batch Spectral Penalization (**BSP**) [7], Spectral Normalized CDAN (**SDAN**) [22], Joint Adaptation Network (**JAN**) [13], Stepwise Adaptive Feature Norm (**SAFN**) [32] and Enhanced Transport Distance (**ETN**) [33]. Note that the authors of ETN have not provided the standard deviation in their results. But we still want to show the performance against very recent works.

Experimental Setup. We follow the standard study protocol [13] for unsupervised deep domain adaptation and use all available labeled source and unlabeled target data. The results are the mean and standard deviation given three random runs using the best-reported target accuracy per training process. All approaches use the same feature extractor. Reported results of former work are directly copied in the result tables 1, 2, and 3 if the experimental designs are the same. The classifier and discriminator are trained from scratch with a learning rate ten times the feature extractors learning rate.

Parameters. The ASAN hyper-parameters are optimized as in [34] on the Office-31 dataset and set to $k = 11$ and $\eta = 10e^{-3}$ for all datasets. The supplementary gives more details about the tuning and behavior of k. All pa-

rameters are trained via mini-batch SGD with a momentum of 0.9. The initial learning rate $\zeta_0 = 0.001$ is modified by a progress based adjustment of $\zeta_p = \zeta_0(1 + \alpha \cdot p)^{-\beta}$, where $\alpha = 10$, $\beta = 0.75$, and $0 \leq p \leq 1$ depending on the training process as suggested by [8]. The λ parameter for the discriminator contribution to the overall loss is progressively increased from 0 to 1 based on $(1 - exp(-\delta \cdot x))/(1 + exp(-\delta \cdot x))$, with $\delta = 10$ as suggested for adversarial architectures [16].

4.3 Performance Results

We report the experiments for Office-31 in Tab. 1, for Image-Clef in Tab. 2, and Office-Home in Tab. 3 as accuracy (0-100%). Overall, **the ASAN architecture** outperforms the compared algorithms in two out of three datasets, while having the **overall best mean performance**. The results are obtained by optimizing the parameters only on the Office-31 dataset. This shows the robustness of the performance of our ASAN in terms of parameter sensitivity across changing tasks, making it a stable approach and easily applicable to related real-world scenarios. At Office-31, ASAN reports the best performance at four out of six comparisons, while showing the best mean performance. The second-best performing algorithm is SDAN, which also relies on CDAN and SN. However, due to no additional alignment, our ASAN is superior to SDAN by learning a bottleneck space, aligning both domain spectra. Further, the BSP approach seems to not create an invariant representation by shrinking the first k singular values competitive with ASANs spectral alignment (RSL). At Image-Clef, ASAN is only second-best in performance. However, better than related CDAN and SDAN. This suggests that learning our proposed RSL within ASAN improves CDAN, leading to better performance than related methods. At Office-Home, ASAN demonstrates outstanding performance by outperforming in ten out of twelve tasks. The results are comparable with the results of Office-31. The ASAN is best, ETN is second, and SDAN is third in mean performance. Again, RSL, combined with SN, is superior to related approaches such as BSP, SDAN, or CDAN and, further, outperforms very recent benchmark performances of ETN.

Table 1: Mean prediction **accuracy** with standard deviation on the **Office-31** dataset over three random runs.

Dataset	**A→W**	**D→W**	**W→D**	**A→D**	**D→A**	**W→A**	Avg.
Resnet [1]	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN (2015) [8]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN (2017) [13]	85.4±0.3	97.4±0.2	99.8±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
CDAN (2018) [16]	93.1±0.2	98.2±0.2	**100±0**	89.8±0.3	70.1±0.4	68.0±0.4	86.6
BSP (2019) [7]	93.3±0.2	98.2±0.2	**100±0**	93.0±0.2	73.6±0.3	72.6±0.3	88.5
SDAN (2020) [22]	95.3±0.2	98.9±0.1	**100±0**	**94.7±0.3**	72.6±0.2	71.7±0.2	88.9
ETN (2020) [33]	92.1	**100.0**	**100.0**	88.0	71.0	67.8	86.2
ASAN (ours)	**95.6±0.4**	98.8±0.2	**100±0**	94.4±0.9	**74.7±0.3**	**74.0±0.9**	**90.0**

Table 2: Mean prediction **accuracy** with standard deviation on the **Image-clef** dataset over three random runs.

Dataset	**I→P**	**P→I**	**I→C**	**C→I**	**C→P**	**P→C**	Avg.
Resnet [1]	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DANN (2015) [8]	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
JAN (2017) [13]	76.8±0.4	88.0±0.2	94.7±0.2	89.5±0.3	74.2±0.3	91.7±0.3	85.8
CDAN (2018) [16]	77.7±0.3	90.7±0.2	97.7±0.3	91.3±0.3	74.2±0.2	94.3±0.3	87.7
SAFN (2019) [32]	78.0±0.4	91.7±0.4	96.2±0.1	91.1±0.6	77.0±0.2	94.7±0.1	88.1
SDAN (2020) [22]	78.1±0.2	91.5±0.2	97.5±0.2	92.1±0.3	76.6±0.3	95.0±0.1	88.4
ETN (2020) [33]	**81.0**	91.7	**97.9**	**93.3**	**79.5**	**95.0**	**89.7**
ASAN (ours)	78.9±0.4	**92.3±0.5**	97.4±0.5	92.1±0.3	76.4±0.7	94.4±0.2	88.6

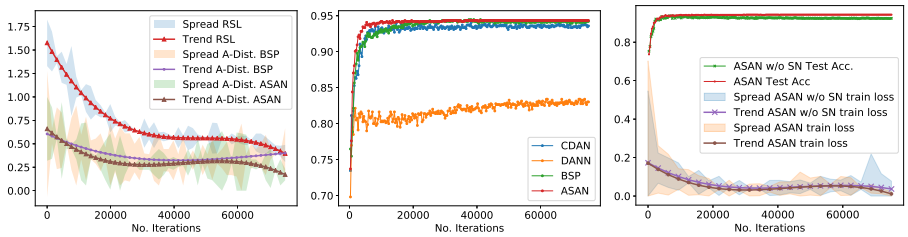
Table 3: Mean prediction **accuracy** on the **Office-Home** dataset over three random runs.

Dataset	**A→C**	**A→P**	**A→R**	**C→A**	**C→P**	**C→R**	**P→A**	**P→C**	**P→R**	**R→A**	**R→C**	**R→P**	Avg.
Resnet [1]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [8]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [13]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN [16]	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
BSP [7]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
SDAN [22]	52.0	72.0	76.3	59.4	71.7	72.6	58.6	52.0	79.2	71.6	58.1	82.8	67.1
ETN [33]	51.3	71.9	**85.7**	57.6	69.2	**73.7**	57.8	51.2	79.3	70.2	57.5	82.1	67.3
ASAN	**53.6**	**73.0**	77.0	**62.1**	**73.9**	72.6	**61.6**	**52.8**	**79.8**	**73.3**	**60.2**	**83.6**	**68.6**

Office-31 and Office-Home experiments demonstrate the ASANs parameter robustness: the parameters optimized on Office-31 are used in different but related tasks and show robust generalization capacities. Robust parametrization and excellent performance make our ASAN favorable.

4.4 Convergence and Spectral Analysis

We report the convergence behavior of our RSL within the ASAN architecture compared to related networks in Fig. 2. The data is obtained by training on **A**→**W** from Office-31 dataset. The $\mathcal{A}$-Distance [35] is defined as $\mathcal{A} = 2(2 - 1\varepsilon)$, where ε is the error of the trained domain classifier. The $\mathcal{A}$-Distance is related to the $d_{\tilde{\mathcal{H}}}(\mathcal{S}, \mathcal{T})$ in Sec. 3.6 and measures the domain classifiers inability to distinguish the source and target domain. In contrast, a low $\mathcal{A}$-Distance is an indicator for an invariant representation [35]. The proposed RSL and the $\mathcal{A}$-Distance of our ASAN and BSP are shown in 2a, which we compare due to the commonality of manipulating the feature spectra. The interpolated lines (red, purple, brown) show the overall learning trend while the colored areas (blue, orange, green) show the fluctuation during learning. We observe that our network learns a better invariant representation via an almost all-time lower $\mathcal{A}$-Distance by not relying on only one spectrum. The plot shows that spectral differences represented by RSL are effectively reduced. Interestingly, the trend curves of the $\mathcal{A}$-Distance of



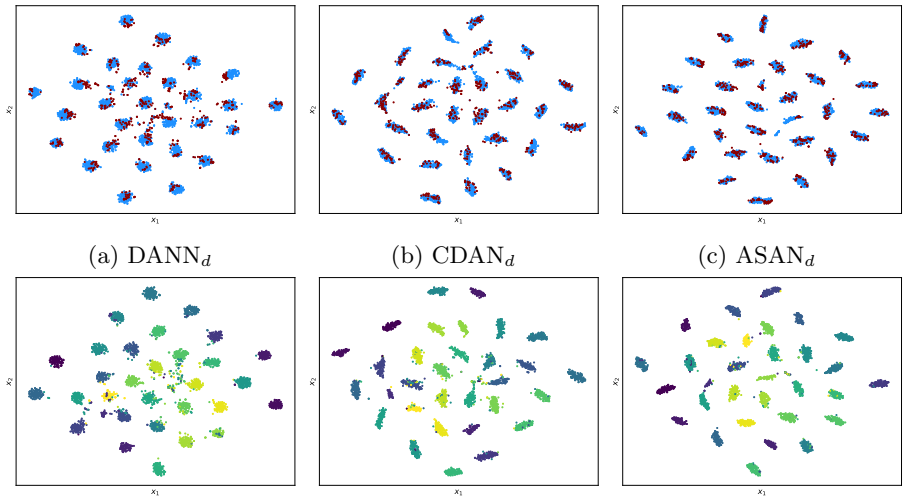
(a) A-Distance & RSL (b) Target Acc. (c) Source loss & Test Acc.

Fig. 2: Learning process of ASAN compared to related networks over time given **A**→**W** images from Office-31 dataset. Best viewed on computer display.

ASAN and the RSL are similar in shape, allowing the presumption that learning RSL is related to learning an invariant representation, i. e., minimizing the $\mathcal{A}$-Distance. Figure 2b represents the target accuracy during learning. It is observable that the ASAN network converges very fast in a higher target accuracy than related approaches. Further, the saturation is very stable and practically does not change once reached. This behavior of ASAN is related to the Spectral Normalization by giving well-defined gradients back to the feature extractor. This assumption is verified in Fig. 2c, where the learning and evaluation process of ASAN itself and ASAN without Spectral Normalization (ASAN w/o SN) is plotted. The target accuracy of ASAN (red) remains stable while the accuracy of ASAN w/o SN (green), after reaching the best performance similar to ASAN, has a decline in target accuracy. In contrast, ASAN remains stable at high accuracy. The trends of train loss (brown for ASAN and purple for ASAN w/o SN) show an almost all-time lower learning loss of ASAN. The fluctuation of the train losses shows that ASAN (blue) is more stable than ASAN w/o SN (orange) and, most of the time, lower in value. Additional results on hyperparameter behavior and an ablation study are presented in the supplementary.

4.5 Feature Analysis

We evaluate the empirical feature representation of the bottleneck features given **A**→**W** images from the Office-31 dataset. The result is reported in Fig. 3 based on T-SNE [36]. The plot is split into two parts: the top row (Fig. 3a - 3c) is a scatter plot of the bottleneck features of trained DANN, CDAN, and ASAN colored with ground truth domain labels. Blue shows the source, and red shows the target domain. ASAN shows the superiority of creating a domain invariant representation by almost perfectly assigning all red points to a blue cluster compared to CDAN and DANN. The bottom row (Fig. 3d - 3f) shows the same representation but with classification labels. The class-label plots show that ASAN representations are easily classifiable by a neural network. However, some points are still located in the wrong cluster, representing the limitation of ASAN described in Sec. 3.6. The ASAN, DANN, and CDAN do not approximate the



(a) DANN$_d$ (b) CDAN$_d$ (c) ASAN$_d$

(d) DANN$_c$ (e) CDAN$_c$ (f) ASAN$_c$

Fig. 3: T-SNE [36] of bottleneck features of selected networks given **A**→**W** images from Office-31 dataset. <Name>$_d$ and <Name>$_c$ show the outputs with ground truth domain and classification labels respectively. For the first row, blue shows the source, and red shows the target domain Best viewed in color.

label distribution of target during learning. Therefore, the target accuracy of ASAN is bounded by the distribution differences of label distributions [28]. As a result, the label distribution difference is directly related to the remaining misclassified samples. However, as shown in the performance evaluation (Sec. 4.3), ASAN performs considerably better than remaining networks and is, therefore, the preferable choice. Additional results to feature, convergence, and spectral analysis are offered in the supplementary material of the contribution.

5 Conclusion

We proposed the ASAN architecture, integrating Relevance Spectral Alignment and Spectral Normalization into the existing CDAN method. ASAN learns a bottleneck space, aligning both domain spectra while minimizing domain-specific information. The theoretical inspection of the gradients of RSL suggests that ASAN learns an invariant representation, empirically confirmed on three standard domain adaptation datasets. Further, ASAN has robust parametrization, making it easy to apply to other tasks. Compared to related approaches, ASAN is more stable and converges faster to a better solution. Prior theoretical evaluations of CDAN show that the performance of the domain classifier bounds the label-classifier of ASAN. Future research should target class conditional or cluster-based spectral alignment.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2016-Decem. (2016) 770–778
2. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. IEEE Computational Intelligence Magazine **13** (2018) 55–75
3. Sun, B., Saenko, K.: Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In Hua, G., Jégou, H., eds.: Computer Vision – ECCV 2016 Workshops, Cham, Springer International Publishing (2016) 443–450
4. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 3320–3328
5. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning Transferable Features with Deep Adaptation Networks. In Bach, F., Blei, D., eds.: Proceedings of the 32nd International Conference on Machine Learning. Volume 37 of Proceedings of Machine Learning Research., Lille, France, PMLR (2015) 97–105
6. Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., Saminger-Platz, S.: Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. (2017)
7. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. Discriminability: Batch Spectral Penalization for Adversarial Domain Adaptation. In Chaudhuri, K., Salakhutdinov, R., eds.: Proceedings of the 36th International Conference on Machine Learning. Volume 97 of Proceedings of Machine Learning Research., Long Beach, California, USA, PMLR (2019) 1081–1090
8. Ganin, Y., Lempitsky, V.S.: Unsupervised Domain Adaptation by Backpropagation. In: Proceedings of the 32nd International Conference on Machine Learning, ICLR 2015, Lille, France, 6-11 July 2015. (2015) 1180–1189
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 2672–2680
10. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine Learning **79** (2010) 151–175
11. Chen, X., Wang, S., Fu, B., Long, M., Wang, J.: Catastrophic Forgetting Meets Negative Transfer: Batch Spectral Shrinkage for Safe Transfer Learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R., eds.: Advances in Neural Information Processing Systems 32. Curran Associates, Inc. (2019) 1908–1918
12. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral Normalization for Generative Adversarial Networks. In: International Conference on Learning Representations. (2018)
13. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep Transfer Learning with Joint Adaptation Networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML'17, JMLR.org (2017) 2208–2217
14. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In Dy, J., Krause, A.,

eds.: Proceedings of the 35th International Conference on Machine Learning. Volume 80 of Proceedings of Machine Learning Research., Stockholmsmässan, Stockholm Sweden, PMLR (2018) 1989–1998
15. Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In McIlraith, S.A., Weinberger, K.Q., eds.: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press (2018) 4058–4065
16. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional Adversarial Domain Adaptation. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada. (2018) 1647–1657
17. Li, S., Liu, C.H., Xie, B., Su, L., Ding, Z., Huang, G.: Joint Adversarial Domain Adaptation. In: Proceedings of the 27th ACM International Conference on Multimedia, New York, NY, USA, ACM (2019) 729–737
18. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A Kernel Two-sample Test. J. Mach. Learn. Res. **13** (2012) 723–773
19. Li, Y., Swersky, K., Zemel, R.: Generative Moment Matching Networks. In Bach, F., Blei, D., eds.: Proceedings of the 32nd International Conference on Machine Learning. Volume 37 of Proceedings of Machine Learning Research., Lille, France, PMLR (2015) 1718–1727
20. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein Generative Adversarial Networks. In Precup, D., Teh, Y.W., eds.: Proceedings of the 34th International Conference on Machine Learning. Volume 70 of Proceedings of Machine Learning Research., International Convention Centre, Sydney, Australia, PMLR (2017) 214–223
21. Balaji, Y., Chellappa, R., Feizi, S.: Normalized Wasserstein for Mixture Distributions With Applications in Adversarial Learning and Domain Adaptation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE (2019) 6499–6507
22. Zhao, L., Liu, Y.: Spectral Normalization for Domain Adaptation. Information **11** (2020)
23. Rakshit, S., Chaudhuri, U., Banerjee, B., Chaudhuri, S.: Class Consistency Driven Unsupervised Deep Adversarial Domain Adaptation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE (2019) 657–666
24. Arjovsky, M., Bottou, L.: Towards Principled Methods for Training Generative Adversarial Networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net (2017)
25. Papadopoulo, T., Lourakis, M.I.A.: Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications. In: Computer Vision - ECCV 2000, Berlin, Heidelberg, Springer Berlin Heidelberg (2000) 554–570
26. Petersen, K.B., Pedersen, M.S.: The Matrix Cookbook (2008)
27. Bro, R., Acar, E., Kolda, T.G.: Resolving the sign ambiguity in the singular value decomposition. Journal of Chemometrics **22** (2008) 135–140
28. Zhao, H., Combes, R.T.D., Zhang, K., Gordon, G.: On Learning Invariant Representations for Domain Adaptation. In Chaudhuri, K., Salakhutdinov, R., eds.: Proceedings of the 36th International Conference on Machine Learning. Volume 97 of Proceedings of Machine Learning Research., Long Beach, California, USA, PMLR (2019) 7523–7532

29. He, K., Sun, J.: Convolutional neural networks at constrained time cost. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 15., IEEE (2015) 5353–5360
30. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting Visual Category Models to New Domains. In Daniilidis, K., Maragos, P., Paragios, N., eds.: Computer Vision – ECCV 2010, Berlin, Heidelberg, Springer Berlin Heidelberg (2010) 213–226
31. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep Hashing Network for Unsupervised Domain Adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2017-Janua., IEEE (2017) 5385–5394
32. Xu, R., Li, G., Yang, J., Lin, L.: Larger Norm More Transferable: An Adaptive Feature Norm Approach for Unsupervised Domain Adaptation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Volume 2019-Octob., IEEE (2019) 1426–1435
33. Li, M., Zhai, Y.M., Luo, Y.W., Ge, P.F., Ren, C.X.: Enhanced Transport Distance for Unsupervised Domain Adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020) 13936–13944
34. Zhong, E., Fan, W., Yang, Q., Verscheure, O., Ren, J.: Cross validation framework to choose amongst models and datasets for transfer learning. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6323 LNAI** (2010) 547–562
35. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of Representations for Domain Adaptation. In Schölkopf, B., Platt, J.C., Hoffman, T., eds.: Advances in Neural Information Processing Systems 19. MIT Press (2007) 137–144
36. Laurens van der Maaten, Geoffrey E., H.: Visualizing Data using t-SNE. Journal of Machine Learning Research **164** (2008) 10