

Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation

Chen-Yu Lee Tanmay Batra Mohammad Haris Baig Daniel Ulbricht
Apple Inc

Abstract

In this work, we connect two distinct concepts for unsupervised domain adaptation: feature distribution alignment between domains by utilizing the task-specific decision boundary [57] and the Wasserstein metric [72]. Our proposed sliced Wasserstein discrepancy (SWD) is designed to capture the natural notion of dissimilarity between the outputs of task-specific classifiers. It provides a geometrically meaningful guidance to detect target samples that are far from the support of the source and enables efficient distribution alignment in an end-to-end trainable fashion. In the experiments, we validate the effectiveness and genericness of our method on digit and sign recognition, image classification, semantic segmentation, and object detection.

1. Introduction

Deep convolutional neural networks [30] is a milestone technique in the development of modern machine perception systems solving various tasks such as classification, semantic segmentation, object detection, etc. However, in spite of the exceptional learning capacity and the improved generalizability, deep learning models still suffer from the challenge of domain shift – a shift in the relationship between data collected in two different domains [3, 2] (e.g. synthetic and real). Models trained on data collected in one domain can perform poorly on other domains. Domain shift can exist in multiple forms: covariate shift (distribution shift in attributes), prior probability shift (shift in labels), and concept shift (shift in the relationship between attributes and labels) [61, 66, 44].

In this paper, we focus on the covariate shift problem for the case where we have access to labeled data from one domain (source) and unlabeled data from another domain (target). This setup is commonly called unsupervised domain adaptation. Most of the work done in this field has focused on establishing a direct alignment between the feature distribution of source and target domains. Such alignment involves minimizing some distance measure of the feature distribution learned by the models [56, 17, 38]. More sophisticated methods use adversarial training [19] to fur-

ther improve the quality of alignment between distributions by adapting representations at feature-level [24, 18], pixel-level [36, 70, 5], or output-level [69] across domains.

A recent advance that moves beyond the direction of plain distribution matching was presented by Saito *et al.* in [57]. They propose a within-network adversarial learning-based method containing a feature generator and two (task-specific) classifiers, which uses the task-specific decision boundary for aligning source and target samples. Their method defines a new standard in developing generic domain adaptation frameworks. However, the system does have some limitations. For instance, their discrepancy loss (L_1 in this case) is only helpful when the two output probability measures from the classifiers overlap.

Inspired by the framework in [57], we focus our efforts on improving the discrepancy measure which plays a central role in such within-network adversarial learning-based approach. Our method aims to minimize the cost of moving the marginal distributions between the task-specific classifiers by utilizing the Wasserstein metric [43, 27, 1], which provides a more meaningful notion of dissimilarity for probability distributions. We make several key contributions in this work: (1) a novel and principled method for aligning feature distributions between domains via optimal transport theory (*i.e.*, Wasserstein distance) and the task-specific decision boundary. (2) enable efficient end-to-end training using sliced Wasserstein discrepancy (a variational formulation of Wasserstein metric). (3) effectively harness the geometry of the underlying manifold created by optimizing the sliced Wasserstein discrepancy in an adversarial manner. (4) the method advances the state-of-the-art across several tasks and can be readily applied to any domain adaptation problem such as image classification, semantic segmentation, and object detection.

2. Related Work

A rich body of approaches to unsupervised domain adaptation aim to reduce the gap between the source and target domains by learning domain-invariant feature representations, through various statistical moment matching techniques. Some methods utilize maximum mean discrepancy (MMD) [38, 39] to match the hidden representations of cer-

tain layers in a deep neural network. Other approaches uses Central moment discrepancy (CMD) method [75] to explicitly match each order and each hidden coordinate of higher order moments. Adaptive batch normalization (Ad-aBN) [33] has also been proposed to modulate the statistics in all batch normalization layers across the network between domains.

Another family of strategies tackles the domain adaptation problem by leveraging the adversarial learning behavior of GANs [19]. Such technique was first used at *feature-level* where a domain discriminator is trained to correctly classify the domain of each input feature and the feature generator is trained to deceive the domain discriminator so that the resulting feature distribution is made domain invariant [71, 24, 18]. Later the technique was applied at *pixel-level* to perform distribution alignment in raw input space, translating source domain to the “style” of target domain and obtaining models trained on transformed source data [36, 70, 5, 62, 23, 59, 45]. Recently the technique was used at *output-level* by assuming that the output space contains similar spatial structure for some specific tasks such as semantic segmentation. The method in [69] thereby aligns pixel-level ground truth through adversarial learning in the output space. Other hybrid approaches have also been proposed in [60, 25].

In contrast, Saito *et al.* in [57] proposed to align distributions by explicitly utilizing task-specific classifiers as a discriminator. The framework maximizes the discrepancy between two classifiers’ output to detect target samples that are outside the support of the source and then minimizes the discrepancy to generate feature representations that are inside the support of the source with respect to the decision boundary. Instead of aligning manifold in feature, input, or output space by heuristic assumptions, this approach focuses on directly reshaping the target data regions that indeed need to be reshaped.

Wasserstein metric, the natural geometry for probability measures induced by the optimal transport theory, has been investigated in several fields such as image retrieval [55], color-based style transfer [51], and image warping [21]. The Wasserstein distance has also recently raised interest in stabilizing generative modeling [1, 14, 73], learning introspective neural networks [32], and obtaining Gaussian mixture models [29] thanks to its geometrically meaningful distance measure even when the supports of the distributions do not overlap.

As for domain adaptation, Courty *et al.* in [10] first learn a transportation plan matching source and target samples with class regularity. JDOT method [9] learns to map input space from source to target by jointly considering the class regularity and feature distribution. DeepJDOT method [12] further improves upon JDOT by jointly matching feature and label space distributions with more discriminative fea-

ture representations in a deep neural network layer. However, the fact that these approaches explicitly enforce an one-to-one mapping between source samples and target samples in label space could largely restrict the practical usages when balanced source-target pairs are unavailable. It is also unclear how to extend these approaches to more generic tasks when one data sample has structured output space such as pixel-wise semantic segmentation.

In this work, we propose a principled framework to marry the two powerful concepts: distribution alignment by task-specific decision boundary [57] and the Wasserstein distance [72]. The Wasserstein metric serves as a reliable discrepancy measure between the task-specific classifiers, which directly measures the support of target samples from source samples instead of producing explicit one-to-one mapping in label space. A variational version of the Wasserstein discrepancy further provides straightforward and geometrically meaningful gradients to jointly train the feature generator and classifiers in the framework efficiently.

3. Method

We first introduce unsupervised domain adaptation setting in Section 3.1. Second, we briefly review the concept of optimal transport in Section 3.2. Finally, we detail how to train the proposed method with the sliced Wasserstein discrepancy in Section 3.3.

3.1. Framework Setup

Given input data \mathbf{x}_s and the corresponding ground truth \mathbf{y}_s drawn from the source set $\{X_s, Y_s\}$, and input data \mathbf{x}_t drawn from the target set X_t , the goal of unsupervised domain adaptation is to establish knowledge transfer from the labeled source set to the unlabeled target set as mentioned in [47]. When the two data distributions X_s and X_t are close enough, one can simply focus on minimizing an empirical risk of the joint probability distribution $\mathcal{P}(X_s, Y_s)$. However, when that the two distributions are substantially different, optimizing a model solely over the source information results in poor generalizability.

Following the Maximum Classifier Discrepancy (MCD) framework [57], we train a feature generator network G and the classifier networks C_1 and C_2 , which take feature responses generated from G and produce the corresponding logits $p_1(\mathbf{y}|\mathbf{x})$, $p_2(\mathbf{y}|\mathbf{x})$ respectively (as shown in Figure 1). The optimization procedure consists of three steps: (1) train both generator G and classifiers (C_1, C_2) on the source domain $\{X_s, Y_s\}$ to classify or regress the source samples correctly,

$$\min_{G, C_1, C_2} \mathcal{L}_s(X_s, Y_s), \quad (1)$$

where \mathcal{L}_s can be any loss functions of interest such as cross entropy loss or mean squared error loss.

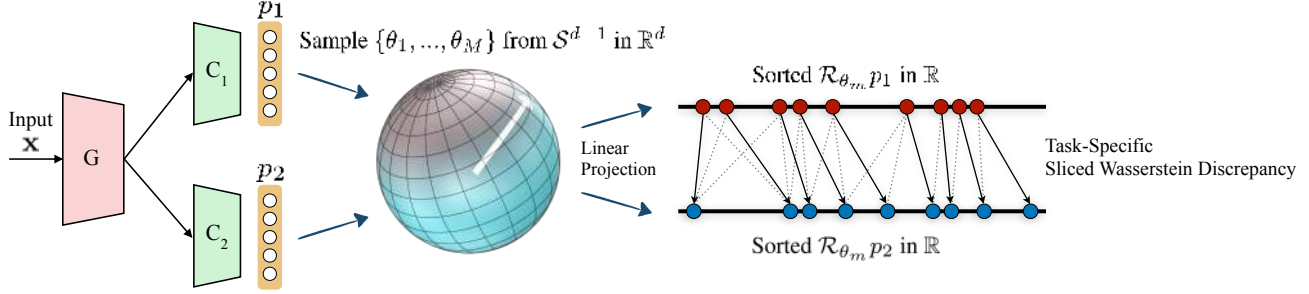


Figure 1: An illustration of the proposed sliced Wasserstein discrepancy (SWD) computation. The SWD is designed to capture the dissimilarity of probability measures p_1 and p_2 in \mathbb{R}^d between the task-specific classifiers C_1 and C_2 , which take input from feature generator G . The SWD enables end-to-end training directly through a variational formulation of Wasserstein metric using radial projections on the uniform measures on the unit sphere \mathcal{S}^{d-1} , providing a geometrically meaningful guidance to detect target samples that are far from the support of the source. Please refer to Section 3.3 for details.

(2) freeze the parameters of the generator G and update the classifiers (C_1, C_2) to maximize the discrepancy between the outputs of the two classifiers on the target set X_t , identifying the target samples that are outside the support of task-specific decision boundaries,

$$\min_{C_1, C_2} \mathcal{L}_s(X_s, Y_s) - \mathcal{L}_{\text{DIS}}(X_t) \quad (2)$$

where $\mathcal{L}_{\text{DIS}}(X_t)$ is the discrepancy loss (L_1 in [57]). $\mathcal{L}_s(X_s, Y_s)$ is also added to this step to retain information from the source domain, and

(3) freeze the parameters of the two classifiers and update the generator G to minimize the discrepancy between the outputs of the two classifiers on the target set X_t ,

$$\min_G \mathcal{L}_{\text{DIS}}(X_t) \quad (3)$$

This step brings the target feature manifold closer to the source.

3.2. Optimal Transport and Wasserstein Distance

The effectiveness of domain adaptation in the aforementioned MCD framework depends entirely on the reliability of the discrepancy loss. Learning without the discrepancy loss, essentially dropping step 2 and step 3 in the training procedure, is simply supervised learning on the source domain.

The Wasserstein distance has recently received great attention in designing loss functions for its superiority over other probability measures [73, 41]. In comparison to other popular probability measures such as total variation distance, Kullback-Leibler divergence, and Jensen-Shannon divergence that compare point-wise histogram embeddings alone, Wasserstein distance takes into account the properties of the underlying geometry of probability space and it is even able to compare distribution measures that do not share support [1]. Motivated by the advantages of the Wasserstein distance, we now describe how we leverage this metric for measuring the discrepancy in our method.

Let Ω be a probability space and μ, ν be two probability measures in $\mathcal{P}(\Omega)$, the Monge problem [43] seeks a transport map $\mathcal{T} : \Omega \rightarrow \Omega$ that minimizes the cost

$$\inf_{\mathcal{T}_{\#}\mu = \nu} \int_{\Omega} c(\mathbf{z}, \mathcal{T}(\mathbf{z})) d\mu(\mathbf{z}), \quad (4)$$

where $\mathcal{T}_{\#}\mu = \nu$ denotes a one-to-one push-forward from μ toward $\nu \forall$ Borel subset $A \subset \Omega$ and $c : \Omega \times \Omega \rightarrow \mathbb{R}^+$ is a geodesic metric that can be either linear or quadratic. However, the solution \mathcal{T}^* may not always exist due to the assumption of no splitting of the probability measures, for example when pushing a Dirac measure toward a non-Dirac measure.

Kantorovitch [27] proposed a relaxed version of Eq 4, which seeks a transportation plan of a joint probability distribution $\gamma \in \mathcal{P}(\Omega \times \Omega)$ such that

$$\inf_{\gamma \in \Pi(\mu, \nu)} \int_{\Omega \times \Omega} c(\mathbf{z}_1, \mathbf{z}_2) d\gamma(\mathbf{z}_1, \mathbf{z}_2), \quad (5)$$

where $\Pi(\mu, \nu) = \{\gamma \in \mathcal{P}(\Omega \times \Omega) | \pi_{1\#}\gamma = \mu, \pi_{2\#}\gamma = \nu\}$ and π_1 and π_2 denote the two marginal projections of $\Omega \times \Omega$ to Ω . The solutions γ^* are called optimal transport plans or optimal couplings [72].

For $q \geq 1$, the q -Wasserstein distance between μ and ν in $\mathcal{P}(\Omega)$ is defined as

$$W_q(\mu, \nu) = \left(\inf_{\gamma \in \Pi(\mu, \nu)} \int_{\Omega \times \Omega} c(\mathbf{z}_1, \mathbf{z}_2)^q d\gamma(\mathbf{z}_1, \mathbf{z}_2) \right)^{1/q}, \quad (6)$$

which is the minimum cost induced by the optimal transportation plan. In our method, we use the 1-Wasserstein distance, also called the earth mover's distance (EMD).

3.3. Learning with Sliced Wasserstein Discrepancy

In this work, we propose to apply 1-Wasserstein distance to the domain adaptation framework described in Section 3.1. We utilize the geometrically meaningful 1-Wasserstein distance as the discrepancy measure in step 2

Algorithm 1 Sliced Wasserstein Discrepancy for Unsupervised Domain Adaptation

Require: Labeled source set $\{X_s, Y_s\}$, unlabeled target set X_t , number of random projections M , and randomly initialized feature generator G and classifiers C_1, C_2 .

while G, C_1 , and C_2 have not converged **do**

Step 1: Train G, C_1 , and C_2 on labeled source set:

$$\min_{G, C_1, C_2} \mathcal{L}_s(X_s, Y_s)$$

Step 2: Train C_1 and C_2 to maximize sliced Wasserstein discrepancy on unlabeled target set:

Obtain classifiers' output p_1 and p_2 on target samples

Sample $\{\theta_1, \dots, \theta_M\}$ from \mathcal{S}^{d-1} in \mathbb{R}^d

Sort $\mathcal{R}_{\theta_m} p_1$ such that $\mathcal{R}_{\theta_m} p_{1(j)} \leq \mathcal{R}_{\theta_m} p_{1(j+1)}$

Sort $\mathcal{R}_{\theta_m} p_2$ such that $\mathcal{R}_{\theta_m} p_{2(j)} \leq \mathcal{R}_{\theta_m} p_{2(j+1)}$

$$\min_{C_1, C_2} \mathcal{L}_s(X_s, Y_s) - \mathcal{L}_{\text{DIS}}(X_t)$$

where $\mathcal{L}_{\text{DIS}}(X_t) = \sum_m \sum_j c(\mathcal{R}_{\theta_m} p_{1(j)}, \mathcal{R}_{\theta_m} p_{2(j)})$

Step 3: Train G to minimize the same sliced Wasserstein discrepancy on unlabeled target set:

$$\min_G \mathcal{L}_{\text{DIS}}(X_t)$$

end while

and step 3 in the aforementioned framework. In practice, we consider the discrete version of classifiers' logits $p_1(\mathbf{y}|\mathbf{x})$ and $p_2(\mathbf{y}|\mathbf{x})$. Computing $W_1(p_1, p_2)$ requires obtaining the optimal transport coupling γ^* by solving a linear programming problem [27], which is not efficient. Although various optimization approaches have been proposed in the past [11, 16], it is unclear how we can directly optimize $W_1(p_1, p_2)$ in an end-to-end trainable fashion efficiently.

To take advantage of the best of both worlds – to align distributions of source and target by utilizing the task-specific decision boundaries and to incorporate the Wasserstein discrepancy, which has well-behaved energy landscape for stochastic gradient descent training, we propose to integrate $W_1(p_1, p_2)$ into our framework by using the sliced Wasserstein discrepancy, a 1-D variational formulation of the 1-Wasserstein distance between the outputs p_1 and p_2 of the classifiers along the radial projections.

Motivated by [52] which defines a sliced barycenter of discrete measures, we define the sliced 1-Wasserstein discrepancy (SWD) as

$$SWD(\mu, \nu) = \int_{\mathcal{S}^{d-1}} W_1(\mathcal{R}_\theta \mu, \mathcal{R}_\theta \nu) d\theta, \quad (7)$$

where \mathcal{R}_θ denotes a one-dimensional linear projection operation on the probability measure μ or ν , and θ is a uniform measure on the unit sphere \mathcal{S}^{d-1} in \mathbb{R}^d such that $\int_{\mathcal{S}^{d-1}} d\theta = 1$. In this manner, computing the sliced Wasserstein discrepancy is equivalent to solving several one-dimensional optimal transport problems which have closed-form solutions [52].

Specifically, let α and β be the permutations that order the N one-dimensional linear projections of N samples such that $\forall 0 \leq i < N - 1, \mathcal{R}_\theta \mu_{\alpha(i)} \leq \mathcal{R}_\theta \mu_{\alpha(i+1)}$

and $\mathcal{R}_\theta \nu_{\beta(i)} \leq \mathcal{R}_\theta \nu_{\beta(i+1)}$, then the optimal coupling γ^* that minimizes such one-dimensional Wasserstein distance is simply assign $\mathcal{R}_\theta \mu_{\alpha(i)}$ to $\mathcal{R}_\theta \nu_{\beta(i)}$ using a sorting algorithm. For discrete probability measures, our SWD can be written as:

$$SWD(\mu, \nu) = \sum_{m=1}^M \sum_{i=1}^N c(\mathcal{R}_{\theta_m} \mu_{\alpha(i)}, \mathcal{R}_{\theta_m} \nu_{\beta(i)}) \quad (8)$$

for M randomly sampled θ and quadratic loss for c unless otherwise mentioned. Our proposed SWD is essentially a variational version of original Wasserstein distance but at a fraction of its computational cost [4]. More importantly, the SWD is differentiable due to the close-form characteristic, so we can focus on using optimal transport as a reliable fidelity measure to guide the optimization of feature generator and classifiers. We summarize our framework in Algorithm 1 and illustrate the SWD computation in Figure 1.

4. Experiments

In principle, our method can be applied to any domain adaptation tasks and does not require any similarity assumptions in input or output space. We perform extensive evaluation of the proposed method on digit and sign recognition, image classification, semantic segmentation, and object detection tasks.

4.1. Digit and Sign Recognition

In this experiment, we evaluate our method using five standard benchmark datasets: Street View House Numbers (SVHN) [46], MNIST [31], USPS [26], Synthetic Traffic Signs (SYNSIG) [42], and German Traffic Sign Recognition Benchmark (GTSRB) [65] datasets. For each domain shift pair, we use the exact CNN architecture provided by Saito *et al.* [57]. We use Adam [28] solver with mini-batch size of 128 in all experiments. Gradient reversal layer (GRL) [17] is used for training the networks so we do not need to control the update frequency between the generator and classifiers. The hyper-parameter particular to our method is the number of radial projections M . We varied the value of M in our experiment and detailed the sensitivity to the hyper-parameter in Figure 2(a) and 2(b).

SVHN \rightarrow MNIST We first examine the adaptation from real-world house numbers obtained from Google Street View images [46] to handwritten digits [31]. The two domains demonstrate distinct distributions because images from SVHN dataset contain cluttered background from streets and cropped digits near the image boundaries. We use the standard training set as training samples, and testing set as test samples both for source and target domains. The feature generator contains three 5×5 conv layers with stride two 3×3 max pooling placed after the first two conv layers. For classifiers, we use 3-layered fully-connected networks.

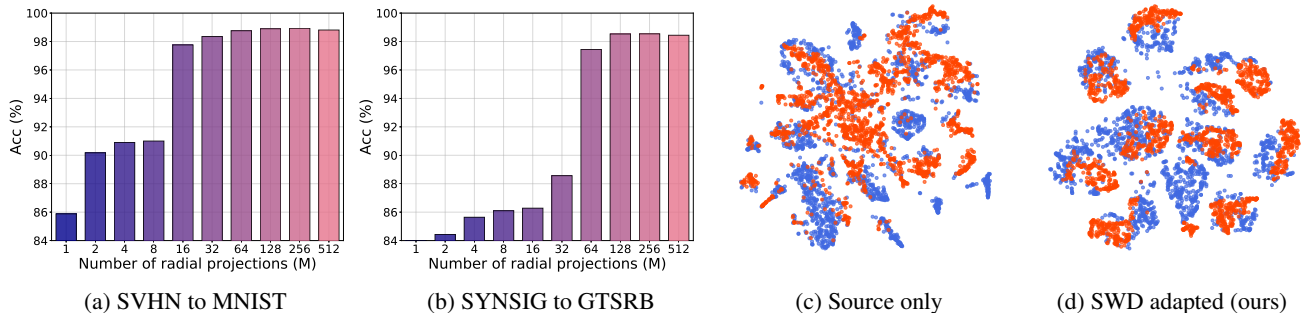


Figure 2: The effect of number of radial projections M to accuracy on (a) SVHN to MNIST adaptation, and (b) SYNSIG to GTSRB adaptation. $M = 128$ is sufficient for stable optimization and good accuracies. T-SNE [40] visualization of features obtained from SVHN to MNIST experiment by (c) source domain only, and (d) SWD adaptation. Blue and red points denote the source and target samples, respectively. Our method generates much more discriminative feature representation compared to source only setting.

SYNSIG \rightarrow GTSRB In this setting, we evaluate the adaptation ability from synthetic images SYNSIG to real images GTSRB. We randomly selected 31367 samples for target training and evaluated the accuracy on the rest. The feature generator contains three 5×5 conv layers with stride two 2×2 max pooling placed after each conv layer. For classifiers, we use 2-layered fully-connected networks. The performance is evaluated for 43 common classes between the two domains.

MNIST \leftrightarrow USPS For the two-way domain shift experiment we also follow the protocols provided by [57] that we use the standard training set as training samples, and testing set as test samples both for source and target domains. The feature generator contains two 5×5 conv layers with stride two 2×2 max pooling placed after the each conv layer. For classifiers, we use 3-layered fully-connected networks.

Results Table 1 lists the accuracies for the target samples by four different domain shifts. We observed that our SWD method outperforms competing approaches in all settings. The proposed method also outperforms the direct comparable method MCD [57] by a large margin – absolute accuracy improvement of 2.8% on average across the four settings. Figure 2(a) and 2(b) show the ablation study on the sensitivity to the number of radial projections M . In our experiment we empirically found that $M = 128$ works well in all cases. We also visualized learned features in Figure 2(c) and 2(d). Our method generates much more discriminative feature representation compared to source only setting.

It is interesting to see that the task-specific discrepancy-aware methods such as MCD [57], DeepJDOT [12], and the proposed SWD are the current leading approaches for the tasks being addressed here. This demonstrates the importance of utilizing the task-specific decision boundaries (discrepancy) to guide the process of transfer learning instead of simply matching the distributions between the source and target domains in pixel, feature, or output space in most of the other distribution matching approaches. In particular,

| Method | SVHN \downarrow MNIST | SYNSIG \downarrow GTSRB | MNIST \downarrow USPS | USPS \downarrow MNIST |
|-----------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Source only | 67.1 | 85.1 | 79.4 | 63.4 |
| MMD [38] | 71.1 | 91.1 | 81.1 | - |
| DANN [17] | 71.1 | 88.7 | 85.1 | 73.0 ± 0.2 |
| DSN [6] | 82.7 | 93.1 | - | - |
| ADDA [70] | 76.0 ± 1.8 | - | - | 90.1 ± 0.8 |
| CoGAN [36] | - | - | - | 89.1 ± 0.8 |
| PixelDA [5] | - | - | 95.9 | - |
| ASSC [20] | 95.7 ± 1.5 | 82.8 ± 1.3 | - | - |
| UNIT [35] | 90.5 | - | 96.0 | 93.6 |
| CyCADA [23] | 90.4 ± 0.4 | - | 95.6 ± 0.2 | 96.5 ± 0.1 |
| I2I Adapt [45] | 92.1 | - | 95.1 | 92.2 |
| GenToAdapt [59] | 92.4 ± 0.9 | - | 95.3 ± 0.7 | 90.8 ± 1.3 |
| MCD [57] | 96.2 ± 0.4 | 94.4 ± 0.3 | 96.5 ± 0.3 | 94.1 ± 0.3 |
| DeepJDOT [12] | 96.7 | - | 95.7 | 96.4 |
| SWD (ours) | 98.9 ± 0.1 | 98.6 ± 0.3 | 98.1 ± 0.1 | 97.1 ± 0.1 |

Table 1: Results of unsupervised domain adaptation across digit and traffic sign datasets. We repeat each experiment 5 times and report the mean and the standard deviation of the accuracy. Our method significantly outperforms the direct comparable method MCD [57] and other methods as well.

the adversarial training based methods require a separate generator and multiple discriminators that are oftentimes larger than the main task network itself. For example, the method in [23] uses a 10-layer generator, a 6-layer image level discriminator, and a 3-layer feature level discriminator while the main task network is a 4-layer network. Besides, the auxiliary discriminators are discarded after the training is completed.

Furthermore, the main distinction between the proposed method and the DeepJDOT [12] approach is that the DeepJDOT requires a multi-staged training process – it trains a CNN and solves a linear programming task iteratively. The DeepJDOT also assumes the true optimal transport coupling between every pair of samples in a mini-batch converges when propagating pseudo labels from the source domain to target domain, which is often not the case in prac-

| Method | plane | beycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck | mean |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Source only | 55.1 | 53.3 | 61.9 | 59.1 | 80.6 | 17.9 | 79.7 | 31.2 | 81.0 | 26.5 | 73.5 | 8.5 | 52.4 |
| MMD [38] | 87.1 | 63.0 | 76.5 | 42.0 | 90.3 | 42.9 | 85.9 | 53.1 | 49.7 | 36.3 | 85.8 | 20.7 | 61.1 |
| DANN [17] | 81.9 | 77.7 | 82.8 | 44.3 | 81.2 | 29.5 | 65.1 | 28.6 | 51.9 | 54.6 | 82.8 | 7.8 | 57.4 |
| MCD [57] | 87.0 | 60.9 | 83.7 | 64.0 | 88.9 | 79.6 | 84.7 | 76.9 | 88.6 | 40.3 | 83.0 | 25.8 | 71.9 |
| SWD (ours) | 90.8 | 82.5 | 81.7 | 70.5 | 91.7 | 69.5 | 86.3 | 77.5 | 87.4 | 63.6 | 85.6 | 29.2 | 76.4 |

Table 2: Results of unsupervised domain adaptation on VisDA 2017 [49] image classification track. Accuracies are obtained by fine-tuning ResNet-101 model pre-trained on ImageNet. This task evaluates the adaptation capability from synthetic CAD model images to real-world MSCOCO images. Our method outperforms the direct comparable method MCD [57] and other methods as well.

tice. This emphasizes the importance of choosing a geometrically meaningful discrepancy measure that makes no assumptions on the optimal transport coupling in the label space and is end-to-end trainable, optimizing over one discrepancy loss instead of solving multiple losses independently.

We note that the method in [63] obtained 99.4% on SVHN to MNIST adaptation task with various engineering efforts such as using instance normalization, adding Gaussian noise, and leveraging a much deeper 18-layer network. This clustering assumption-based approach achieves performance on-par with ours, and we leave this architectural search for future exploration.

4.2. VisDA Image Classification

Next, we evaluate the proposed method on image classification task. We use the VisDA dataset [49], which is designed to assess the domain adaptation capability from synthetic to real images across 12 object classes. The source domain contains 152,397 synthetic images generated by rendering 3D CAD models from different angles and under different lighting conditions. For target domain we use the validation set collected from MSCOCO [34] and it consists of 55,388 real images. Following the protocol in [57], we evaluate our method by fine-tuning ImageNet [13] pre-trained ResNet-101 [22] model. The ResNet model except the last fully-connected layer was used as our feature generator and randomly initialized three-layered fully-connected networks were used as our classifiers. We used Adam solver with mini-batch size of 32 in the experiment. The number of radial projections M is set to 128. We apply horizontal flipping of input images during training as the only data augmentation. Learning rate was set to value of 10^{-6} throughout the training. GRL [17] is used for training the network so we do not need to control the update frequency between the generator and classifiers.

Results Table 2 lists the results that are based on the same evaluation protocol¹. We can see that the task-specific discrepancy-aware methods MCD [57] and our

¹Method in [15] involves various data augmentation including scaling, cropping, flipping, rotation, brightness and color space perturbation, etc. The authors reported mean accuracy of 74.2% in the minimal augmentation setting using a much deeper ResNet-152 backbone.

SWD method perform better than the source only model in all object classes, while pure distribution matching based methods perform worse than the source only model in some categories. Our method also outperforms the direct comparable method MCD [57] by a large margin. We emphasize that the main difference between MCD and our method is the choice of discrepancy loss. This validated the effectiveness of the proposed sliced Wasserstein discrepancy in this challenging synthetic to real image adaption task.

4.3. Semantic Segmentation

Unlike image classification task, obtaining ground truth label for each pixel in an image requires a lot more amount of human labor. Here we extend our framework to perform domain adaptation for the semantic segmentation task.

Datasets In this experiment we used three benchmark datasets: GTA5 [53], Synthia [54], and Cityscapes [8]. All three datasets contain dense pixel-level semantic annotations that are compatible with one another. GTA5 contains 24966 vehicle-egocentric images synthesized from a photorealistic open-world computer game Grand Theft Auto V. Synthia consists of 9400 images generated by rendering a virtual city created with the Unity engine. Frames in Synthia are acquired from multiple camera viewpoints – up to eight views per location that are not necessary vehicle-egocentric. Cityscapes has 2975 training images, 500 validation images, and 1525 test images with dense pixel-level annotation of urban street scenes in Germany and neighboring countries. During training, the synthetic GTA5 or Synthia is used as source domain and the real Cityscapes is used as target domain. We used the standard training split for training and validation split for evaluation purpose.

Implementation details In an effort to demonstrate the effectiveness of the proposed method and to decouple the performance gain due to architectural search, we adopted the commonly used VGG-16 [64] and ResNet-101 [22] models for our feature generator. For classifiers we used the decoder in PSPNet [77] for its simplicity of implementation. Based on this straightforward design choice, our segmentation model achieves mean intersection-over-union (mIoU) of 60.5% for VGG-16 backbone and 64.1% for ResNet-101 backbone when trained on the Cityscapes training set and

| Method | road | sdwk | blndg | wall | fence | pole | light | sign | vgtrn | trm | sky | person | rider | car | truck | bus | train | mcycl | beycl | mIoU |
|-------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|------------|-------------|-------------|-------------|
| Source only (VGG16) | 25.9 | 10.9 | 50.5 | 3.3 | 12.2 | 25.4 | 28.6 | 13.0 | 78.3 | 7.3 | 63.9 | 52.1 | 7.9 | 66.3 | 5.2 | 7.8 | 0.9 | 13.7 | 0.7 | 24.9 |
| FCN Wld [24] | 70.4 | 32.4 | 62.1 | 14.9 | 5.4 | 10.9 | 14.2 | 2.7 | 79.2 | 21.3 | 64.6 | 44.1 | 4.2 | 70.4 | 8.0 | 7.3 | 0.0 | 3.5 | 0.0 | 27.1 |
| MCD [57] | 86.4 | 8.5 | 76.1 | 18.6 | 9.7 | 14.9 | 7.8 | 0.6 | 82.8 | 32.7 | 71.4 | 25.2 | 1.1 | 76.3 | 16.1 | 17.1 | 1.4 | 0.2 | 0.0 | 28.8 |
| CDA [76] | 74.9 | 22.0 | 71.7 | 6.0 | 11.9 | 8.4 | 16.3 | 11.1 | 75.7 | 13.3 | 66.5 | 38.0 | 9.3 | 55.2 | 18.8 | 18.9 | 0.0 | 16.8 | 14.6 | 28.9 |
| AdaSegNet [69] | 87.3 | 29.8 | 78.6 | 21.1 | 18.2 | 22.5 | 21.5 | 11.0 | 79.7 | 29.6 | 71.3 | 46.8 | 6.5 | 80.1 | 23.0 | 26.9 | 0.0 | 10.6 | 0.3 | 35.0 |
| CyCADA [23] | 85.2 | 37.2 | 76.5 | 21.8 | 15.0 | 23.8 | 22.9 | 21.5 | 80.5 | 31.3 | 60.7 | 50.5 | 9.0 | 76.9 | 17.1 | 28.2 | 4.5 | 9.8 | 0.0 | 35.4 |
| CBST [78] | 90.4 | 50.8 | 72.0 | 18.3 | 9.5 | 27.2 | 28.6 | 14.1 | 82.4 | 25.1 | 70.8 | 42.6 | 14.5 | 76.9 | 5.9 | 12.5 | 1.2 | 14.0 | 28.6 | 36.1 |
| DCAN [74] | 82.3 | 26.7 | 77.4 | 23.7 | 20.5 | 20.4 | 30.3 | 15.9 | 80.9 | 25.4 | 69.5 | 52.6 | 11.1 | 79.6 | 24.9 | 21.2 | 1.30 | 17.0 | 6.70 | 36.2 |
| SWD (ours) | 91.0 | 35.7 | 78.0 | 21.6 | 21.7 | 31.8 | 30.2 | 25.2 | 80.2 | 23.9 | 74.1 | 53.1 | 15.8 | 79.3 | 22.1 | 26.5 | 1.5 | 17.2 | 30.4 | 39.9 |
| Source only (ResNet101) | 75.8 | 16.8 | 77.2 | 12.5 | 21.0 | 25.5 | 30.1 | 20.1 | 81.3 | 24.6 | 70.3 | 53.8 | 26.4 | 49.9 | 17.2 | 25.9 | 6.5 | 25.3 | 36.0 | 36.6 |
| AdaSegNet [69] | 86.5 | 36.0 | 79.9 | 23.4 | 23.3 | 23.9 | 35.2 | 14.8 | 83.4 | 33.3 | 75.6 | 58.5 | 27.6 | 73.7 | 32.5 | 35.4 | 3.9 | 30.1 | 28.1 | 42.4 |
| SWD (ours) | 92.0 | 46.4 | 82.4 | 24.8 | 24.0 | 35.1 | 33.4 | 34.2 | 83.6 | 30.4 | 80.9 | 56.9 | 21.9 | 82.0 | 24.4 | 28.7 | 6.1 | 25.0 | 33.6 | 44.5 |

Table 3: Adaptation results from GTA5 to Cityscapes. We compare our results with other state-of-the-art approaches that are based on the standard VGG-16 or ResNet-101 backbone.

| Method | road | sdwk | blndg | light | sign | vgtrn | sky | person | rider | car | bus | mcycl | beycl | mIoU |
|-------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Source only (VGG16) | 6.4 | 17.7 | 29.7 | 0.0 | 7.2 | 30.3 | 66.8 | 51.1 | 1.5 | 47.3 | 3.9 | 0.1 | 0.0 | 20.2 |
| FCN Wld [24] | 11.5 | 19.6 | 30.8 | 0.1 | 11.7 | 42.3 | 68.7 | 51.2 | 3.8 | 54.0 | 3.2 | 0.2 | 0.6 | 22.9 |
| Cross-city [7] | 62.7 | 25.6 | 78.3 | 1.2 | 5.4 | 81.3 | 81.0 | 37.4 | 6.4 | 63.5 | 16.1 | 1.2 | 4.6 | 35.7 |
| CBST [78] | 69.6 | 28.7 | 69.5 | 11.9 | 13.6 | 82.0 | 81.9 | 49.1 | 14.5 | 66.0 | 6.6 | 3.7 | 32.4 | 36.1 |
| AdaSegNet [69] | 78.9 | 29.2 | 75.5 | 0.1 | 4.8 | 72.6 | 76.7 | 43.4 | 8.8 | 71.1 | 16.0 | 3.6 | 8.4 | 37.6 |
| SWD (ours) | 83.3 | 35.4 | 82.1 | 12.2 | 12.6 | 83.8 | 76.5 | 47.4 | 12.0 | 71.5 | 17.9 | 1.6 | 29.7 | 43.5 |
| Source only (ResNet101) | 55.6 | 23.8 | 74.6 | 6.1 | 12.1 | 74.8 | 79.0 | 55.3 | 19.1 | 39.6 | 23.3 | 13.7 | 25.0 | 38.6 |
| AdaSegNet [69] | 79.2 | 37.2 | 78.8 | 9.9 | 10.5 | 78.2 | 80.5 | 53.5 | 19.6 | 67.0 | 29.5 | 21.6 | 31.3 | 45.9 |
| SWD (ours) | 82.4 | 33.2 | 82.5 | 22.6 | 19.7 | 83.7 | 78.8 | 44.0 | 17.9 | 75.4 | 30.2 | 14.4 | 39.9 | 48.1 |

Table 4: Adaptation results from Synthia to Cityscapes. We compare our results with other state-of-the-art approaches that are based on the standard VGG-16 or ResNet-101 backbone.

evaluated on the Cityscapes validation set for 19 compatible classes, which match the same oracle performance reported in the recent literature [69, 74].

We used Momentum SGD solver with a fixed momentum of 0.9 and weight decay of 0.0001 in all experiments. Learning rate was set to value of 0.0001 for GTA5 to Cityscapes setting and 0.001 for Synthia to Cityscapes setting. During training, we randomly sampled a single image from both source and target domains for each mini-batch optimization. All images are resized to 1024×512 resolution. No data augmentation is used (such as flipping, cropping, scaling, and multi-scale ensemble) to minimize the performance gain due to the engineering efforts and to ensure the reproducibility. Since the sliced Wasserstein discrepancy is computed at every pixel of an image in a mini-batch fashion, we empirically found that the number of radial projections $M = 8$ is sufficient to produce good results.

Results We use the evaluation protocol released along with VisDA challenge [49] to calculate the PASCAL VOC intersection-over-union (IoU). We show quantitative and qualitative results of adapting GTA5 to Cityscapes in Table 3 and Figure 3, respectively. We can see clear improvement from models trained on source domain only to models trained with the proposed SWD method for both VGG-16 and ResNet-101 backbones. Also, our method consistently outperforms other recent approaches that utilize generative

adversarial networks [69, 23] and the style transfer based technique [74].

Table 4 shows the results of adapting Synthia to Cityscapes. The domain shift is even more significant between these two datasets because images from Synthia are not only generated by a rendering engine but also contain multiple viewpoints that are not necessary vehicle-egocentric. Our method shows consistent improvement over other approaches and generalizes well with such dramatic viewpoint shift.

Note that most of the competing approaches are specifically designed only for semantic segmentation tasks and they often require the assumption of input space similarity, output space similarity, or geometric constrains. For instance, the method in [78] incorporated spatial priors by assuming sky is likely to appear at the top and road is likely to appear at the bottom of an image, etc. The frequencies of ground truth labels per pixel are computed from GTA5 dataset and are then multiplied with the softmax output of the segmentation network. However, it is unclear how to generalize this prior when large viewpoint differences present between the source and target domains, such as adaptation from Synthia to Cityscapes. Our method does not require any prior assumptions of the characteristics of the task of interest and nevertheless achieves better performance.

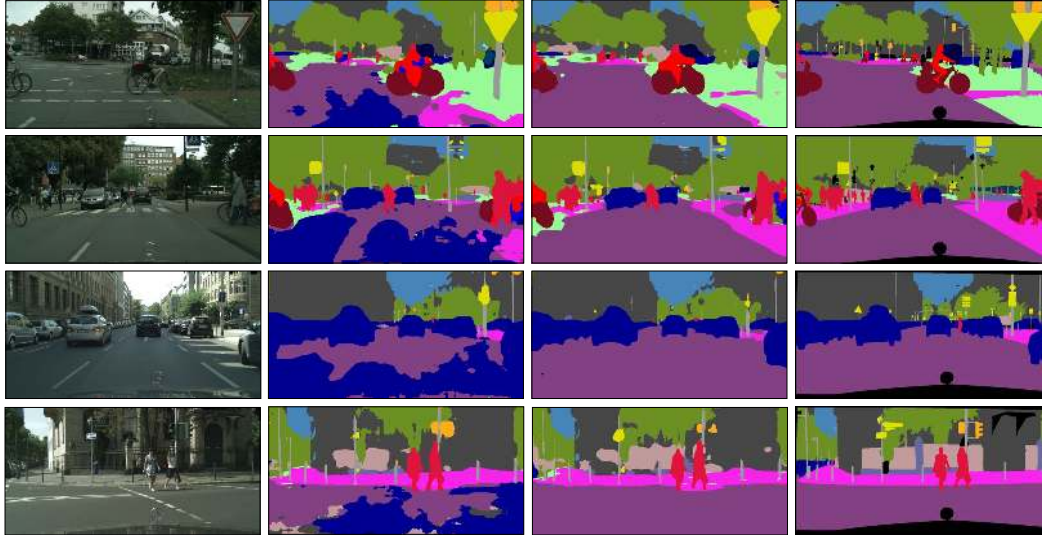


Figure 3: Qualitative semantic segmentation results on adaptation from GTA5 to Cityscapes. From left to right: input, source only model, our method, and ground truth. Our method produces cleaner predictions and less confusion between challenging classes such as road, car, sidewalk, and vegetation.

| Method | plane | bcycl | bus | car | horse | knife | mcycl | person | plant | sktbrd | train | truck | mAP |
|-------------|-------------|------------|-------------|------------|-------------|------------|------------|------------|------------|------------|------------|------------|------------|
| Source only | 0.5 | 0.4 | 9.4 | 9.4 | 7.2 | 0.1 | 1.3 | 4.6 | 0.5 | 0.3 | 1.5 | 0.9 | 3.0 |
| MCD [57] | 11.8 | 1.3 | 10.3 | 3.3 | 10.2 | 0.1 | 8.0 | 2.4 | 0.6 | 1.5 | 5.8 | 1.1 | 4.7 |
| SWD (ours) | 11.9 | 2.0 | 15.5 | 5.4 | 13.1 | 0.1 | 4.7 | 9.8 | 0.9 | 1.0 | 6.2 | 0.7 | 5.9 |

Table 5: Results of unsupervised domain adaptation on VisDA 2018 [50] object detection track. This task evaluates the adaptation capability from synthetic CAD model images to real-world MSCOCO images (COCO17-val). We report mean average precision (mAP) at 0.5 IoU using SSD with Inception-V2 backbone. Our method outperforms the direct comparable method MCD [57] by 25% relatively.

4.4. Object Detection

To demonstrate if our method generalizes to other tasks as well, we extend it to object detection task. We use the recent released VisDA 2018 dataset [50], which contains source images generated by rendering 3D CAD models and target images collected from MSCOCO [34]. This dataset is especially challenging due to uncalibrated object scales and positions between the synthetic and real images.

We use a standard off-the-shelf Single Shot Detector (SSD) [37] with Inception-V2 [68] backbone without any architectural modifications or heuristic assumptions. The model predicts class labels, locations and size shifts for a total of 1.9k possible anchor boxes. The feature generator in this case is the backbone network pre-trained on ImageNet and the classifiers comprise of all the additional layers which are present after the backbone network. We employ the proposed sliced Wasserstein discrepancy to both classification and bounding box regression outputs to the existing loss functions in SSD. No other modifications are made to the network. We also implement MCD [57] method with the exact network architecture for baseline comparison. All networks are optimized with Momentum SGD solver with a fixed momentum of 0.9, mini-batch size of 16, and weight decay of 0.0001. Learning rate is set to value of 0.0001. The number of radial projections M is set to

128. We apply random cropping and flipping to all network training.

Results We report mean average precision (mAP) at 0.5 IoU in Table 5. These results show that even with large domain shift in image realism, object scales, and relative object positions, our method is able to improve the performance by a large margin compared to models trained on source image only. Our method also outperforms the direct comparable method MCD [57] by a significant 25% relatively.

5. Conclusion

In this paper, we have developed a new unsupervised domain adaptation approach, which aligns distributions by measuring sliced Wasserstein discrepancy between task-specific classifiers. The connection to Wasserstein metric paves the way to make better use of its geometrically meaningful embeddings in an efficient fashion, which in the past has primarily been restricted to obtaining one-to-one mapping in label space. Our method is generic and achieves superior results across digit and sign recognition, image classification, semantic segmentation, and object detection tasks. Future work includes extension of our approach to domain randomization [67], open set adaptation [58], and zero-shot domain adaptation [48].

References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 1, 2, 3
- [2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010. 1
- [3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007. 1
- [4] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *JMIV*, 2015. 4
- [5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017. 1, 2, 5
- [6] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NIPS*, 2016. 5
- [7] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017. 7
- [8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6
- [9] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *NIPS*, 2017. 2
- [10] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *TPAMI*, 2016. 2
- [11] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013. 4
- [12] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, 2018. 2, 5
- [13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6
- [14] Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative modeling using the sliced wasserstein distance. In *CVPR*, 2018. 2
- [15] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv:1706.05208*, 2017. 6
- [16] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *NIPS*, 2015. 4
- [17] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2014. 1, 4, 5, 6
- [18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1, 2
- [19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2
- [20] Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *ICCV*, 2017. 5
- [21] Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *IJCV*, 2004. 2
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6
- [23] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 2, 5, 7
- [24] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:1612.02649*, 2016. 1, 2, 7
- [25] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *ECCV*, 2018. 2
- [26] Jonathan J. Hull. A database for handwritten text recognition research. *TPAMI*, 1994. 4
- [27] Leonid Kantorovitch. On the translocation of masses. *Management Science*, 1958. 1, 3, 4
- [28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 4
- [29] Soheil Kolouri, Gustavo K Rohde, and Heiko Hoffmann. Sliced wasserstein distance for learning gaussian mixture models. In *CVPR*, 2018. 2
- [30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1
- [31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 4
- [32] Kwonjoon Lee, Weijian Xu, Fan Fan, and Zhuowen Tu. Wasserstein introspective neural networks. In *CVPR*, 2018. 2
- [33] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 2018. 2
- [34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6, 8
- [35] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. 5
- [36] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NIPS*, 2016. 1, 2, 5

- [37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 8
- [38] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1, 5, 6
- [39] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016. 1
- [40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 5
- [41] Liang Mi, Wen Zhang, Xianfeng Gu, and Yalin Wang. Variational wasserstein clustering. In *ECCV*, 2018. 3
- [42] Boris Moiseev, Artem Konev, Alexander Chigorin, and Anton Konushin. Evaluation of traffic sign recognition methods trained on synthetically generated data. In *ACIVS*. Springer, 2013. 4
- [43] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781. 1, 3
- [44] Jose G Moreno-Torres, Troy Raeder, Rocío Alai-z-Rodríguez, Nitish V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 2012. 1
- [45] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018. 2, 5
- [46] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bis-sacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop*, 2011. 4
- [47] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2010. 2
- [48] Kuan-Chuan Peng, Ziyang Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *ECCV*, 2018. 8
- [49] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv:1710.06924*, 2017. 6, 7
- [50] Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark for synthetic-to-real visual domain adaptation. *arXiv:1806.09755*, 2018. 8
- [51] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to colour transfer. In *ICCV*, 2005. 2
- [52] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernet. Wasserstein barycenter and its application to texture mixing. In *SSVM*, 2011. 4
- [53] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 6
- [54] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 6
- [55] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *ICCV*, 1998. 2
- [56] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 1
- [57] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6, 7, 8
- [58] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *ECCV*, 2018. 8
- [59] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018. 2, 5
- [60] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*, 2018. 2
- [61] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 2000. 1
- [62] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017. 2
- [63] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018. 6
- [64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6
- [65] Johannes Stalkamp, Marc Schlipf, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011. 4
- [66] Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 2009. 1
- [67] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018. 8
- [68] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 8
- [69] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schuster, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 2, 7
- [70] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1, 2, 5
- [71] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv:1412.3474*, 2014. 2

- [72] Cedric Villani. Optimal transport, old and new. *Springer-Verlag*, 2009. [1](#), [2](#), [3](#)
- [73] Jiqing Wu, Zhiwu Huang, Janine Thoma, Dinesh Acharya, and Luc Van Gool. Wasserstein divergence for gans. In *ECCV*, 2018. [2](#), [3](#)
- [74] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *ECCV*, 2018. [7](#)
- [75] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. In *ICLR*, 2017. [2](#)
- [76] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017. [7](#)
- [77] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. [6](#)
- [78] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. [7](#)