Contrastive Test-Time Adaptation

Dian Chen[*]
Toyota Research Institute

Dequan Wang
UC Berkeley

Trevor Darrell
UC Berkeley

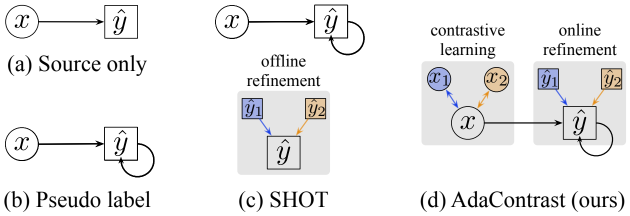
Sayna Ebrahimi[†]
UC Berkeley

Abstract

Test-time adaptation is a special setting of unsupervised domain adaptation where a trained model on the source domain has to adapt to the target domain without accessing source data. We propose a novel way to leverage self-supervised contrastive learning to facilitate target feature learning, along with an online pseudo labeling scheme with refinement that significantly denoises pseudo labels. The contrastive learning task is applied jointly with pseudo labeling, contrasting positive and negative pairs constructed similarly as MoCo but with source-initialized encoder, and excluding same-class negative pairs indicated by pseudo labels. Meanwhile, we produce pseudo labels online and refine them via soft voting among their nearest neighbors in the target feature space, enabled by maintaining a memory queue. Our method, AdaContrast, achieves state-of-the-art performance on major benchmarks while having several desirable properties compared to existing works, including memory efficiency, insensitivity to hyper-parameters, and better model calibration. Code is released at https://github.com/DianCh/AdaContrast.

1. Introduction

Deep networks are remarkably successful in learning visual tasks when training and test data follow the same distribution. However, their ability to generalize to unseen data suffers in the presence of *domain shift* [42, 43]. Building models that can adapt to distribution shifts is the focus of domain adaptation where the goal is to transfer knowledge from a labeled *source* domain to a new but related *target* domain [2, 13, 19, 31, 44, 53]. In this work we focus on the problem of test-time [50, 56] or source-free [23, 28, 58] domain adaptation where the source data is no longer available during adapting to unlabeled test data. Since test-time adaptation (TTA) only requires access to the source model, it is appealing to real-world applications where data privacy and transmission bandwidth become critical issues.



(a) Source only

(b) Pseudo label

(c) SHOT

(d) AdaContrast (ours)

Figure 1. Illustration of how our method, AdaContrast, leverages target domain data vs. prior works. (a) Without adaptation, source-only model is only evaluated on target data. (b) With pseudo labeling, the source classifier predictions are used as pseudo labels for self-training. (c) Existing pseudo labeling approach, SHOT [28], uses offline global refinement to reduce noisy pseudo labels. (d) In AdaContrast, we consider two kinds of relations among target samples: we use contrastive learning to exploit information from sample pairs to learn better target representation, while refining pseudo labels by aggregating knowledge in the neighborhood. Colors indicate pseudo-labeled classes.

However, the challenging setting of TTA raises two major questions: 1) how to learn the target-domain representation without the help of ground truth annotation 2) how to build the target-domain classifier with only the source-domain classifier available as a proxy for the source domain. To address these difficulties, existing works have leveraged image/feature generation [23, 27], class prototypes [28, 57], entropy minimization [28, 56], self-training or pseudo labeling [28], and self-supervised auxiliary task training [50]. Generative models have the drawback of requiring a large computation capacity for generating target-style images/features [27]. Entropy minimization-based methods have been competitive but the direct optimization of entropy disrupts the model calibration on target. Pseudo-labeling methods have shown promising results but their performance can suffer from noisy pseudo labels [28]. Test-time training [50] introduced a self-supervised auxiliary rotation prediction task to be optimized jointly during both source and target training. This approach is limited because it requires altering the source training protocols, which may not be feasible for all models of interest. Moreover, the contrastive learning paradigm has been shown to learn more

[*] Work done when author previously worked at UC Berkeley.
[†] Author is now at Google.

transferable representations compared to rotation prediction as a pre-text task. Recently, [55] used self-supervised learning in the pre-training stage, however, we argue this method does not fully leverage the strength of self-supervised representation learning during the adaptation stage.

In this work, we introduce AdaContrast, a novel test-time adaptation strategy that uses self-supervised contrastive learning on the target domain to exploit the pairwise information among target samples, which is optimized jointly with pseudo labeling. Compared to the pretrain-and-finetune paradigm [4, 7, 17], the joint optimization on target domain allows the model to reuse source knowledge to quickly adapt, while benefiting mutually with pseudo labeling. The intuition is that a better target representation facilitates the learning of the decision boundaries [3], while the useful priors contained in pseudo labels further enhances the effectiveness of contrastive learning in representation learning. We also show that our auxiliary contrastive learning brings robustness to the pseudo labeling, preventing divergence and allowing the online pseudo labels to consistently provide high-accuracy supervision.

As for the pseudo labeling, we introduce a new online pseudo label refinement scheme that results in generating significantly more correct pseudo labels by using soft k-nearest neighbors voting [33] in the target domain's feature space for each target sample. As shown in Fig. 1, unlike prior works which typically require an offline global memory bank to store pseudo labels/features generated every single or a few epochs [28, 55], we produce and refine pseudo labels at a per-batch basis by aggregating probabilities from nearest neighbors based on feature distances. Relying on a relatively small memory queue instead makes our approach both computationally affordable and suitable for online streaming where target data cannot be revisited such as robotics applications.

The two key factors in AdaContrast, self-supervised contrastive learning trained jointly with pseudo labeling, offer several empirical merits including hyper parameter insensitivity and better model calibration. Hyper-parameter selection in TTA setting is a key design factor that is often neglected in TTA literature where tuning hyper-parameters is not an option due to lack of access to target labels. We empirically show our proposed AdaContrast approach consistently performs well under a wide range of hyper-parameters. We also found AdaContrast to have a better model calibration [16, 41] compared to entropy minimization-based methods [28]. We have evaluated our method on major domain adaptation benchmarks where it achieves state-of-the-art test-time adaptation performance. Its 86.8% average accuracy and 84.5% overall accuracy on VisDA-C surpass the previous state-of-the-art by +3.8% and +6.2%, respectively. We are also the first TTA method to evaluate on the large-scale DomainNet dataset, where AdaContrast achieves state-of-the-art 67.8% accuracy averaged over 7 domain shifts.

2. Related Work

Domain adaptation has been extensively explored for many visual tasks, including image classification [54], semantic segmentation [52] and object detection [9]. The goal of unsupervised domain adaptation (UDA) is to close the performance gap when the source model is deployed on a different target domain without any target annotation. Existing works have made tremendous progress revolving around the idea of feature space alignment, with different mechanisms. [39] align the statistics of the distributions, notably the moments at different orders. [31, 54] exploit maximum mean discrepancy. [12] achieve domain confusion by adversarially training the feature encoder and a domain discriminator, whereas GAN-based methods [19] employ generative task to make indistinguishable source and target images. All these methods, however, need to access both source data and target data during the adaptation, making the learning essentially transductive.

Some recent works on source-free/test-time adaptation focus on the more challenging setting where only source model and unlabeled target data are available [23, 27, 27, 28, 56, 57]. TENT [56] introduce entropy minimization as test-time optimization objective. SHOT [28] combined entropy minimization with pseudo labeling. These methods are limited in several aspects. First, the entropy minimization objective does not model the relation among different samples and more importantly, disrupts the model calibration on target data due to direct entropy optimization. Second, the pseudo labels are updated only on a per-epoch basis, which fails to reflect the most recent model improvement during an epoch. In contrast, our method is equipped with contrastive learning for contextual modeling and online pseudo-label for the latest update.

Self-supervised learning methods [3–7, 14, 15, 17, 24, 36, 37, 59] have shown tremendous success in producing transferable visual representations. Researchers have found that contrastive-based proxy tasks [4, 6, 7, 37] can help models to learn a representation that has the potential to replace supervised pre-training (*e.g.*, ImageNet [11]). Consequently, researchers have explored using self-supervised learning for domain adaptation [46, 49, 50, 55]. [46, 49] tackle the unsupervised domain adaptation (UDA) setting where concurrent access to source and target data is allowed. TTT [50] utilizes rotation prediction task as a proxy to update backbone parameters. On-target adaptation [55] leverages contrastive learning to initialize the target-domain feature, as a separate stage in the proposed framework. In contrast, we propose a joint learning approach that combines contrastive learning and pseudo labeling.

Pseudo labeling has been widely adopted in semi-supervised learning [26,48], self-supervised learning [1,3], and domain adaptation [28–30,55]. It is a simple yet effective strategy: for unlabeled samples, the predicted labels or cluster assignment are treated as if they were ground truth labels to provide "supervision". FixMatch [48] is a semi-supervised learning method benefitting from pseudo labeling and consistency regularization. The most recent proposed on-target adaptation [55] augment the FixMatch-style teacher-student learning with contrastive learned target-domain model. Our method utilizes weak-strong consistency as a regularizer while additionally denoise pseudo labels via the proposed online refinement Sec. 3.1.

3. Method

We address the closed-set test-time adaptation problem in image classification where source data is not used during the adaptation. The source model is trained on source pairs of $\{x_s^i, y_s^i\}_{i=1}^{n_s} \in \mathcal{D}_s$ where $x_s^i \in \mathcal{X}_s$ and $y_s^i \in \mathcal{Y}_s$ are images and labels, respectively. Given the trained source model, the goal is to adapt it to unlabeled target data denoted as $\{x_t^i\}_{i=1}^{n_t} \in \mathcal{X}_t$. The underlying labels $\{y_t^i\}_{i=1}^{n_t} \in \mathcal{Y}_t$ are accessed only for evaluation purpose. In the closed-set case the source and target domain share the same label space $\mathcal{Y}_s = \mathcal{Y}_t = \mathcal{Y}$. The source model has a general architecture consisting of a feature extractor $f_s(\cdot) : \mathcal{X}_s \rightarrow \mathbf{R}^D$ and a classifier $h_s(\cdot) : \mathbf{R}^D \rightarrow \mathbf{R}^C$ where D and C are feature dimension and number of classes. To obtain the source model, we follow [28] to first train a model on source data with the standard cross-entropy loss $L_s^{ce} = -\sum_{c=1}^{C} \tilde{y}_s^c \log p_s^c$ where $p_s^c = \sigma_c(h_s(f_s((x_s)))$ is the c-th element of the model's output after softmax operation $\sigma_c(a) = \frac{\exp(a_c)}{\sum_{k=1}^{C} \exp{(a_k)}}$, and $\tilde{y}_s^c$ is the c-th element of the converted one-hot label with label-smoothing [51]: $\tilde{y}_s^c = (1-\alpha)y_s^c + \alpha/C$, where $\alpha = 0.1$ is the smoothness coefficient.

In the test-time adaptation phase, we initialize the target model $g_t(\cdot) = h_t(f_t(\cdot))$ with the source model's parameters θ_s.

3.1. Online pseudo label refinement

During the adaptation, we produce pseudo labels $\{\hat{y}^i\}_{i=1}^{n_t}$ for the unlabeled target data using the target model initialized with source weights as a way to re-use knowledge learned from the source domain while gradually bootstrapping to the target domain. Instead of refining and updating pseudo labels only after each epoch [28,55], we propose to predict and refine the pseudo labels at a per-batch basis, so that the model's progressive improvement is reflected in the most recent pseudo labels. The refinement is accomplished via a nearest-neighbor soft voting, which is enabled by a memory queue Q_w representing the target feature space. Specifically, as shown in Fig. 2(a), given a target image x_t and a weak augmentation t_w randomly drawn from a distribution $\mathcal{T}_w$, the weakly-augmented image $t_w(x_t)$ is encoded into a feature vector $w = f_t(t_w(x_t))$, which we use to find its nearest neighbors in the target feature space. The direct prediction for the image is then refined by averaging the probabilities associated with the nearest neighbors, followed by an argmax operation to get the pseudo label $\hat{y}$. Note that this procedure is executed at each mini-batch step.

Memory queue To enable the nearest-neighbor search, we maintain a memory queue Q_w of length M storing features and predicted probabilities $\{w'^j, p'^j\}_{j=1}^{M}$ of the weakly-augmented target samples, and update it on-the-fly with the current mini-batch. The memory queue Q_w is initialized with features and probabilities of M randomly selected target samples. Update is done by enqueue and dequeue similar to [17]. To make the maintained feature space more stable, we use a slowly changing momentum model $g_t'(\cdot) = h_t'(f_t'(\cdot))$ to calculate update features w' and probabilities p':

$$w' = f_t'(t_w(x_t)), \quad p' = \sigma(h_t'(w')) \tag{1}$$

The momentum model g_t''s parameters θ_t' are initialized with the same source weights θ_s at the beginning of the adaptation, and updated with momentum m at each mini-batch step instead of back-propagation:

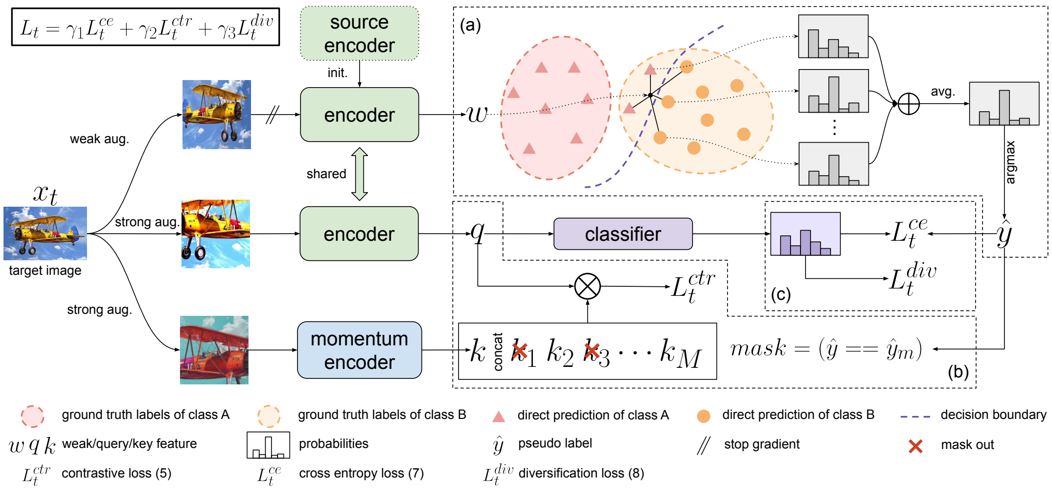
$$\theta_t' \leftarrow m\theta_t' + (1-m)\theta_t \tag{2}$$

Nearest-neighbor soft voting The intuition of soft voting [33] is shown in the feature space of Fig. 2 (a): the current classifier makes incorrect decisions for some target samples due to domain shift; however, by aggregating knowledge of nearby points we can get a more informed estimate, potentially recovering the correct label. The memory queue Q_w effectively represents our estimate of the evolving target feature space. Therefore, we use the feature vector w of the weakly-augmented image $t_w(x_t)$ to retrieve its N nearest neighbors from Q_w based on the cosine distance between w and the entire set of features $\{w'^j\}_{j=1}^{M}$ stored by Q_w. We perform a soft voting among these N neighbors by averaging their probabilities:

$$\hat{p}^{(i,c)} = \frac{1}{N} \sum_{j \in \mathcal{I}_i} p'^{(j,c)} \tag{3}$$

where $\mathcal{I}_i$ is the indices of the N nearest neighbors of w in the memory queue Q_w. After the voting, we get a less noisy estimate of the categorical probability for target sample, upon which we decide the pseudo label:

$$\hat{y}^i = \arg\max_c \hat{p}^{(i,c)} \tag{4}$$



$L_t = \gamma_1 L_t^{ce} + \gamma_2 L_t^{ctr} + \gamma_3 L_t^{div}$
source encoder
init.
encoder
weak aug.
x_t
target image
strong aug.
strong aug.
shared
encoder
momentum encoder
w
(a)
avg.
argmax
q
classifier
L_t^{ce}
$\hat{y}$
L_t^{div}
L_t^{ctr}
(c)
k
concat
k_1 k_2 $k_3 \cdots k_M$
$mask = (\hat{y} == \hat{y}_m)$
(b)
ground truth labels of class A
ground truth labels of class B
direct prediction of class A
direct prediction of class B
decision boundary
$w\ q\ k$ weak/query/key feature
probabilities
$\hat{y}$ pseudo label
stop gradient
mask out
L_t^{ctr} contrastive loss (5)
L_t^{ce} cross entropy loss (7)
L_t^{div} diversification loss (8)

Figure 2. **Framework of our contrastive test-time adaptation approach (AdaContrast):** In the beginning of adapation, the model and momentum model are initialized by source model. A target image is transformed by one weak and two strong augmentations. (a) The weakly-augmented image is encoded into feature vector w that is used to find nearest neighbors based on cosine distance from the target feature space, which is maintained as a memory queue. The associated probabilities are averaged followed by an argmax to get the refined pseudo label $\hat{y}$ for self-training and contrastive learning. (b) Two strongly augmented versions of the image are encoded into query and key features q, k for momentum contrast [6, 17], which is applied jointly with self-training. No projection heads are used; current pseudo label and historical pseudo labels are used to exclude same-class negative pairs. (c) The pseudo label $\hat{y}$ obtained from the weakly-augmented image is also used to supervise predictions for the strongly-augmented image, enforcing the weak-strong consistency in the self-training. Diversity regularization is also posed on the same predictions. Note that the queues used for nearest neighbors search and contrastive learning are separate, which are updated (not illustrated here) with w and k, respectively.

The obtained pseudo labels will be used in the joint optimization of contrastive learning and self-training, shown in Fig. 2 (b) and (c).

3.2. Joint self-supervised contrastive learning

Taking inspirations from existing self-supervised contrastive learning works [4, 7, 15, 17] which exploit pairwise information with contrastive objectives, we apply self-supervised contrastive learning on target data jointly with self-training during test-time adaptation, as illustrated in Fig. 2 (b). In particular, we design our contrastive task following the shared instance-discrimination principle: features of different views of the same image (positive pairs) are pulled closer while features of different images (negative pairs) are pushed away. Different image views are obtained by augmentation: as shown in Fig. 2 on the left, given a target image x_t, we randomly draw two strong augmentations t_s, t'_s from the same distribution $\mathcal{T}_s$ and augment x_t into two versions $t_s(x_t), t'_s(x_t)$. More specifically, we use MoCo [17] as our prototype and introduce several key modifications, which we elaborate next.

Encoder initialization by source Instead of training the image encoder from scratch for a large number of epochs (usually hundreds) as needed by representation learning [4, 15, 17], we reuse the target encoder f_t which is initialized with source model weights. We adopt the momentum encoder f'_t from MoCo as well and initialize it with source weights. We note that this momentum encoder is in fact the same one used for updating memory queue Q_w in Sec. 3.1, here reused for producing contrastive features. By reusing knowledge contained in the source weights θ_s, the contrastive learning starts from an informative feature space, therefore requires very few epochs to converge.

Exclusion of same-class negative pairs The two versions of the target image is encoded into query and key features $q = f_t(t_s(x_t)), k = f'_t(t'_s(x_t))$, respectively. A memory queue Q_s of length P storing features $\{k^j\}_{j=1}^P$ is in turn updated by k. The InfoNCE loss applied in MoCo strives to minimize the cosine distance between q and k while maximizing the cosine distances between q and *every* k^j in Q_s. Instead, we argue that not pushing away same-class pairs helps learn better semantically meaningful clusters. Specifically, we augment the memory queue Q_s by also storing pseudo labels $\{\hat{y}^j\}_{j=1}^P$ associated with past key features, to exclude same-class pairs from all negative pairs:

$$L_t^{ctr} = L_{\text{InfoNCE}} = -\log \frac{\exp q \cdot k_+/\tau}{\sum_{j \in \mathcal{N}_q} q \cdot k_j/\tau} \quad (5)$$

$$\mathcal{N}_q = \{j | 1 \leq j \leq P, j \in \mathbf{Z}, \hat{y} \neq \hat{y}^j\} \cup \{0\} \quad (6)$$

Joint optimization with self-training While existing self-supervised contrastive learning works [4, 15, 17] intend to learn transferrable features in a large-scale pre-training stage which is followed by transferring to specific downstream tasks, AdaContrast jointly optimizes the contrastive objective together with self-training in the test-time adaptation phase. Specifically, the modified InfoNCE term Eq. (5) is combined with the self-training loss in a multi-task fashion (see Eq. (10)). The contrastive learning facilitates self-training with better representation, which in turn benefits from the prior brought by more accurate pseudo labels.

3.3. Additional regularization

Weak-strong consistency Inspired by FixMatch [48], we use the pseudo label $\hat{y}$ obtained from the weakly-augmented target image to "supervise" the model's prediction for the strongly-augmented version as shown in Fig. 2 (c). There are several important distinctions: 1) we do not have access to any ground truth labels, 2) we refine the pseudo labels before using them, 3) we do not apply any confidence thresholding, and 4) our model starts from source initialization. The regularization is reflected in the standard cross entropy loss:

$$L_t^{ce} = -\mathbb{E}_{x_t \in \mathcal{X}_t} \sum_{c=1}^{C} \hat{y}^c \log p_q^c \tag{7}$$

where $p_q = \sigma(g_t(t_s(x_t)))$ are the predicted probabilities for the strongly-augmented query image $t_s(x_t)$.

Diversity regularization While the online pseudo label refinement introduced in Sec. 3.1 effectively reduces noises in pseudo labels brought by domain shift, they are still not ideal as the ground truth labels. To prevent the model from blindly trusting the false labels during the adaptation, we use a regularization term in the loss function to encourage class diversification:

$$L_t^{div} = \mathbb{E}_{x_t \in \mathcal{X}_t} \sum_{c=1}^{C} \bar{p}_q^c \log \bar{p}_q^c \tag{8}$$

$$\bar{p}_q = \mathbb{E}_{x_t \in \mathcal{X}_t} \sigma(g_t(t_s(x_t))) \tag{9}$$

This concludes the overall loss function used for training shown in Eq. (10). We set $\gamma_1 = \gamma_2 = \gamma_3 = 1.0$ without any tuning for all experiments, showing the merit of hyper-parameter insensitivity:

$$L_t = \gamma_1 L_t^{ce} + \gamma_2 L_t^{ctr} + \gamma_3 L_t^{div} \tag{10}$$

4. Experiments

We conduct experiments of closed-set adaptation on major benchmarks. In the following, we first compare the proposed AdaContrast with the previous state-of-the-art algorithms. Then, we discuss several desirable test-time properties of AdaContrast, followed by ablation and analysis of the important design elements that brought the gains.

4.1. Experimental setup

Datasets and Metrics: We use VisDA-C [40] and DomainNet-126 [39] for evaluating our method and comparison. Please see the supplemental material for a detailed description for the datasets. It is worth to know that since the original DomainNet has noisy labels, we follow the authors' followup work [45] to use a subset of it that contains 126 classes from 4 domains (Real, Sketch, Clipart, Painting), which we refer to as DomainNet-126. We follow [45] to evaluate the methods on 7 domain shifts constructed from the 4 domains, and report top-1 accuracy under each domain shift as well as the 7-shift average (denoted Avg.). For VisDA-C we compare the per-class top-1 accuracies, their average (denoted Avg.), and the overall top-1 accuracy (denoted Acc.).

Model Architecture Our method assumes a general method architecture with a feature encoder followed by a classifier. For comparison purpose, we choose ResNet-18/50/101 models [18] as our backbones in different experiments. We follow SHOT [28] to add a 256-dimensional bottleneck consisting of a fully-connected layer followed by a BatchNorm layer [20] after the backbone, and apply WeightNorm [47] on the classifier. Since a lower dimensional bottleneck is applied, we drop the original projection heads from MoCo [6, 17] without seeing performance drop.

Baselines We compare our method with both classical unsupervised domain adaptation (UDA) baselines and source-free/test-time adaptation baselines. For UDA methods we compare to DANN [12], CDAN [32], CDAN+BSP [8], CAN [22], SWD [25], and MCC [21]. It is worth noting that all UDA methods have access to source data during adaptation. For TTA methods we compare to MA [27], BAIT [57], TENT [56], SHOT [28], On-target [55] as representative methods based on image generation, class prototypes, entropy minimization, pseudo labeling, and the combination of contrastive feature and pseudo labeling. For MCC[1], SHOT, and TENT we run the code released by the authors; for other baselines we cite their numbers.

Implementation We use Pytorch [38] for all implementation. **For source training** we initialize the ResNet backbone with ImageNet-1K [11] pre-trained weights in the Pytorch model zoo. We follow [28] to randomly split the source data into 9:1 ratio where 90% is used to train the source model and 10% is used for validation. Source training has 10, 60 epochs for VisDA-C and DomainNet-126 respectively. **For target training** we use only 15 epochs for all datasets unless otherwise noted. **For all experiments**,

[1] for DomainNet; VisDA-C numbers are cited

Table 1. Classification accuracy (%) on VisDA-C train → val. All methods use ResNet-101 backbone except the on-target rows, which use ResNet-18 as student network. Bold is the highest; underline is the second highest. The proposed AdaContrast surpasses the previous state-of-the-art by 3.8% Avg. When applying an extra knowledge distillation stage following [55], we achieve the highest 87.2% with a small ResNet-18 backbone. AdaContrast also achieves competitive performance of 78.7% Avg. when used in online adaptation setting.

Method	source-free	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
DANN [12]	no	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
CDAN [32]	no	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CDAN+BSP [8]	no	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	77.9	82.1	38.4	75.9
CAN [22]	no	97.0	87.2	82.5	74.3	**97.8**	**96.2**	90.8	80.7	**96.6**	**96.3**	87.5	**59.9**	**87.2**
SWD [25]	no	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
MCC [21]	no	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
Source only	-	57.2	11.1	42.4	66.9	55.0	4.4	81.1	27.3	57.9	29.4	86.7	5.8	43.8
MA [27]	yes	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
BAIT [57]	yes	93.7	83.2	84.5	65.0	92.9	95.4	88.1	80.8	90.0	89.0	84.0	45.3	82.7
SHOT [28]	yes	95.3	87.5	78.7	55.6	94.1	94.2	81.4	80.0	91.8	90.7	86.5	59.8	83.0
+ On-target [55]	yes	96.0	**89.5**	84.3	67.2	95.9	94.2	91.0	81.5	93.8	89.9	89.1	58.2	85.9
AdaContrast (Ours)	yes	97.0	84.7	84.0	77.3	96.7	93.8	91.9	84.8	94.3	93.1	**94.1**	49.7	86.8
+ On-target [55]	yes	**97.2**	87.0	**86.7**	**81.7**	95.5	91.6	**93.5**	**86.6**	95.3	90.9	92.8	47.9	**87.2**
AdaContrast (Ours, online)	yes	95.0	68.0	82.7	69.6	94.3	80.8	90.3	79.6	90.6	69.7	87.6	36.0	78.7

Table 2. Classification accuracy (%) on 7 domain shifts of DomainNet-126. All methods use ResNet-50 backbone. Bold is the highest. The proposed AdaContrast achieves the highest average performance, and on 4 domain shifts. Its performance under online test-time adaptation setting also reaches a competitive number at 62.6%.

Method	Source-free	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Avg.
MCC [21]	no	44.8	65.7	41.9	34.9	47.3	35.3	72.4	48.9
Source only	-	55.5	62.7	53.0	46.9	50.1	46.3	75.0	55.6
TENT [56]	yes	58.5	65.7	57.9	48.5	52.4	54.0	67.0	57.7
SHOT [28]	yes	67.7	68.4	66.9	**60.1**	**66.1**	59.9	**80.8**	67.1
AdaContrast (Ours)	yes	**70.2**	**69.8**	**68.6**	58.0	65.9	**61.5**	80.5	**67.8**
AdaContrast (Ours, online)	yes	61.1	66.9	60.8	53.4	62.7	54.5	78.9	62.6

we use SGD optimizer with momentum 0.9 and weight decay 1e-4, and cosine annealing on the learning rate, decaying from initial value to zero based on the training progress: $\eta = \eta_0 \cdot 0.5(\cos(a \cdot \pi/2) + 1)$. The newly added bottleneck and classifier layers have learning rate 10 times of the backbone. The initial learning rate for the backbone is set to 2e-4. Batch size is set to 128.

4.2. Results

VisDA-C train → val Tab. 1 compares AdaContrast with state-of-the-art unsupervised domain adaptation and test-time adaptation methods on VisDA-C's "train" to "val" shift. For UDA, our method is on-par with a strong UDA baseline CAN [22] and significantly outperforms the others by a large margin, even though we do not utilize source data at all during test-time adaptation. In the more challenging TTA setting, we achieve the highest per-class average accuracy by a notable margin (+3.8%) upon SHOT. Compared to on-target adaption which is also built with SHOT, we gain an extra 0.9% improvement, demonstrating the power of joint training and online refinement. In addition, when applying an extra knowledge distillation phase following [55], we are able to reach 87.2% per-class average with a contrastive (MoCo v2 [6]) pre-trained ResNet-18 backbone.

DomainNet-126 seven domain shifts Tab. 2 shows the comparison between AdaContrast and state-of-the-art UDA (first section) and TTA (second sections) methods. Without needing source data during the adaptation AdaContrast outperforms the UDA method MCC [21] by +18.9% on the averaged performance. When being compared to TTA methods AdaContrast outperforms TENT by +10.1% on the averaged performance. It achieves the best performance on 4 out of 7 domain shifts as well as the highest averaged performance.

4.3. Analysis and Discussion

AdaContrast has better model calibration than entropy minimization-based methods. Entropy minimization-based methods [28, 56] achieved competitive results by explicitly making the model "certain" on target predictions. However, one setback of this is that the model calibration is disrupted due to direct entropy optimization regardless of true labels. We argue that a good model calibration is an important property of TTA algorithm in practice because it provides a measure to help gauge how much we should trust the adapted model. In Fig. 3, we show the comparison of model calibration for both SHOT and AdaContrast on VisDA-C validation split. We follow the practice of network

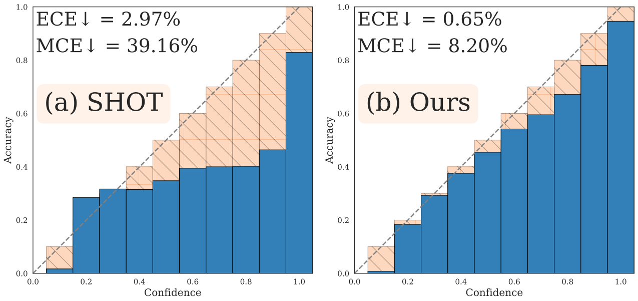


Figure 3. Model calibration comparison of SHOT vs. proposed AdaContrast on VisDA-C validation split after adaptation. The proposed AdaContrast has much better model calibration metrics. For ECE (expected calibration error) and MCE (maximum calibration error): lower is better; the more the bars are aligned with $y = x$ the better.

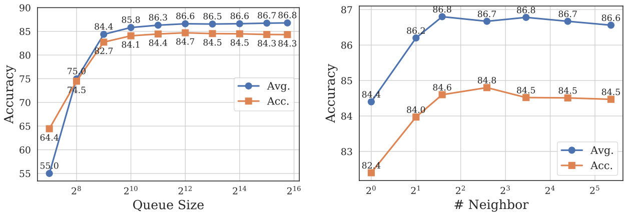


Figure 4. Ablation on memory queue size M and number of neighbors N used in soft voting, on VisDA-C. AdaContrast is able to achieve state-of-the-art performance consistently over a wide range of choices. Notably, M can be as small as less than 4% of the full dataset and still maintains on-par performance.

calibration [16] and illustrate reliability diagrams [10, 35]. For each adapted target model, we divide the probability range $[0, 1]$ into 10 bins and calculate the model's average accuracy versus average confidence on target data for each bin. The more close the model's bars are to the diagonal line $y = x$, the better calibration it has. As shown in Fig. 3, the bars for SHOT significantly falls below $y = x$, meaning its predictions on target data are over-confident, whereas the curve for AdaContrast aligns much better with $y = x$. In addition, we calculate two scalar summary statistic of calibration [34]: expected calibration error (ECE) and maximum calibration error (MCE), where the perfectly calibrated classifier should have both scores as low as zero. Given lower ECE and MCE indicate better calibration, AdaContrast has only 0.65% ECE and 8.20% MCE, which decrease by a factor of 4.5+ compared to the over-confident SHOT with 2.97% ECE and 39.16% MCE, demonstrating the effectiveness of steering away from entropy minimization.

AdaContrast is insensitive to hyper-parameters choices. Hyper-parameter sensitivity is often neglected in TTA literature [23, 28, 56], which we believe is an important aspect of TTA algorithms. In Fig. 4, we show that under a wide range of hyper-parameters choices that are specific to AdaContrast, the performance is consistently state-of-the-art on VisDA-C. Specifically, we show performance with queue size $M \in \{128, 256, \cdots, 32768, 55388\}$ and number of neighbors in soft voting $N \in \{1, 2, 3, 6, 11, 21, 41\}$. While we report the strongest results of AdaContrast in Tab. 1 with memory size $M = 55388$, queue update, and $N = 11$ nearest neighbors for soft voting, as shown in the plots, we see negligible performance degradation when using a much smaller M, or varying N, achieving around 84.5% overall accuracy and around 86.7% per-class averaged accuracy consistently. By using as few as $M = 512$ queue size, we are able to reach state-of-the-art TTA performance at 84.4% per-class average accuray, and on-par (86.3%) with the full version ($M = 55388$) performance when using $M = 2048$, less than 4% of the full size. In Tab. 3, we show AdaContrast is insensitive to learning rate choices as well. With 1x, 3x, 10x the learning rate used for reporting the main results in Sec. 4.2, AdaContrast consistently achieves state-of-the-art performance on VisDA-C both VisDA-C and DomainNet-126, whereas performance of SHOT [28] drops noticeably on DomainNet-126 and significantly on VisDA-C.

AdaContrast has strong performance in online test-time adaptation setting. Since AdaContrast does not rely on global memory banks or processing the entire dataset before the adaptation [28], it is naturally suited for online adaptation where target images arrive in a flow of mini-batches and each image is seen only once. Under this setting, we do not decay the learning rate and turn off the pseudo label refinement (note that pseudo labels are still acquired on the fly for each mini-batch, only that the direct predictions are used instead) for the first X samples, and turn it on once the memory queue Q_w has accumulated X feature-probability pairs. We emprically show that with $X = 2048$ for VisDA-C (less than 4% of the entire dataset) and $X = 1024$ (less than 4% on the entire datasets on average), we are able to achieve state-of-the-art online adaptation performance by large margins. AdaContrast achieves 62.6% accuracy averaged on 7 domain shifts in **DomainNet-126**, surpassing the UDA method MCC [21] (see Tab. 2) by +13.7%. On **VisDA-C**, AdaContrast's impressive 78.7% overall accuracy slightly surpasses the performance of the offline SHOT [28] by +0.5%, the 78.7% per-class average accuracy surpassing 4 UDA methods listed in Tab. 1.

4.4. Ablation studies

In Tab. 4, we start with applying the simplest form of pseudo labeling (referred as #1), which makes inference on the entire target dataset at the beginning of each epoch and takes all predictions as pseudo labels for the epoch. This achieves 58.5% average accuracy (Avg.) on DomainNet-126, 55.0% per-class averaged accuracy (Avg.) on VisDA-

Table 3. Comparison of classification accuracy (%) on DomainNet-126 and VisDA-C between AdaContrast and SHOT under 1x, 3x, and 10x learning rate scaling. AdaContrast is less sensitive to the choice of learning rate, achieving consistently high performance on both datasets.

Method	lr scale	DN-126	VisDA-C	
		Avg.	Acc.	Avg.
SHOT [28]	1×	67.1	78.3	83.0
	3×	66.4	77.6	82.2
	10×	64.7	66.8	72.1
AdaContrast (Ours)	1×	67.8	84.5	86.8
	3×	67.8	84.7	86.8
	10×	67.5	85.0	86.6

Table 4. Ablation study of algorithmic components of proposed AdaContrast measured by classification accuracy (%) on DomainNet-126 under 1x learning rate scaling and VisDA-C under both 1x, 10x learning rate scaling. #0 for source-only baseline to start with. Online pseudo label refinement Sec. 3.1, joint contrastive learning Sec. 3.2 and regularzation techniques Sec. 3.3 are able to bring significant performance gain as well as hyperparameter insensitivity.

#	Pseudo labeling	Online pl. ref	Joint ctr.	Reg.	DN-126 (lr1x)	VisDA-C (lr1x)	VisDA-C (lr10x)
0					55.6	43.8	43.8
1	✓				58.5	55.0	44.5
2	✓	✓			64.7	86.5	9.9
3	✓	✓	✓		67.9	85.7	84.3
4	✓	✓	✓	✓	67.8	86.8	86.6

C with 1x learning rate, but merely 44.5% Avg. with 10x learning rate. We note that for VisDA-C we include expriments with 10x learning rate of that used in reporting the main results in Tab. 1, to emphasize the effect of each component of AdaContrast under an unfortunate choice of learning rate, diving deeper into observations from Tab. 3.

Online pseudo label refinement In row #2 we change the pseudo labeling scheme to the one introduced in 3.1. Due to having more accurate pseudo labels, the performance on DomainNet-126 increases by +6.2% to 64.7% and significantly by +31.5% to 86.5% on VisDA-C with 1x learning rate. However, switching to the online refinement scheme is not trivial, since it is prone to diverge due to bad hyper-parameter selection and compounding errors. As shown in the VisDA-C 10x learning rate performance where the accuracies drop significantly down to near random (12 classes). However, the cross-entropy loss did not diverge from our observation, which means the model severely overfitted to the highly-noisy pseudo labels which we are unable to know.

Joint self-supervised contrastive learning In row #3 we show results obtained by enabling the joint contrastive learning introduced in 3.2, which is simply done by setting $\gamma_2 = 1.0$ for L_t^{ctr} in Eq. (10). This brings another significant performance gain on DomainNet-126, from 64.7% average accuracy to 67.9%. Notably on VisDA-C with 10x learning rate, the joint contrastive learning is able to recover the diverged accuracy from 10.0% in row #2 to 84.3% Avg. This demonstrates the huge potential of our joint contrastive learning in stablizing the feature space, therefore ensuring the model is less susceptible to the compounding errors in pseudo labels as well as hyper-parameter choices. It is worth noting that the improvements include gains from using pseudo labels to exclude same-class negatives (Sec. 3.2), on top of 67.7% (+0.2%), 83.6% (+2.1%), and 81.5 (+2.8%) for the three entries of DomainNet-126 and VisDA-C without excluding same-class negatives. This validates the effectiveness of using semantic priors in pseudo labels to benefit contrastive learning.

Diversity and weak-strong regularization In row #4 and we show the effect of two additional regularization in Sec. 3.3: the weak-strong consistency and diversity term L_t^{div}. On DomainNet-126 they keep the model's high performance around 67.8% consistently, whereas on VisDA-C with 10x learning rate they bring further improvements: we get +0.9% gains on per-class average accuracy.

5. Limitations

Domain adaptation methods are foundational, and as such have as much potential for misuse as they have for beneficial application. Adaptation methods have the potential to increase the robustness of models deployed to new domains, which could amplify the benefits and harms of larger AI applications. Our method improves model calibration, which in general provides for more reliable systems; however this could lead to inappropriate trust in deployed systems.

6. Conclusion

We introduced AdaContrast, a novel test-time adaptation approach for closed-set DA in image classification. AdaContrast starts from a pretrained model on the source domain and uses contrastive learning along with pseudo labeling on the target domain. We proposed an online refinement scheme that generates pseudo labels in a per-batch basis and refines the predictions using nearest neighbor soft voting technique which results in significantly more accurate pseudo labels. We showed AdaContrast not only surpassed the existing TTA approaches on major DA benchmarks but also has several empirical merits: hyper-parameter insensitivity, better model calibration, and no need for global memory banks, which we believe are all desirable properties of successful TTA algorithms.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 3

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017. 1

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 3

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 4, 5

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 2

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 4, 5, 6

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2, 4

[8] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pages 1081–1090. PMLR, 2019. 5, 6

[9] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2

[10] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983. 7

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 2, 5, 6

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2, 4, 5

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. 2, 7

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2, 3, 4, 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[19] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 1, 2

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5

[21] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020. 5, 6, 7

[22] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 5, 6

[23] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020. 1, 2, 7

[24] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017. 2

[25] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019. 5, 6

[26] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 3

[27] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation

without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 1, 2, 5, 6

[28] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 1, 2, 3, 5, 6, 7, 8

[29] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021. 3

[30] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[31] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 1, 2

[32] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *arXiv preprint arXiv:1705.10667*, 2017. 5, 6

[33] HB Mitchell and PA Schaefer. A “soft” k-nearest neighbor voting scheme. *International journal of intelligent systems*, 16(4):459–468, 2001. 2, 3

[34] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. 7

[35] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005. 7

[36] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 2, 5

[40] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 5

[41] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 2

[42] Joaquin Quiñonero-Candela, Masashi Sugiyama, Neil D Lawrence, and Anton Schwaighofer. *Dataset shift in machine learning*. MIT Press, 2009. 1

[43] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and N Lawrence. Covariate shift and local learning by distribution matching, 2008. 1

[44] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 1

[45] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. *ICCV*, 2019. 5

[46] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*, 2020. 2

[47] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29:901–909, 2016. 5

[48] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 3, 5

[49] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019. 2

[50] Yu Sun, Xiaolong Wang, Liu Zhuang, John Miller, Moritz Hardt, and Alexei A. Efros. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020. 1, 2

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[52] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 2

[53] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 1

[54] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[55] Dequan Wang, Shaoteng Liu, Sayna Ebrahimi, Evan Shelhamer, and Trevor Darrell. On-target adaptation. *arXiv preprint arXiv:2109.01087*, 2021. 2, 3, 5, 6

[56] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1, 2, 5, 6, 7

[57] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020. 1, 2, 5, 6

[58] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021. 1

[59] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 2