CDTRANS: CROSS-DOMAIN TRANSFORMER FOR UNSUPERVISED DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised domain adaptation (UDA) aims to transfer knowledge learned from a labeled source domain to a different unlabeled target domain. Most existing UDA methods focus on learning domain-invariant feature representation, either from the domain level or category level, using convolution neural networks (CNNs)-based frameworks. One fundamental problem for the category level based UDA is the production of pseudo labels for samples in target domain, which are usually too noisy for accurate domain alignment, inevitably compromising the UDA performance. With the success of Transformer in various tasks, we find that the cross-attention in Transformer is robust to the noisy input pairs for better feature alignment, thus in this paper Transformer is adopted for the challenging UDA task. Specifically, to generate accurate input pairs, we design a two-way center-aware labeling algorithm to produce pseudo labels for target samples. Along with the pseudo labels, a weight-sharing triple-branch transformer framework is proposed to apply self-attention and cross-attention for source/target feature learning and source-target domain alignment, respectively. Such design explicitly enforces the framework to learn discriminative domain-specific and domain-invariant representations simultaneously. The proposed method is dubbed CDTrans (cross-domain transformer), and it provides one of the first attempts to solve UDA tasks with a pure transformer solution. Experiments show that our proposed method achieves the best performance on public UDA datasets, e.g. VisDA-2017 and DomainNet.

1 INTRODUCTION

Deep neural network have achieved remarkable success in a wide range of application scenarios but it still suffers poor generalization performance to other new domain because of the domain shift problem (Csurka, 2017; Zhao et al., 2020; Zhang et al., 2020; Oza et al., 2021). To handle this issue and avoid the expensive laborious annotations, lots of research efforts (Bousmalis et al., 2017; Kuroki et al., 2019; Wilson & Cook, 2020; VS et al., 2021) are devoted on Unsupervised Domain Adaptation (UDA). The UDA task aims to transfer knowledge learned from a labeled source domain to a different unlabeled target domain. In UDA, most approaches focus on aligning distributions of source and target domain and learning domain-invariant feature representations. One kind of such UDA methods are based on category-level alignment (Kang et al., 2019; Zhang et al., 2019; Jiang et al., 2020; Li et al., 2021b), which have achieved promising results on public UDA datasets using deep convolution neural networks (CNNs). The fundamental problems in category-level based alignment is the production of pseudo labels for samples in target domain to generate the input source-target pairs. However, the current CNNs-based methods are not robust to the generated noisy pseudo labels for accurate domain alignment (Morerio et al., 2020; Jiang et al., 2020).

With the success of Transformer in natural language processing (NLP) (Vaswani et al., 2017; Devlin et al., 2018) and vision tasks (Dosovitskiy et al., 2020; Han et al., 2020; He et al., 2021; Khan et al., 2021), it is found that cross-attention in Transformer is good at aligning different distributions, even from different modalities *e.g.*, vision-to-vision (Li et al., 2021e), vision-to-text (Tsai et al., 2019; Hu & Singh, 2021) and text-to-speech (Li et al., 2019). And we find that it is robust to noise in pseudo labels to some extent. Hence, in this paper, we apply transformers to the UDA task to take advantage of its robustness to noise and super power for feature alignment to deal with the problems as described above in CNNs.

In our experiment, we conclude that even with noise in the labeling pair, the cross-attention can still work well in aligning two distributions, thanks to the attention mechanism. To obtain more accurate pseudo labels, we designed a two-way center-aware labeling algorithm for samples in the target domain. The pseudo labels are produced based on the cross-domain similarity matrix, and a center-aware matching is involved to weight the matrix and weaken noise into the tolerable range. With the help of pseudo labels, we design the cross-domain transformer (CDTrans) for UDA. It consists of three weight-sharing transformer branches, of which two branches are for source and target data respectively and the third one is the feature alignment branch, whose inputs are from source-target pairs. The self-attention is applied in the source/target transformer branches and cross-attention is involved in the feature alignment branch to conduct domain alignment. Such design explicitly enforces the framework to learn discriminative domain-specific and domain-invariant representations simultaneously. In summary, our contributions are three-fold:

- We propose a weight-sharing triple-branch transformer framework, namely, CDTrans, for accurate unsupervised domain adaptation, taking advantage of its robustness to noisy labeling data and great power for feature alignment.
- To produce pseudo labels with high quality, a two-way center-aware labeling method is proposed, and it boosts the final performance in the context of CDTrans.
- CDTrans achieves the best performance compared to state-of-the-arts with a large margin on VisDA-2017 (Peng et al., 2017) and DomainNet (Peng et al., 2019) datasets.

2 Related Work

2.1 Transformer for Vision

Transformer is proposed in (Vaswani et al., 2017) to model sequential data in the field of NLP. Many works have shown its effectiveness for computer-vision tasks (Han et al., 2020; Khan et al., 2021; Li et al., 2021d; Han et al., 2021b; Yu et al., 2021; Li et al., 2021c; Yang et al., 2021). Pure Transformer based models are becoming more and more popular. For example, ViT (Dosovitskiy et al., 2020) is proposed recently by feeding transformer with sequences of image patches; Touvron et al. (Touvron et al., 2021) propose DeiT that introduces a distillation strategy for transformers to help with ViT training; many other ViT variants (Yuan et al., 2021a; Wang et al., 2021; Han et al., 2021a; Chen et al., 2021; Ranftl et al., 2021; Liu et al., 2021) are proposed from then, which achieve promising performance compared with its counterpart CNNs for both image classification and downstream tasks, such as object detection (Liu et al., 2021), semantic segmentation (Yuan et al., 2021b) and object ReID (He et al., 2021). For multi-modal based networks, there are several works (Tsai et al., 2019; Li et al., 2021e; Hu & Singh, 2021) that apply cross-attention for multi-modal feature fusion, which demonstrates that attention mechanism is powerful at distilling noise and feature alignment. This paper adopts cross-attention in the context of pure transformers for UDA tasks.

2.2 Unsupervised Domain Adaptation

There are mainly two levels for UDA methods: domain-level (Tzeng et al., 2014; Long et al., 2015; Ghifary et al., 2016; Tzeng et al., 2017; Bousmalis et al., 2017; Hoffman et al., 2018) and category-level (Saito et al., 2018; Kang et al., 2019; Du et al., 2021; Li et al., 2021a). Domain-level UDA mitigates the distribution divergence between the source and target domain by pulling them into the same distribution at different scale levels. The commonly used divergence measures include Maximum Mean Discrepancy (MMD) (Gretton et al., 2006; Tzeng et al., 2014; Long et al., 2015) and Correlation Alignment (CORAL) (Sun et al., 2016; Sun & Saenko, 2016). Recently, some works (Saito et al., 2018; Du et al., 2021; Li et al., 2021a) focus on the fine-grained category-level label distribution alignment through an adversarial manner between the feature extractor and two domain-specific classifiers. Unlike coarse-grained alignment at the domain scale, this approach aligns each category distribution between the source and target domain data by pushing the target samples to the distribution of source samples in each category. Obviously, the fine-grained alignment results in more accurate distribution alignment within the same label space. Although the adversarial approach achieves new improvements by fusing fine-grained alignment operations of source and target samples at the category level, it still does not solve the problem of noisy samples in the wrong category. Our method adopts Transformers for category-level UDA to solve the noise problem.

2.3 PSEUDO LABELING

Pseudo labeling (Lee et al., 2013) is first introduced for semi-supervised learning and gains popularity in domain adaptation tasks. It learns to label unlabeled data using predicted probabilities and performs fine-tuning together with labeled data. In terms of using pseudo labeling for domain adaptation tasks, (Long et al., 2017; 2018) adopt pseudo labels to conduct conditional distribution alignment; (Zhang et al., 2018; Choi et al., 2019) use pseudo labels as a regularization for domain adaptation; Zou et al. (2018) designs a self-training framework by alternately solving pseudo labels; Caron et al. (2018) propose a deep self-supervised method by generating pseudo labels via k-means cluster to progressively train the model; Liang et al. (2020) develop a self-supervised pseudo labeling method to alleviate the effects of noisy pseudo labels. Based on Liang et al. (2020), in this work, we propose a two-way center-aware labeling algorithm to further filter the noisy pseudo pairs.

3 THE PROPOSED METHOD

We first introduce the cross attention module and analyze its robustness to the noise in Section 3.1. Then the two-way center-aware labeling method is presented in Section 3.2. With the produced pseudo labels as inputs, our cross-domain transformer (CDTrans) is proposed in Section 3.3, consisting of three weight-sharing transformers.

3.1 THE CROSS ATTENTION IN TRANSFORMER

3.1.1 PRELIMINARY

Vision Transformer (ViT) (Dosovitskiy et al., 2020) has achieved comparable or even superior performance on computer vision tasks. One of the most important structures in ViT is the self-attention module (Vaswani et al., 2017). In ViT, an image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times C}$ is reshaped into a sequence of flattened 2D patches $x \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches. For self-attention, the patches are first projected into three vectors, *i.e.* queries $\boldsymbol{Q} \in \mathbb{R}^{N \times d_k}$, keys $\boldsymbol{K} \in \mathbb{R}^{N \times d_k}$ and values $\boldsymbol{V} \in \mathbb{R}^{N \times d_v}$. d_k and d_v indicates their dimensions. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The N patches serve as the inputs for the self-attention module, and the process can be formulated as below. The self-attention module aims to emphasize relationships among patches of the input image.

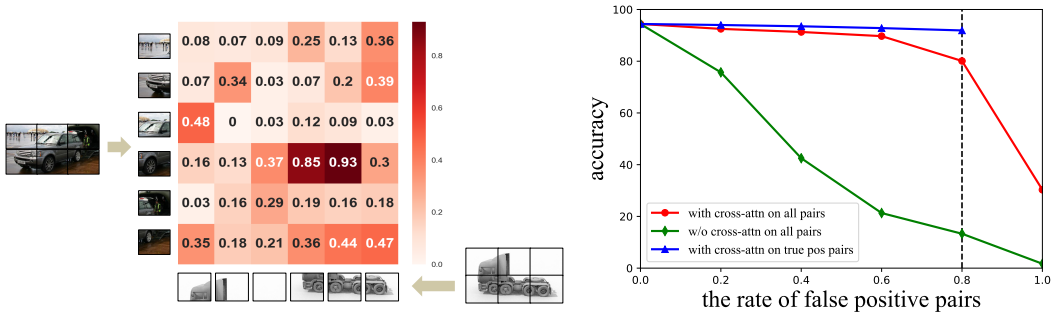
$$Attn_{self}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \tag{1}$$

The cross-attention module is derived from the self-attention module. The difference is that the input of cross-attention is a pair of images, *i.e.* $\boldsymbol{I}_s$ and $\boldsymbol{I}_t$. Its query and key/value are from patches of $\boldsymbol{I}_s$ and $\boldsymbol{I}_t$ respectively. The cross-attention module can be calculated as follows:

$$Attn_{cross}(\boldsymbol{Q}_s, \boldsymbol{K}_t, \boldsymbol{V}_t) = softmax(\frac{\boldsymbol{Q}_s\boldsymbol{K}_t^T}{\sqrt{d_k}})\boldsymbol{V}_t \tag{2}$$

where $\boldsymbol{Q}_s \in \mathbb{R}^{M \times d_k}$ are queries from M patches of image $\boldsymbol{I}_s$, and $\boldsymbol{K}_t \in \mathbb{R}^{N \times d_k}, \boldsymbol{V}_t \in \mathbb{R}^{N \times d_v}$ are keys and values from N patches of image $\boldsymbol{I}_t$. The output of the cross-attention module holds the same length M as the number of the queries. For each output, it is calculated by multiplying $\boldsymbol{V}_t$ with attention weights, which comes from the similarity between the corresponding query in $\boldsymbol{I}_s$ and all the keys in $\boldsymbol{I}_t$. As a result, among all patches in $\boldsymbol{I}_t$, the patch that is more similar to the query of $\boldsymbol{I}_s$ would hold a larger weight and contribute more to the output. In other words, the output of the cross-attention module manages to aggregate the two input images based on their similar patches.

So far, many researchers have utilized the cross-attention for feature fusion, especially in multi-modal tasks (Tsai et al., 2019; Li et al., 2019; Hu & Singh, 2021; Li et al., 2021e). In these works, the inputs of the cross-attention module are from two modalities, *e.g.* vision-to-text (Tsai et al., 2019; Hu & Singh, 2021), text-to-speech (Li et al., 2019) and vision-to-vision (Li et al., 2021e). They apply the cross-attention to aggregate and align the information from two modalities. Given



(a) (b)

Figure 1: (a): The heatmap of the cross-attention weights for a false positive pair (Car vs. Truck). (b): The changes of UDA performance by the ratio of false positive pairs. The red/green curves represent the model with and without the cross-attention module. The blue curve means that only true positive pairs are involved in the cross-attention module.

its great power in feature alignment, we propose to use the cross attention module to solve the unsupervised domain adaptation problem.

3.1.2 Robustness to Noise

As mentioned above, the input of the cross-attention module is a pair of images, which usually comes from two domains, and the cross-attention module aims to align these two images. If label noise exists, there would be false positive pairs in the training data. Images in the false positive pairs would have dissimilar appearance, and forcibly aligning their features would inevitably injure the training and compromise the performance. We assume that the dissimilar patches in false positive pairs are more harmful to the performance than the similar patches. In the cross-attention module, two images are aligned based on their patch similarity. As shown in Fig. 1a, the cross-attention module would assign a low weight to the dissimilar patches in false positive pairs. Thus it weakens the negative effects of the dissimilar patches on the final performance to some extent.[1]

To further analyze this issue, an experiment is carefully designed. Specifically, we randomly sample true positive pairs from source and target domain in VisDA-2017 dataset (Peng et al., 2017) as the training data. Then we manually replace the true positive pairs with random false positive pairs to increase the noise, and watch the changes of the performance as shown in Fig. 1b. The x-axis indicates the rate of false positive pairs in the training data, and the y-axis shows the performance of different methods on the UDA task. The red curve represents the results by aligning pairs with the cross-attention module, while the green curve is that without cross-attention, *i.e.* to directly train the target data with the label of corresponding source data in the pair. It can be seen that the red curve achieves a much better performance than the green one, which implies the robustness of the cross-attention module to the noise.

We also provide another baseline shown as the blue curve in Fig. 1b, which is to remove the false positive pairs from the training data and train the cross-attention with only true positive pairs. Without the noisy data, this baseline can be considered as the upper bound to our methods. We can see the red curve is very close to the blue curve, and both of them are much better than the green one. It further implies that the cross-attention module is robust to the noisy input pair.

3.2 Two-Way Center-Aware Pseudo Labeling

3.2.1 Two-Way Labeling

To build the training pairs for the cross-attention module, an intuitive method is that for each image in the source domain, we manage to find the most similar image from the target domain. The set $\mathbb{P}_S$

[1]The patches in true positive pairs, either similar and dissimilar, would bring no noise to the final performance, which is out of the discussion of our paper.

of selected pairs is:

$$\mathbb{P}_S = \{(s,t)|t = \min_k d(\boldsymbol{f}_s, \boldsymbol{f}_k), \forall k \in T, \forall s \in S\} \tag{3}$$

where S, T are the source and target data respectively. $d(\boldsymbol{f}_i, \boldsymbol{f}_j)$ means the distance between features of image i and j. The advantage of this strategy is to make full use of source data, while its weakness is obvious that only a part of target data is involved. To eliminate this training bias from target data, we introduce more pairs $\mathbb{P}_T$ from the opposite way, consisting of all the target data and their corresponding most similar images in the source domain.

$$\mathbb{P}_T = \{(s,t)|s = \min_k d(\boldsymbol{f}_t, \boldsymbol{f}_k), \forall t \in T, \forall k \in S\} \tag{4}$$

As a result the final set $\mathbb{P}$ is the union of two sets, *i.e.* $\mathbb{P} = \{\mathbb{P}_S \cup \mathbb{P}_T\}$, making the training pairs include all the source and target data.

3.2.2 CENTER-AWARE FILTERING

The pairs in $\mathbb{P}$ are built based on the feature similarities of images from both domains, thus the accuracy of the pseudo labels of pairs is highly dependent on the feature similarities. Inspired by Liang et al. (2020), we find that the pre-trained model of the source data is also useful to further improve the accuracy. Firstly, we send all the target data through the pre-trained model and obtain their probability distributions δ on the source categories from the classifier. Similar to Liang et al. (2020), these distributions can be used to compute initial centers of each category in the target domain by weighted k-means clustering:

$$\boldsymbol{c}_k = \frac{\sum_{t \in T} \delta_t^k \boldsymbol{f}_t}{\sum_{t \in T} \delta_t^k} \tag{5}$$

where δ_t^k indicates the probability of image t on category k. Pseudo labels of the target data can be produced via the nearest neighbor classifier:

$$y_t = \arg\min_k d(\boldsymbol{c}_k, \boldsymbol{f}_t) \tag{6}$$

where $t \in T$ and $d(i,j)$ is the distance of features i and j. Based on the pseudo labels, we can calculate new centers:

$$\boldsymbol{c}'_k = \frac{\sum_{t \in T} \mathbb{1}(y_t = k) \boldsymbol{f}_t}{\sum_{t \in T} \mathbb{1}(y_t = k)} \tag{7}$$

In Liang et al. (2020), Eq. 6 and 7 could be updated for multiple rounds, and we only adopt one round in our paper. The final pseudo labels are then used to refine the selected pairs. Specifically, for every pair, if the pseudo label of the target image is consistent with the label of the source image, this pair would be kept for our training, otherwise it will be discarded as a noise.

3.3 CDTRANS: CROSS-DOMAIN TRANSFORMER

The framework of the proposed Cross-domain Transformer (CDTrans) is shown in Fig. 2, which consists of three weight-sharing transformers. There are three data flows and constraints for the weight-sharing branches.

The inputs of the framework are the selected pairs from our labeling method mentioned above. The three branches are named as *source branch*, *target branch*, *source-target branch*. As shown in Fig. 2, the source and target images in the input pair are sent to source branch and target branch respectively. In these two branches, the self-attention module is involved to learn the domain-specific representations. And the softmax cross-entropy loss is used to train the classification. It is worth noting that all three branches share the same classifier due to the same label of two images.

The cross-attention module is imported in the source-target branch. The inputs of the source-target branch are from the other two branches. In the N-th layer, the query of the cross-attention module comes from the query in the N-th layer of the source branch, while the keys and values are from

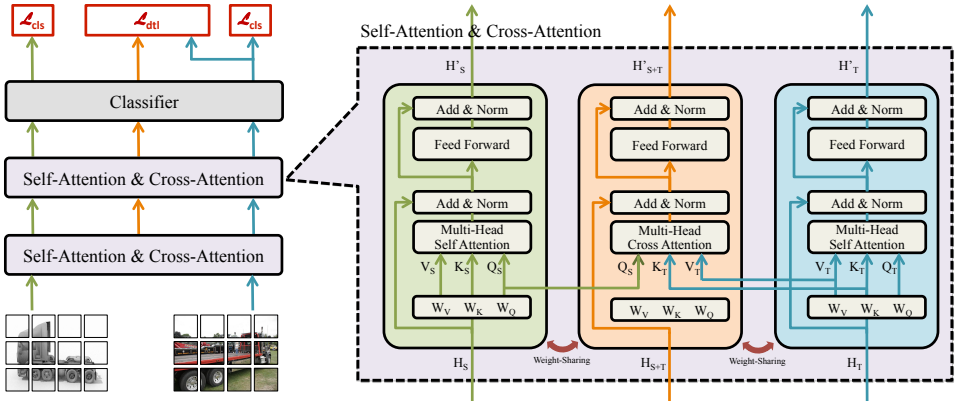


Figure 2: The proposed CDTrans framework. It consists of three weight-sharing transformers fed by inputs from the selected pairs using the two-way center-aware labeling method. Cross-entropy is adopted to *source branch* (H_S) and *target branch* (H_T), while the distillation loss is applied between *source-target branch* (H_{S+T}) and H_T.

those of the target branch. Then the cross-attention module outputs aligned features which are added with the output of the $(N-1)$-th layer.[2]

The features of the source-target branch not only align distributions of two domains, but are robust to the noise in the input pairs thanks to the cross-attention module. Thus we use the output of the source-target branch to guide the training of the target branch. Specifically, the source-target branch and target branch are denoted as teacher and student respectively. We consider the probability distribution of the classifier in source-target branch as a soft label that can be used to further supervise the target branch through a distillation loss (Hinton et al., 2015):

$$L_{dtl} = \sum_k q_k \log p_k \tag{8}$$

where q_k and p_k are the probabilities of category k from the source-target branch and the target branch respectively.

During inference, only the target branch is used. The input is an image from testing data, and only the target data flow is triggered, *i.e.* the blue lines in Fig. 2. Its output of the classifier is utilized as the final predicted labels.

4 Experiments

4.1 Datasets and Implementation

The proposed method is verified on four popular UDA benchmarks, including VisDA-2017 (Peng et al., 2017), Office-Home (Venkateswara et al., 2017), Office-31 (Saenko et al., 2010) and DomainNet (Peng et al., 2019). The input image size in our experiments is 224×224. Both the DeiT-small and DeiT-base (Touvron et al., 2021) are adopted as our backbone for fair comparison. We use the Stochastic Gradient Descent algorithm with the momentum of 0.9 and weight decay ratio 1e-4 to optimize the training process. The learning rate is set to 3e-3 for Office-Home, Office-31 and DomainNet, 5e-5 for VisDA-2017 since it can easily converge. The batch size is set to 64.

4.2 Comparison to SoTA

We compare our method with state-of-the-art methods on UDA tasks, including MinEnt (Grandvalet et al., 2005), DAN (Long et al., 2015), DANN (Ganin & Lempitsky, 2015), ADR (Saito et al., 2017), CDAN+E (Long et al., 2018), CDAN+BSP (Chen et al., 2019), CDAN+TN (Wang et al., 2019), rRGrad+CAT (Deng et al., 2019), MCD (Saito et al., 2018), SWD (Lee et al., 2019), AFN (Xu

[2]The addition operation is not included in the 1st layer.

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
ResNet-50	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
DANN	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
MinEnt	80.3	75.5	75.8	48.3	77.9	27.3	69.7	40.2	46.5	46.6	79.3	16.0	57.0
MCD	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
ADR	87.8	79.5	83.7	65.3	92.3	61.8	88.9	73.2	87.8	60.0	85.5	32.3	74.8
SWD	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
CDAN+E	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
AFN	93.6	61.3	84.1	70.6	94.1	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
BNM	89.6	61.5	76.9	55.0	89.3	69.1	81.3	65.5	90.0	47.3	89.1	30.1	70.4
MSTN+DSBN	94.7	86.7	76.0	72.0	95.2	75.1	87.9	81.3	91.1	68.9	88.3	45.5	80.2
CGDM	93.7	82.7	73.2	68.4	92.9	94.5	88.7	82.1	93.4	82.5	86.8	49.2	82.3
CGDM*	92.8	85.1	76.3	64.5	91.0	93.2	81.3	79.3	92.4	83.0	85.6	44.8	80.8
SHOT	94.3	88.5	80.1	57.3	93.1	93.1	80.7	80.3	91.5	89.1	86.3	58.2	82.9
SHOT*	95.5	87.5	80.1	54.5	93.6	94.2	80.2	80.9	90.0	89.9	87.1	58.4	82.7
TVT$^\circ$	92.9	85.6	77.5	60.5	93.6	98.2	89.4	76.4	93.6	**92.0**	91.7	55.7	83.9
Baseline-B	97.7	48.1	86.6	61.6	78.1	63.4	**94.7**	10.3	87.7	47.7	**94.4**	35.5	67.1
CGDM-B*	96.3	87.1	**86.8**	**83.5**	92.2	98.3	91.6	78.5	96.3	48.4	89.4	39.0	82.3
SHOT-B*	**97.9**	90.3	86.0	73.4	**96.9**	**98.8**	94.3	54.8	95.4	87.1	93.4	62.7	85.9
Ours-B	97.1	**90.5**	82.4	77.5	96.6	96.1	93.6	**88.6**	**97.9**	86.9	90.3	**62.8**	**88.4**

Table 1: Comparison with SoTA methods on VisDA-2017. "S/B" implies the DeiT-small/DeiT-base backbone respectively. $*$ indicates the results are reproduced by ourselves. $\circ$ implies its pretrained model is trained on ImageNet21K instead of ImageNet1K. The best performance is marked as **bold**.

Method	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Avg.
ResNet-50	44.9	66.3	74.3	51.8	61.9	63.6	52.4	39.1	71.2	63.8	45.9	77.2	59.4
MinEnt	51.0	71.9	77.1	61.2	69.1	70.1	59.3	48.7	77.0	70.4	53.0	81.0	65.8
MCC	56.3	77.3	80.3	67.0	77.1	77.0	66.2	55.1	81.2	73.5	57.4	84.1	71.0
BNM	56.7	77.5	81.0	67.3	76.3	77.1	65.3	55.1	82.0	73.6	57.0	84.3	71.1
ATDOC-NA	58.3	78.8	82.3	69.4	78.2	78.2	67.1	56.0	82.7	72.0	58.2	85.5	72.2
CDAN+E	54.6	74.1	78.1	63.0	72.2	74.1	61.6	52.3	79.1	72.3	57.3	82.8	68.5
SAFN	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
CADA-P	56.9	76.4	80.7	61.3	75.2	75.2	63.2	54.5	80.7	73.9	61.5	84.1	70.2
DCAN	54.5	75.7	81.2	67.4	74.0	76.3	67.4	52.7	80.6	74.1	59.1	83.5	70.5
SHOT	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
SHOT*	56.2	77.6	81.6	67.5	78.2	78.8	67.8	54.0	82.0	72.5	58.8	84.5	71.6
TVT$^\circ$	**74.9**	**86.8**	**89.5**	**82.8**	**88.0**	**88.3**	**79.8**	**71.9**	**90.1**	**85.5**	**74.6**	**90.6**	**83.6**
Baseline-S	55.6	73.0	79.4	70.6	72.9	76.3	67.5	51.0	81.0	74.5	53.2	82.7	69.8
Ours-S	60.6	79.5	82.4	75.6	81.0	82.3	72.5	56.7	84.4	77.0	59.1	85.5	74.7
Baseline-B	61.8	79.5	84.3	75.4	78.8	81.2	72.8	55.7	84.4	78.3	59.3	86.0	74.8
CGDM-B*	67.1	83.9	85.4	77.2	83.3	83.7	74.6	64.7	85.6	79.3	69.5	87.7	78.5
SHOT-B*	67.1	83.5	85.5	76.6	83.4	83.7	76.3	65.3	85.3	80.4	66.7	83.4	78.1
Ours-B	68.8	85.0	86.9	81.5	87.1	87.3	79.6	63.3	88.2	82.0	66.0	**90.6**	80.5

Table 2: Comparison with SoTA methods on Office-Home. The best performance is marked as **bold**.

et al., 2019), MSTN+DSBN (Chang et al., 2019), SAFN (Xu et al., 2019), SAFN+ENT (Xu et al., 2019), CADA-P (Kurmi et al., 2019), BNM (Cui et al., 2020), DCAN (Li et al., 2020), SHOT (Liang et al., 2020), MCC (Jin et al., 2020), ATDOC-NA (Liang et al., 2021), CGDM (Du et al., 2021) and TVT (Yang et al.). The results are shown in Table 1, 2, 3 and 4.

For Office-Home, Office-31 and DomainNet, as most of the methods use ResNet-50 as their backbones, we provide results with DeiT-small as our backbone for a fair comparison, which has a comparable model size as ResNet-50, but we also show the results using DeiT-base. And for VisDA-2017, we adopt the DeiT-base backbone for fair comparisons, where other methods utilize ResNet-101 for their results.

The "Baseline-S/B" indicates directly training a DeiT-small/DeiT-base on the source domain and testing on the target domain. The baseline shows a competitive result even compared to other SoTA methods on most datasets. It demonstrates that Transformers has better generalization ability over ConvNets. We also provide some insights about why transformers can generalize well from source domain to target domain in the supplementary materials. To further eliminate the unfairness of using different backbones, we reproduce the results of SHOT and CGDM (marked as "*"), and replace their backbones with DeiT-base as the same as ours, denoted as "-B*".

MCD	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	15.4	25.5	3.3	44.6	31.2	24.0
inf	24.1	-	24.0	1.6	35.2	19.7	20.9
pnt	31.1	14.8	-	1.7	48.1	22.8	23.7
qdr	8.5	2.1	4.6	-	7.9	7.1	6.0
rel	39.4	17.8	41.2	1.5	-	25.2	25.0
skt	37.3	12.6	27.2	4.1	34.5	-	23.1
Avg.	28.1	12.5	24.5	2.4	34.1	21.2	20.5

CDAN	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	13.5	28.3	9.3	43.8	30.2	25.0
inf	18.9	-	21.4	1.9	36.3	21.3	20.0
pnt	29.6	14.4	-	4.1	45.2	27.4	24.2
qdr	11.8	1.2	4.0	-	9.4	9.5	7.2
rel	36.4	18.3	40.9	3.4	-	24.6	24.7
skt	38.2	14.7	33.9	7.0	36.6	-	26.1
Avg.	27.0	12.4	25.7	5.1	34.3	22.6	21.2

BNM	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg.	37.6	10.3	31.4	4.2	40.9	27.3	25.3

SWD	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	14.7	31.9	10.1	45.3	36.5	27.7
inf	22.9	-	24.2	2.5	33.2	21.3	20.0
pnt	33.6	15.3	-	4.4	46.1	30.7	26.0
qdr	15.5	2.2	6.4	-	11.1	10.2	9.1
rel	41.2	18.1	44.2	4.6	-	31.6	27.9
skt	44.2	15.2	37.3	10.3	44.7	-	30.3
Avg.	31.5	13.1	28.8	6.4	36.1	26.1	23.6

CGDM	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	16.9	35.3	10.8	53.5	36.9	30.7
inf	27.8	-	28.2	4.4	48.2	22.5	26.2
pnt	37.7	14.5	-	4.6	59.4	33.5	30.0
qdr	14.9	1.5	6.2	-	10.9	10.2	8.7
rel	49.4	20.8	47.2	4.8	-	38.2	32.0
skt	50.1	16.5	43.7	11.1	55.6	-	35.4
Avg.	36.0	14.0	32.1	7.1	45.5	28.3	27.2

Base-S	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	18.8	40.5	14.4	57.1	44.3	35.0
inf	38.8	-	37.3	3.9	53.0	30.2	32.6
pnt	45.5	18.5	-	4.7	58.9	35.2	32.6
qdr	2.8	0.3	0.4	-	0.7	4.1	1.7
rel	46.5	17.7	44.6	6.2	-	30.0	29.0
skt	52.7	14.9	37.8	15.6	48.9	-	34.0
Avg.	37.3	14.0	32.1	9.0	43.7	28.8	27.5

Ours-S	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	24.2	47.0	22.3	64.3	50.6	41.7
inf	45.3	-	45.3	6.6	62.8	38.3	39.7
pnt	53.6	20.4	-	10.6	63.9	42.4	38.2
qdr	2.8	0.2	0.6	-	0.7	4.2	1.7
rel	47.1	17.9	45.0	7.9	-	31.7	29.9
skt	61.0	19.3	46.8	22.8	59.2	-	41.8
Avg.	42.0	16.4	36.9	14.0	50.2	33.4	32.2

Base-B	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	24.2	49.8	16.8	65.3	53.3	41.9
inf	47.0	-	44.5	5.1	61.2	37.5	39.1
pnt	52.0	21.9	-	7.2	64.4	41.8	37.5
qdr	2.9	0.3	0.3	-	0.5	4.7	1.7
rel	48.4	18.9	47.2	6.4	-	31.5	30.5
skt	59.3	18.8	44.8	17.6	57.5	-	39.6
Avg.	41.9	16.8	37.3	10.6	49.8	33.8	31.7

Ours-B	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	27.9	57.6	27.9	73.0	58.8	49.0
inf	58.6	-	53.4	9.6	71.1	47.6	48.1
pnt	60.7	24.0	-	13.0	69.8	49.6	43.4
qdr	2.9	0.4	0.3	-	0.7	4.7	1.8
rel	49.3	18.7	47.8	9.4	-	33.5	31.7
skt	66.8	23.7	54.6	27.5	68.0	-	48.1
Avg.	47.7	18.9	42.7	17.5	56.5	38.8	**37.0**

Table 4: Comparison with SoTA methods on DomainNet. "Base" is the Baseline.

From Table 1, 2, 3 and 4, it can be seen that our method outperforms the baseline with a large margin on all four datasets, *e.g.* nearly 21% on VisDA. With our improvements, the new Transformer with cross-attention module shows a much better generalization power, and achieves the best performance on VisDA-2017 compared to other SoTAs methods. It further implies the effectiveness of our method on the UDA task.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
ResNet-50	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DANN	79.7	82.0	68.2	96.9	67.4	99.1	82.2
CDAN+E	92.9	94.1	71.0	98.6	69.3	**100.**	87.7
rRGrad+CAT	90.8	94.4	72.2	98.0	70.2	**100.**	87.6
SAFN+ENT	90.7	90.1	73.0	98.6	70.2	99.8	87.1
CDAN+BSP	93.0	93.3	73.6	98.2	72.6	**100.**	88.5
CDAN+TN	94.0	95.7	73.4	98.7	74.2	**100.**	89.3
SHOT	94.0	90.1	74.7	98.4	74.3	99.9	88.6
SHOT*	93.8	91.8	74.8	98.2	74.1	99.8	88.8
TVT°	96.4	96.4	**84.9**	**99.4**	**86.1**	**100.**	**93.8**
Baseline-S	87.6	86.9	74.9	97.7	73.5	99.6	86.7
Ours-S	94.6	93.5	78.4	98.2	78.0	99.6	90.4
Baseline-B	90.8	90.4	76.8	98.2	76.4	**100.**	88.8
CGDM-B*	94.6	95.3	78.8	97.6	81.2	99.8	91.2
SHOT-B*	95.3	94.3	79.4	99.0	80.2	**100.**	91.4
Ours-B	**97.0**	**96.7**	81.1	99.0	81.9	**100.**	92.6

Table 3: Comparison with SoTA methods on Office-31. The best performance is marked as **bold**.

Taking a closer look at the results, for the hard categories, such as "person" in VisDA-2017 dataset, the baseline is very low, which indicates the initial model of our method has a poor classification ability on this category, leading to the pseudo labels with more noise. Even with such a poor baseline and poor quality of pseudo labels, our method can still achieve a much higher performance boost (from 10.3% to 88.6%). It suggests that our method has a great robustness to the labeling noise and can overcome the noise problem to some extent. In Table 4, although the average performance of our methods outperforms other approaches by a large margin, it is noticed that the results of qdr-to-$others$ is much lower due to the low baseline. It is because the Transformer is weak in shape modeling caused by its hard patch partition (Naseer et al., 2021).

We can see that TVT achieves a better result on Office-Home and Office-31. Because TVT utilizes ViT (Dosovitskiy et al., 2020) as backbone which is pretrained on ImageNet21K. While the pretrained model of our CDTrans and other UDA methods are trained on ImageNet1K.

4.3 Ablation Study

4.3.1 Different Pseudo Labeling

We have conducted experiments on different pseudo labeling methods to verify their influence on the final performance. The results on VisDa-2017 are listed in Table. 5. RPLL (Zheng & Yang, 2021) and MRKLD+LRENT (Zou et al., 2019) are two commonly used pseudo-label generation methods, we reproduce their pseudo-label generations on our baseline to compare with our proposed pseudo labeling method. Rec_s, Rec_t means the recall of the selected training pairs in the source and target

Pseudo labels	Rec_s	Rec_t	Prec	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
One-way-source	100.	6.6	90.6	96.1	52.7	85.5	69.6	95.0	90.2	95.1	66.6	88.8	54.6	95.4	29.5	76.6
One-way-target	8.0	100.	76.3	98.2	32.0	87.7	84.1	95.5	89.9	98.3	66.8	95.7	57.5	95.6	22.0	76.9
Two-way	100.	100.	81.8	97.5	49.6	88.7	73.9	94.6	85.8	96.6	58.6	93.3	63.6	94.8	27.9	77.1
Tw + Ca	97.8	94.8	91.3	98.1	86.9	87.9	80.9	97.9	97.3	96.8	85.3	97.6	83.2	94.0	54.4	88.4
RPLL	-	-	-	98.4	63.4	85.8	68.8	97.0	95.4	97.77	59.3	96.2	57.2	96.2	48.1	80.3
MRKLD+LRENT	-	-	-	97.8	77.3	81.4	64.3	94.6	93.9	93.3	77.5	93.1	74.9	92.6	59.0	83.3
Groundtruth	100.	100.	100.	97.9	89.1	92.3	91.9	98.4	97.2	97.5	86.8	98.6	90.7	96.3	60.0	91.5

Table 5: Comparison among different pseudo labeling methods on VisDa-2017. Rec_s, Rec_t express the recall of pseudo labels in source and target data, while $Prec$ represents the accuracy of the pairs. "One-way-source/target" denotes only using the source/target pair set for training. "Tw+Ca" implies the proposed two-way center-aware labeling method.

L_s	L_{s+t}	L_t	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
cls	-	-	97.7	48.1	86.6	61.6	78.1	63.4	94.7	10.3	87.7	47.7	94.4	35.5	67.1
-	-	*cls*	98.3	85.0	88.0	76.3	98.1	96.1	96.9	61.1	97.2	85.5	94.6	54.9	86.0
cls	-	*cls*	98.3	87.4	89.1	77.3	98.0	97.4	95.4	69.5	97.1	86.3	95.3	49.5	86.7
cls	*cls*	*cls*	98.2	88.4	88.0	76.8	98.2	97.2	95.6	80.1	97.1	84.7	94.5	54.1	87.7
cls	*dtl*	*cls*	98.0	86.9	87.9	80.9	97.9	97.3	96.8	85.3	97.6	83.2	94.0	54.4	88.4

Table 6: Comparison among different losses on VisDa-2017. L_s,L_t and L_{s+t} represent the loss used in source, target and source+target branches respectively. cls and dtl imply the classification loss and the distillation loss.

data, while $Prec$ represents the accuracy of the pairs. "One-way-source" and "One-way-target" denote only using the pair set $\mathbb{P}_S$ in Eq. 3 or $\mathbb{P}_T$ in Eq. 4 for training. "Two-way" indicates results of using the union of $\mathbb{P}_s$ and $\mathbb{P}_t$ without the center-aware strategy. "Tw+Ca" implies our two-way center-aware labeling method, and "Groundtruth" means all training pairs are from groundtruthes.

By looking at Rec_s and Rec_t in Table. 5, it can be found that the one-way methods have an apparent bias on either source or target data, and its results are lower than the two-way method. By comparing "Two-way" and "Tw+Ca", we can conclude that although the center-aware method filters the training pair and slightly reduces the recall, it largely improves the precision and leads to a better final performance. We also find that our two-way center-aware labeling method achieves a very high result, not only better than other pseudo-label generation methods, but also very close to the upper bound trained with groundtruth pairs.

4.3.2 Different Losses

As there are three losses in our method, we conduct another experiment to verify the effectiveness of each loss on VisDa-2017, as shown in Table. 6. "cls" in L_{s+t} denotes that we replace the distillation loss with a classification loss for the source-target branch. We can see that the 3rd row with both L_s and L_t having classification loss achieves a better result than the first row where only L_s has the cls loss, which means the target branch with the pseudo labels is helpful to improve the UDA result. With the addition of "cls" in L_{s+t}, the performance is further improved, which demonstrates the advantages of using the cross-attention module for feature alignment. Using "dtl" instead of "cls" on the source-target branch can further improve the results, showing the effectiveness of our distillation loss.

5 Conclusion

In this paper, we tackle the problem of unsupervised domain adaptation by introducing the cross-attention module into Transformer in a novel way. We propose a new network structure CDTrans which is a pure transformer-based structure with three branches, and we also propose to generate high-quality pseudo labels using a two-way center-aware labeling method. Training CDTrans using the generated high-quality pseudo labels yields a robust solution and also achieves state-of-the-art results on four popular UDA datasets, outperforming previous methods by a large margin. We believe that transformer-based approaches will have great potential in the UDA community, and our work, as one of the first attempts along this direction, has pushed forward the frontiers and shed lights for future research.

REFERENCES

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, 2019.

Boyu Chen, Peixia Li, Baopu Li, Chuming Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Psvit: Better vision transformer via token pooling and attention sharing. *arXiv preprint arXiv:2108.03428*, 2021.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, 2019.

Jaehoon Choi, Minki Jeong, Taekyung Kim, and Changick Kim. Pseudo-labeling curriculum for unsupervised domain adaptation. *arXiv preprint arXiv:1908.00262*, 2019.

Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, 2020.

Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *CVPR*, 2021.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *NIPS*, 2006.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*, 2020.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021a.

Liang Han, Pichao Wang, Zhaozheng Yin, Fan Wang, and Hao Li. Context and structure mining network for video object detection. *IJCV*, 2021b.

Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. *arXiv preprint arXiv:2102.04378*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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 *ICML*, 2018.

Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. *arXiv preprint arXiv:2102.10772*, 2021.

Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *ICML*, 2020.

Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *ECCV*, 2020.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

Vinod Kumar Kurmi, Shanu Kumar, and Vinay P Namboodiri. Attending to discriminative certainty for domain adaptation. In *CVPR*, pp. 491–500, 2019.

Seiichi Kuroki, Nontawat Charoenphakdee, Han Bao, Junya Honda, Issei Sato, and Masashi Sugiyama. Unsupervised domain adaptation based on source-guided discrepancy. In *AAAI*, 2019.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019.

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*, 2013.

Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *CVPR*, 2021a.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *AAAI*, 2019.

Shuai Li, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Category dictionary guided unsupervised domain adaptation for object detection. In *AAAI*, 2021b.

Shuang Li, Chi Liu, Qiuxia Lin, Binhui Xie, Zhengming Ding, Gao Huang, and Jian Tang. Domain conditioned adaptation network. In *AAAI*, volume 34, pp. 11386–11393, 2020.

Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, and Pichao Wang. Lifting transformer for 3d human pose estimation in video. *arXiv preprint arXiv:2103.14304*, 2021c.

Xiangyu Li, Yonghong Hou, Pichao Wang, Zhimin Gao, Mingliang Xu, and Wanqing Li. Transformer guided geometry model for flow-based unsupervised visual odometry. *Neural Computing and Applications*, 2021d.

Xiangyu Li, Yonghong Hou, Pichao Wang, Zhimin Gao, Mingliang Xu, and Wanqing Li. Trear: Transformer-based rgb-d egocentric action recognition. *T-CDS*, 2021e.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, pp. 16632–16642, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *NeurIPS*, 2018.

Pietro Morerio, Riccardo Volpi, Ruggero Ragonesi, and Vittorio Murino. Generative pseudo-label refinement for unsupervised domain adaptation. In *WACV*, 2020.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021.

Poojan Oza, Vishwanath A Sindagi, Vibashan VS, and Vishal M Patel. Unsupervised domain adaption of object detectors: A survey. *arXiv preprint arXiv:2105.13502*, 2021.

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.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*, 2017.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *ACL*, 2019.

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

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

Vibashan VS, Vikram Gupta, Poojan Oza, Vishwanath A Sindagi, and Vishal M Patel. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In *CVPR*, 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable normalization: Towards improving transferability of deep neural networks. 2019.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *TIST*, 2020.

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *iccv*, pp. 1426–1435, 2019.

Guanglei Yang, Hao Tang, Zhun Zhong, Mingli Ding, Ling Shao, Nicu Sebe, and Elisa Ricci. Transformer-based source-free domain adaptation. *arXiv preprint arXiv:2105.14138*, 2021.

Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation. *arXiv preprint arXiv:2108.05988*.

Zitong Yu, Xiaobai Li, Pichao Wang, and Guoying Zhao. Transrppg: Remote photoplethysmography transformer for 3d mask face presentation attack detection. *IEEE SPL*, 2021.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021a.

Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021b.

Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *NIPS*, 2019.

Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, 2018.

Yabin Zhang, Bin Deng, Hui Tang, Lei Zhang, and Kui Jia. Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *PAMI*, 2020.

Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E Gonzalez, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, et al. A review of single-source deep unsupervised visual domain adaptation. *T-NNLS*, 2020.

Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *IJCV*, 2021.

Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019.