Supplementary Material : Aligning Non-Causal Factors for Transformer-Based Source-Free Domain Adaptation

Sunandini Sanyal* Ashish Ramayee Asokan* Suvaansh Bhambri Pradyumna YM
Akshay Kulkarni Jogendra Nath Kundu R Venkatesh Babu
Vision and AI Lab, Indian Insitute of Science, Bengaluru

The supplementary material provides further details of the proposed approach, additional quantitative results, ablations, and implementation details. We have released our code on our project page: `https://val.cds.iisc.ac.in/C-SFTrans/`. The remainder of the supplementary material is organized as follows:

- Section 1: Proposed Approach (Table 1, Algorithm 1)
- Section 2: Implementation Details
  - Datasets (Section 2.1)
  - Style augmentations (Section 2.2)
  - Experimental Settings (Section 2.3)
- Section 3: Additional Comparisons (Tables 2)
- Section 4: Ablations on target-side goal task training (Tables 3, 4, and 5)

1. Proposed Approach

We summarize all the notations used in the paper in Table 1. The notations are grouped into the following 6 categories - models, transformers, datasets, spaces, losses, and criterion. Our proposed method has been outlined in Algorithm 1

Table 1. **Notation Table**

	Symbol	Description
Models	f	Backbone feature extractor
	f_g	Goal task classifier
	f_n	Style classifier
Transformers	z_c	Class token of last layer
	z_n	Style token of last layer
	N_P	Number of patch tokens
	h_n^l	Non-causal heads of layer l
	h^l	All attention-heads of layer l
	$h^l \setminus h_n^l$	Causal heads of layer l
	W_K	Key weights
	W_Q	Query weights
	W_V	Value weights
Datasets	$\mathcal{D}_s$	Labeled source dataset
	$\mathcal{D}_t$	Unlabeled target dataset
	a_i	Augmentation function i
	$\mathcal{D}_s^{[i]}$	i^{th} augmented source dataset
	$\mathcal{D}_t^{[i]}$	i^{th} augmented target dataset
	(x_s, y_s)	Labeled source sample
	$(x_s^{[i]}, y_s, y_d)$	Augmented source sample
	x_t	Unlabeled target sample
	$(x_t^{[i]}, y_d)$	Target augmented sample
	x	Clean input sample
	x_{SCI}	Style Characterizing Input
Spaces	$\mathcal{X}$	Input space
	$\mathcal{C}_g$	Label set for goal task
	$\mathcal{Z}_c$	Class token feature space
	$\mathcal{Z}_n$	Style token feature space
	$\mathcal{Z}_1, \dots, \mathcal{Z}_{N_P}$	Patch tokens
Losses	$\mathcal{L}_{style}$	Style Classification loss
	$\mathcal{L}_{cls}$	Task Classification loss
	$\mathcal{L}_{ent}$	Entropy loss
	$\mathcal{L}_{div}$	Diversity loss
Criterion	β_{1_i}	Importance weight for style feature
	β_{2_i}	Importance weight for task feature
	CIS_i	Causal Influence Score for head i
	τ	Threshold

Target adaptation losses. We use the Information Maximization loss [8] that consists of entropy loss $\mathcal{L}_{ent}$ and diversity loss $\mathcal{L}_{div}$.

$$\mathcal{L}_{ent} = -\mathop{\mathbb{E}}_{x_t \in \mathcal{X}} \sum_{k=1}^{K} \delta_k(f_g(z_c)) \log \delta_k(f_g(z_c)) \tag{1}$$

$$\mathcal{L}_{div} = \sum_{k=1}^{K} \hat{p_k} \log \hat{p_k} = KL(\hat{p}, \frac{1}{K} 1_K) - \log K \tag{2}$$

where $\delta_k(b) = \frac{\exp(b_k)}{\sum_i \exp(b_i)}$ is the k^{th} element of softmax output of $b \in \mathbb{R}^K$. The entropy loss $\mathcal{L}_{ent}$ ensures that the model predicts more confidently for a particular label and the diversity loss $\mathcal{L}_{div}$ ensures that the predictions are well-balanced across different classes. We optimize all parameters of the transformer backbone h, except the non-causal heads h_n^l.

*Equal Contribution

$$\min_{h^l \backslash h_n^l, f_g} \mathop{\mathbb{E}}_{\mathcal{D}_t}[\mathcal{L}_{ent} + \mathcal{L}_{div}] \tag{3}$$

Pseudo-labeling. We use the clustering method of SHOT [8] to obtain pseudo-labels. At first, the centroid of each class is calculated using the following,

$$c_k = \frac{\sum_{x_t \in \mathcal{X}} \delta_k(f_g(z_c)) z_c}{\sum_{x_t \in \mathcal{X}} \delta_k(f_g(z_c))} \tag{4}$$

The closest centroid is chosen as the pseudo-label for each sample using the following cosine distance formulation,

$$\hat{y_c} = \arg\min_k D_c(z_c, c_k) \tag{5}$$

where D_c denotes the cosine-distance in the class-token feature space $\mathcal{Z}_c$ between a centroid c_k and the input sample features z_c. In successive iterations, the centroids keep updating and the pseudo-labels get updates with respect to the new centroids.

Attention heads in vision transformers. A ViT takes an image x as input of size $H \times W \times C$ and divides it into N_P patches of size (P, P) each. The total number of patches are $N_P = \frac{H \times W}{P^2}$. In every layer, l, a head h_i^l takes the patches as input and transforms a patch into K, Q, V using the weights W_K, W_Q, W_V, respectively. The self-attention [16] is computed as follows,

$$h_i^l = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{6}$$

where d_k represents the dimension of the keys/queries.

2. Implementation details

2.1. Datasets

We use four standard object classification benchmarks for DA to evaluate our approach. The **Office-Home** dataset [17] consists of images from 65 categories of everyday objects from four domains - Art **(Ar)**, Clipart **(Cl)**, Product **(Pr)**, and Real World **(Rw)**. Office-31 [12] is a simpler benchmark containing images from 31 categories belonging to three domains of objects in office settings - Amazon **(A)**, Webcam **(W)**, and DSLR **(D)**. **VisDA** [11] is a large-scale benchmark containing images from two domains - 152,397 synthetic source images and 55,388 real-world target images. Lastly, **DomainNet** [10] is the largest and the most challenging dataset due to severe class imbalance and diversity of domains. It contains 345 categories of objects from six domains - Clipart **(clp)**, Infograph **(inf)**, Painting **(pnt)**, Quickdraw **(qdr)**, Real **(rel)**, Sketch **(skt)**.

Algorithm 1 C-SFTrans Training Algorithm

Vendor-side training

Input: Let $\mathcal{D}_s$ be the source data, $\mathcal{D}_{sty}$ be the style dataset, ImageNet pre-trained DeiT-B backbone h from [19], randomly initialized goal classifier f_g and randomly initialized style classifier f_n.

Non-causal attention heads selection ▷ Fig. 3A (main paper)

for $iter < MaxTaskIters$ **do**:
  Sample batch x_i from $\mathcal{D}_s$
  Construct x_{SCI} from x_i
  Compute $\mathcal{A}_i^l$ using Eq. 3 (main paper)
  Compute $\mathcal{L}_{cls}$ using Eq. 4 (main paper)
  update β_{1j}, β_{2j} for head j by minimizing $\mathcal{L}_{cls}$
end for
$h_n^l = \{h : h \in h_l, CIS_h > \tau\}$
for $iter < MaxIter$ **do**:
  Goal task training ▷ Fig. 3B (main paper)
  for $iter < MaxTaskIters$ **do**:
    Sample batch from $\mathcal{D}_s$
    Compute $\mathcal{L}_{cls}$ using Eq. 6 (main paper)
    update $\theta_{h^l} \setminus \theta_{h_n^l}$, θ_{f_g} by minimizing $\mathcal{L}_{cls}$
  end for
  Style classifier training ▷ Fig. 3B (main paper)
  for $iter < MaxDomainIters$ **do**:
    Sample batch from $\mathcal{D}_{sty}^s$
    Compute $\mathcal{L}_{dom}$ using Eq. 1 (main paper)
    update $\theta_{h_n^l}$, θ_{f_n} by minimizing $\mathcal{L}_{dom}$
  end for
  ▷ The two steps are carried out alternatively
end for

Client-side training

Input: Target data $\mathcal{D}_t$, Target augmented DRI data $\mathcal{D}_t^{[i]}$, source-side pretrained backbone h, goal classifier f_g and domain classifier f_d.

for $iter < MaxIter$ **do**:
  Goal Task Training ▷ Fig. 3B (main paper)
  for $iter < MaxTaskIters$ **do**:
    Sample batch from $\mathcal{D}_t$
    Compute $\mathcal{L}_{im}$ and $\mathcal{L}_{div}$ using Eq. 1, 2 (suppl.)
    update $\theta_{h^l} \backslash \theta_{h_n^l}$, θ_{f_g} by minimizing $\mathcal{L}_{im} + \mathcal{L}_{div}$
  end for
  Style classifier training ▷ Fig. 3B (main paper)
  for $iter < MaxDomainIters$ **do**:
    Sample batch from $\mathcal{D}_{sty}^t$
    Compute $\mathcal{L}_{dom}$ using Eq. 1 (main paper)
    update $\theta_{h_n^l}$, θ_{f_n} by minimizing $\mathcal{L}_{dom}$
  end for
  ▷ The two steps are carried out alternatively.
end for

Table 2. Single-Source Domain Adaptation (SSDA) results on the DomainNet dataset. * indicates results taken from [13].

ResNet-101 [2]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	19.3	37.5	11.1	52.2	41.0	32.2
inf	30.2	-	31.2	3.6	44.0	27.9	27.4
pnt	39.6	18.7	-	4.9	54.5	36.3	30.8
qdr	7.0	0.9	1.4	-	4.1	8.3	4.3
rel	48.4	22.2	49.4	6.4	-	38.8	33.0
skt	46.9	15.4	37.0	10.9	47.0	-	31.4
Avg.	34.4	15.3	31.3	7.4	40.4	30.5	**26.6**

CDAN [9]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	20.4	36.6	9.0	50.7	42.3	31.8
inf	27.5	-	25.7	1.8	34.7	20.1	22.0
pnt	42.6	20.0	-	2.5	55.6	38.5	31.8
qdr	21.0	4.5	8.1	-	14.3	15.7	12.7
rel	51.9	23.3	50.4	5.4	-	41.4	34.5
skt	50.8	20.3	43.0	2.9	50.8	-	33.6
Avg.	38.8	17.7	32.8	4.3	41.2	31.6	**27.7**

MIMFTL [1]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	15.1	35.6	10.7	51.5	43.1	31.2
inf	32.1	-	31.0	2.9	48.5	31.0	29.1
pnt	40.1	14.7	-	4.2	55.4	36.8	30.2
qdr	18.8	3.1	5.0	-	16.0	13.8	11.3
rel	48.5	19.0	47.6	5.8	-	39.4	32.1
skt	51.7	16.5	40.3	12.3	53.5	-	34.9
Avg.	38.2	13.7	31.9	7.2	45.0	32.8	**28.1**

MDD+SCDA [21]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	20.4	43.3	15.2	59.3	46.5	36.9
inf	32.7	-	34.5	6.3	47.6	29.2	30.1
pnt	46.4	19.9	-	8.1	58.8	42.9	35.2
qdr	31.1	6.6	18.0	-	28.8	22.0	21.3
rel	55.5	23.7	52.9	9.5	-	45.2	37.4
skt	55.8	20.1	46.5	15.0	56.7	-	38.8
Avg.	44.3	18.1	39.0	10.8	50.2	37.2	**33.3**

DeiT-B [14]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	24.3	49.6	15.8	65.3	52.1	41.4
inf	45.9	-	45.9	6.7	61.4	39.5	39.9
pnt	53.2	23.8	-	6.5	66.4	44.7	38.9
qdr	31.9	6.8	15.4	-	23.4	20.6	19.6
rel	59.0	25.8	56.3	9.16	-	44.8	39.0
skt	60.6	20.6	48.4	16.5	61.2	-	41.5
Avg.	50.1	20.3	43.1	10.9	55.5	40.3	**36.7**

SHOT-B [8]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	27.0	49.7	16.5	65.4	53.2	46.1
inf	46.4	-	45.9	7.4	60.6	40.1	40.1
pnt	54.6	25.7	-	8.1	66.3	49.0	40.7
qdr	33.3	6.8	15.5	-	23.8	24.0	20.7
rel	59.3	28.1	57.4	9.0	-	47.3	40.2
skt	64.0	26.5	55.0	18.2	63.8	-	45.5
Avg.	51.5	26.6	44.7	11.8	56.0	42.7	**38.9**

CDTrans* [19]	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**

SSRT-B* [13]	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	33.8	60.2	19.4	75.8	59.8	49.8
inf	55.5	-	54.0	9.0	68.2	44.7	46.3
pnt	61.7	28.5	-	8.4	71.4	55.2	45.0
qdr	42.5	8.8	24.2	-	37.6	33.6	29.3
rel	69.9	37.1	66.0	10.1	-	58.9	48.4
skt	70.6	32.8	62.2	21.7	73.2	-	52.1
Avg.	60.0	28.2	53.3	13.7	65.3	50.4	**45.2**

C-SFTrans (Ours)	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	26.6	53.6	23.6	71.4	54.6	46.0
inf	55.9	-	51.7	11.4	69.6	46.0	46.9
pnt	60.0	25.2	-	14.3	71.2	51.1	44.4
qdr	43.2	8.2	17.4	-	40.2	28.8	27.5
rel	60.4	28.1	56.5	12.2	-	49.8	41.4
skt	66.7	26.5	56.2	25.1	71.0	-	49.1
Avg.	57.2	22.9	47.1	17.3	64.7	46.1	**42.5**

2.2. Style augmentations

We construct novel stylised images using 5 label-preserving augmentations on the original clean images to enable non-causal factor alignment during the training process. The augmentations are as follows:

1. **FDA augmentation:** We use the FDA augmentation [20] to generate stylized images based on a fixed set of style images [3]. In this augmentation, a given input image is stylized by interchanging the low-level frequencies between the FFTs of the input image and the reference style image.

2. **Weather augmentations:** We employ the frost and snow augmentations from [5] to simulate the weather augmentation. Specifically, we use the lowest severity of frost and snow (*severity* = 1) to augment the input images.

3. **AdaIN augmentation:** AdaIN [3] uses a reference style image to stylize a given input image by altering the feature statistics in an instance normalization (IN) layer [15]. We use the same reference style image set as in FDA, and set the augmentation strength to 0.5.

4. **Cartoon augmentation:** We employ the cartoonization augmentation from [5] to produce cartoon-style images with reduced texture from the input.

5. **Style augmentation:** We use the style augmentation from [4] that augments an input image through random style transfer. This augmentation alters the texture, contrast and color of the input while preserving its geometrical features.

2.3. Experimental settings

In all our experiments, we use the Stochastic Gradient Descent (SGD) optimizer [6] with a momentum of 0.9 and batch size of 64. We follow [8] and use label smoothing in the training process. For the source-side, we train the goal task classifier for 20 epochs, and the style classifier until it achieves 80% accuracy. On the target-side, we train the goal task classifier for 2 epochs, and use the same criteria for the style classifier as the source-side. The first 5 epochs of the source-side training are used for warm-up with a warm-up factor of 0.01. On the source-side, we use a learning rate of 8×10^{-4} for the VisDA dataset, and 8×10^{-3} for the remaining benchmarks. For the target-side goal task training, we use a learning rate of 5×10^{-5} for VisDA, 2×10^{-3} for DomainNet, and 8×10^{-3} for the rest. Our proposed method comprises an alternate training mechanism where the goal task training and style classifier training are done alternatively for a total of 25 rounds, which is equivalent to 50 epochs of target adaptation in [8]. For comparisons, we implement the source-free methods DIPE [18] and Feature Mixup [7] by replacing the backbone with DeiT-B. While CDTrans [19] uses the entire domain for training and evaluation with the DomainNet dataset, we follow the setup of [13] to ensure fair comparisons. We train on the *train* split and evaluate on the *test* split of each domain.

3. Additional comparisons

We present additional comparisons with the **DomainNet** benchmark in Table 2. Our method achieves the best results

Table 3. Sensitivity analysis of alternate training on Single-Source Domain Adaptation (SSDA) on Office-Home. The goal task epochs are varied from 1 to 5.

Epochs	Ar → Cl	Cl → Pr	Pr → Rw	Rw → Ar	Avg.
1	63.7	79.8	79.8	75.7	74.8
2	70.0	86.8	87.6	82.5	81.7
3	69.9	86.7	87.5	82.3	81.6
5	70.6	87.7	88.5	82.3	82.2

Table 4. Ablation study for the three components of the target-side goal task training. *SSPL* denotes self-supervised pseudo-labelling.

Method	$\mathcal{L}_{ent}$	$\mathcal{L}_{div}$	textitSSPL	Avg.
Source-Only	✗	✗	✗	76.4
C-SFTrans	✓	✗	✗	74.0
	✓	✓	✗	79.7 (+5.7)
	✓	✓	✓	81.7 (+7.7)

among the existing source-free prior arts and outperforms the source-free SHOT-B* by 3.6%. We also observe that C-SFTrans surpasses the non-source-free method CDTrans by an impressive 5.5%.

4. Ablations on target-side goal task training

(a) Target-side goal task training loss. Table 4 shows the influence of the three loss terms in the target-side goal task training - entropy loss $\mathcal{L}_{ent}$, diversity loss $\mathcal{L}_{div}$ and self-supervised pseudo-labeling *SSPL*. We observe that using $\mathcal{L}_{ent}$ alone produces lower results even compared to the source baseline. On the other hand, using both $\mathcal{L}_{ent}$ and $\mathcal{L}_{div}$ significantly improves the performance, which highlights the importance of the diversity term $\mathcal{L}_{div}$. Finally, we obtain the best results when all three components are used together for target-side adaptation, further showing the significance of the pseudo-labeling step.

(b) Sensitivity analysis of alternate training. In our proposed method, we perform style classifier training and goal task training in an alternate fashion, *i.e.* the task classifier f_g is trained for a few epochs, followed by the training of the style classifier f_n until it reaches a certain accuracy threshold (empirically set to 80%). In Table 3, we show the effect of varying the number of epochs of the goal task training from 1 to 5, and observe the impact on the goal task accuracy during non-causal factor alignment. We observe that 2 epochs of goal task training achieves the optimal balance between target accuracy and training effort. We observe that just a single epoch of task classifier training negatively impacts the goal task performance. While 3 epochs achieves the best performance, it involves significant training effort for merely 0.5% improvement in the task accuracy. Therefore, 2 epochs of goal task training achieves the optimal balance between target accuracy and training effort.

(c) Selection of non-causal heads. We select a set of non-causal attention heads based on their *Causal Influence Score* (CIS). We sort the CIS in descending order and select the top λ% of heads satisfying the condition *CIS* $> \tau$. In Table 5, we present the effect of altering this hyperparameter λ on the overall performance. We observed that with a lower value of λ, the pathways formed by non-causal heads do not adequately extract and learn the non-causal factors, which consequently hinders the domain-invariant alignment and leads to non-optimal task performance. Similarly, increasing λ too much reduces the ability of the network to learn causal factors and leads to lower performance. Overall, our approach is not very sensitive towards this hyperparameter.

Table 5. Sensitivity analysis on non-causal heads (%) for Single-Source DA on 4 settings of Office-Home

λ	Ar → Cl	Cl → Pr	Pr → Rw	Rw → Ar	Avg.
0.1	70.2	86.7	87.5	82.4	81.7
0.2	70.0	86.8	87.6	82.5	81.7
0.3	70.3	86.9	87.7	82.6	81.9
0.4	70.2	86.5	87.2	82.1	81.5

Table 6. Ablation study for the effect of augmentations for target-side goal task training.

No. of augs.	Ar → Cl	Cl → Pr	Pr → Rw	Rw → Ar	Avg.
3	64.3	79.9	84.6	80.0	77.2
6	70.0	86.8	87.6	82.5	81.7

(d) Effect of augmentations. Table 6 demonstrates that fewer augmentations for the style classifier significantly deteriorate the adaptation performance in comparison to the full set of augmentations. This indicates that a more complex style classification task better facilitates the non-causal factor alignment. However, due to the scarcity of more complex augmentations, we use the six outlined in Sec. 2.2

References

[1] Jian Gao, Yang Hua, Guosheng Hu, Chi Wang, and Neil M Robertson. Reducing distributional uncertainty by mutual information maximisation and transferable feature learning. In *ECCV*, 2020. 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 3

[4] Philip T Jackson, Amir Atapour-Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: Data augmentation via style randomization. In *CVPR Workshops*, 2019. 3

[5] Jung. imgaug. In *https://github.com/aleju/imgaug*, 2020. 3

[6] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 3

[7] Jogendra Nath Kundu, Akshay Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Kulkarni, Varun Jampani, and R. Venkatesh Babu. Balancing discriminability and transfer-

ability for source-free domain adaptation. In *ICML*, 2022. 3

[8] 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. 1, 2, 3

[9] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2017. 3

[10] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2

[11] 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. 2

[12] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2

[13] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *CVPR*, 2022. 3

[14] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 3

[15] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, 2017. 3

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[17] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 2

[18] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *CVPR*, 2022. 3

[19] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. CDTrans: Cross-domain transformer for unsupervised domain adaptation. In *ICLR*, 2022. 2, 3

[20] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020. 3

[21] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019. 3