Source-Free Domain Adaptation with Frozen Multimodal Foundation Model

Song Tang[1,2,3], Wenxin Su[1], Mao Ye[*4], and Xiatian Zhu[*5]
[1]University of Shanghai for Science and Technology [2]Universität Hamburg [3]ComOriginMat Inc.
[4]University of Electronic Science and Technology of China [5]University of Surrey
tangs@usst.edu.cn, {suwenxin43, cvlab.uestc}@gmail.com , xiatian.zhu@surrey.ac.uk

1. A Proof of Theorem 1.

Restatement of Theorem 1 *Given two random variables X, Y. Their mutual information $\mathrm{I}(X,Y)$ and KL divergence $D_{\mathrm{KL}}(X||Y)$ satisfy the unequal relationship as follows.*

$$-\mathrm{I}(X,Y) \le D_{\mathrm{KL}}(X||Y)\,. \tag{1}$$

Proof. Suppose the probability density function (PDF) of X and Y are $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$, respectively; their join PDF is $p(\boldsymbol{x},\boldsymbol{y})$. We have

$$\begin{aligned}\mathrm{I}(X,Y) &= \sum p(\boldsymbol{x},\boldsymbol{y}) \log \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})\cdot p(\boldsymbol{y})} \\ &= D_{\mathrm{KL}}\left(p(\boldsymbol{x},\boldsymbol{y})\,||\,p(\boldsymbol{x})\cdot p(\boldsymbol{y})\right).\end{aligned}$$

Well known, the KL divergence is non-negative [2]. Thus,

$$-\mathrm{I}(X,Y) \le 0 \le D_{\mathrm{KL}}(X||Y)$$

2. Evaluation Datasets

We evaluate four standard benchmarks below.

- **Office-31** [16] is a small-scaled dataset including three domains, i.e., Amazon (A), Webcam (W), and Dslr (D), all of which are taken of real-world objects in various office environments. The dataset has 4,652 images of 31 categories in total. Images in (A) are online e-commerce pictures. (W) and (D) consist of low-resolution and high-resolution pictures.
- **Office-Home** [23] is a medium-scale dataset that is mainly used for domain adaptation, all of which contains 15k images belonging to 65 categories from working or family environments. The dataset has four distinct domains, i.e., Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-word images (Rw).
- **VisDA** [12] is a challenging large-scale dataset with 12 types of synthetic to real transfer recognition tasks. The source domain contains 152k synthetic images (Sy), whilst the target domain has 55k real object images (Re) from the famous Microsoft COCO dataset.
- **DomainNet-126** [13] is another large-scale dataset. As a subset of DomainNet containing 600k images of 345 classes from 6 domains of different image styles, this dataset has 145k images from 126 classes, sampled from 4 domains, Clipart (C), Painting (P), Real (R), Sketch (S), as [17] identify severe noisy labels in the dataset.

3. Implementation Details

Souce model pre-training. For all transfer tasks on the three datasets, we train the source model θ_s on the source domain in a supervised manner using the following objective of the classic cross-entropy loss with smooth label, like other methods [8, 21, 25].

$$L_{\mathrm{s}}\left(\mathcal{X}_s,\mathcal{Y}_s;\theta_s\right) = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{c=1}^{C}\tilde{l}_{i,c}^{s}\log p_{i,c}^{s},$$

where n_s is the number of the source data, $p_{i,c}^s$ is the c-th element of $\boldsymbol{p}_i^s = \theta_s(\boldsymbol{x}_i^s)$ that is the category probability vector of input instance $\boldsymbol{x}_i^s$ after θ_s mapping; $\tilde{l}_{i,c}^s$ is the c-th element of the smooth label [11] $\tilde{\boldsymbol{l}}_i^s = (1-\sigma)\,\boldsymbol{l}_i^s + \sigma/C$, in which $\boldsymbol{l}_i^s$ is a one-hot encoding of hard label y_i^s and $\sigma = 0.1$. The source dataset is divided into the training set and testing set in a 0.9:0.1 ratio.

Network setting. The DIFO model contains two network branches. In the target model branch, the feature extractor consists of a deep architecture and a fully-connected layer followed by a batch-normalization layer. Same to the previous work [8, 10, 15, 24, 25], the deep architecture is transferred from the deep models pre-trained on ImageNet (i.e., ResNet-50 is used on **Office-31**, **Office-Home** and **DomainNet-126**, whilst ResNet-101 is adopted on **VisDA**). The ending classifier is a fully-connected layer with weight normalization. On the other hand, the ViL model branch chooses the most adopted CLIP as the implementation where the text encoder's transformer-based architecture follows modification proposed in [14] as the backbone. Regarding the image encoder, we adopt two versions corresponding to

[*] Corresponding author

Table 1. Full results (%) of Closed-set SFDA on **VisDA**. **SF** and **M** mean source-free and multimodal, respectively.

Method	Venue	**SF**	**M**	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Perclass
Source	-	-	-	60.7	21.7	50.8	68.5	71.8	5.4	86.4	20.2	67.1	43.3	83.3	10.6	49.2
DAPL-RN [3]	TNNLS23	✗	✓	97.8	83.1	88.8	77.9	97.4	91.5	94.2	79.7	88.6	89.3	92.5	62.0	86.9
PADCLIP-RN [6]	ICCV23	✗	✓	96.7	88.8	87.0	82.8	97.1	93.0	91.3	83.0	95.5	91.8	91.5	63.0	88.5
ADCLIP-RN [19]	ICCVW23	✗	✓	**98.1**	83.6	**91.2**	76.6	**98.1**	93.4	**96.0**	81.4	86.4	91.5	92.1	64.2	87.7
SHOT [8]	ICML20	✓	✗	95.0	87.4	80.9	57.6	93.9	94.1	79.4	80.4	90.9	89.8	85.8	57.5	82.7
NRC [25]	NIPS21	✓	✗	96.8	91.3	82.4	62.4	96.2	95.9	86.1	**90.7**	94.8	94.1	90.4	59.7	85.9
GKD [20]	IROS21	✓	✗	95.3	87.6	81.7	58.1	93.9	94.0	80.0	80.0	91.2	91.0	86.9	56.1	83.0
AaD [26]	NIPS22	✓	✗	97.4	90.5	80.8	76.2	97.3	96.1	89.8	82.9	95.5	93.0	92.0	64.7	88.0
AdaCon [1]	CVPR22	✓	✗	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
CoWA [7]	ICML22	✓	✗	96.2	89.7	83.9	73.8	96.4	**97.4**	89.3	86.8	94.6	92.1	88.7	53.8	86.9
SCLM [21]	NN22	✓	✗	97.1	90.7	85.6	62.0	97.3	94.6	81.8	84.3	93.6	92.8	88.0	55.9	85.3
ELR [27]	ICLR23	✓	✗	97.1	89.7	82.7	62.0	96.2	97.0	87.6	81.2	93.7	94.1	90.2	58.6	85.8
PLUE [9]	CVPR23	✓	✗	94.4	**91.7**	89.0	70.5	96.6	94.9	92.2	88.8	92.9	95.3	91.4	61.6	88.3
TPDS [22]	IJCV23	✓	✗	97.6	91.5	89.7	83.4	97.5	96.3	92.2	82.4	**96.0**	94.1	90.9	40.4	87.6
DIFO-C-RN	-	✓	✓	97.7	87.6	90.5	**83.6**	96.7	95.8	94.8	74.1	92.4	93.8	92.9	65.5	88.8
DIFO-C-B32	-	✓	✓	97.5	89.0	90.8	83.5	97.8	97.3	93.2	83.5	95.2	**96.8**	**93.7**	**65.9**	**90.3**

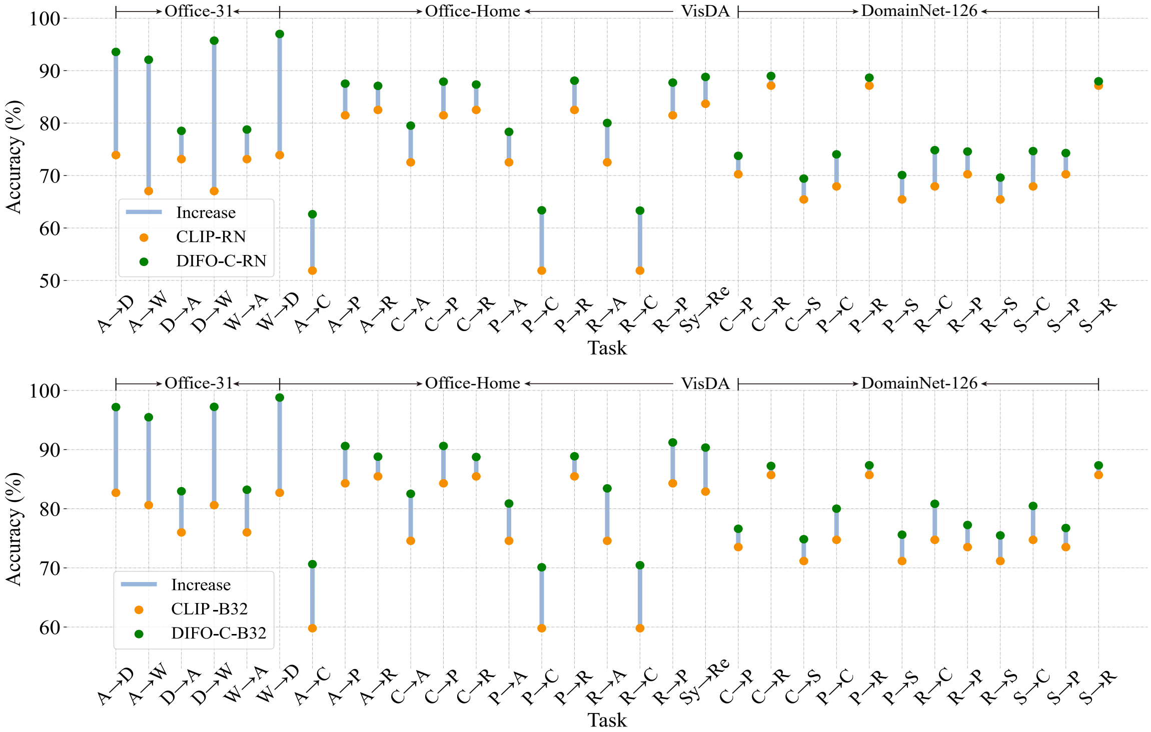

Figure 1. Transfer performance comparison of **DIFO** and CLIP on all tasks of the four evaluation datasets. **Top: DIFO**-C-RN **v.s.** CLIP-RN. **Bottom: DIFO**-C-B32 **v.s.** CLIP-B32.

the two implementations of DIFO in this paper, including DIFO-C-B32 and DIFO-C-RN. Specifically, in DIFO-C-B32, image encoders follow ViT-B/32 architecture proposed in CLIP [14] while DIFO-C-RN uses ResNet [4] as the backbone. The same as the target model mentioned above, ResNet-101 is adopted on **VisDA** and ResNet-50 is used on the rest datasets.

Parameter setting. For the trade-off parameter α and β in the objective L_{PC} (Eq. (6)) and L_{MKA} (Eq. (7)) is set to 1.0 and 0.4 on all datasets, respectively. The parameter of Exponential distribution λ in Eq. (4) is specified to 10.0. The temperature parameters in Eq. (5) are $\tau = 0.1$. The number of the most-likely categories is set to $N = 2$.

Training setting. We adopt the batch size of 64, SGD optimizer with momentum 0.9 and 15 training epochs on all datasets. The prompt template for initiation is the mostly used *'a photo of a [CLASS].'* [14] where [CLASS] stands for the class name. All experiments are conducted with PyTorch

Table 2. Full results (%) of Partial-set SFDA and Open-set SFDA on **Office-Home**.

Partial-set SFDA	Venue	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Source	–	45.2	70.4	81.0	56.2	60.8	66.2	60.9	40.1	76.2	70.8	48.5	77.3	62.8
SHOT [8]	ICML20	64.8	85.2	**92.7**	76.3	77.6	88.8	79.7	64.3	89.5	80.6	66.4	85.8	79.3
HCL [5]	NIPS21	65.6	85.2	**92.7**	77.3	76.2	87.2	78.2	66.0	89.1	81.5	68.4	87.3	79.6
CoWA [7]	ICML22	69.6	93.2	92.3	78.9	81.3	92.1	79.8	71.7	90.0	83.8	**72.2**	**93.7**	83.2
AaD [26]	NIPS22	67.0	83.5	93.1	80.5	76.0	87.6	78.1	65.6	90.2	83.5	64.3	87.3	79.7
CRS [28]	CVPR23	68.6	85.1	90.9	80.1	79.4	86.3	79.2	66.1	90.5	82.2	69.5	89.3	80.6
DIFO-C-B32	–	**70.2**	**91.7**	91.5	**87.8**	**92.6**	**92.9**	**87.3**	**70.7**	**92.9**	**88.5**	69.6	91.5	**85.6**

Open-set SFDA	Venue	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg.
Source	–	36.3	54.8	69.1	33.8	44.4	49.2	36.8	29.2	56.8	51.4	35.1	62.3	46.6
SHOT [8]	ICML20	64.5	80.4	84.7	63.1	75.4	81.2	65.3	59.3	83.3	69.6	64.6	82.3	72.8
HCL [5]	NIPS21	64.0	78.6	82.4	64.5	73.1	80.1	64.8	59.8	75.3	**78.1**	**69.3**	81.5	72.6
CoWA [7]	ICML22	63.3	79.2	85.4	67.6	**83.6**	82.0	66.9	56.9	81.1	68.5	57.9	**85.9**	73.2
AaD [26]	NIPS22	63.7	77.3	80.4	66.0	72.6	77.6	69.1	**62.5**	79.8	71.8	62.3	78.6	71.8
CRS [28]	CVPR23	**65.2**	76.6	80.2	66.2	75.3	77.8	70.4	61.8	79.3	71.1	61.1	78.3	73.2
DIFO-C-B32	–	64.5	**86.2**	**87.9**	**68.2**	79.3	**86.1**	**67.2**	62.1	**88.3**	71.9	65.3	84.4	**75.9**

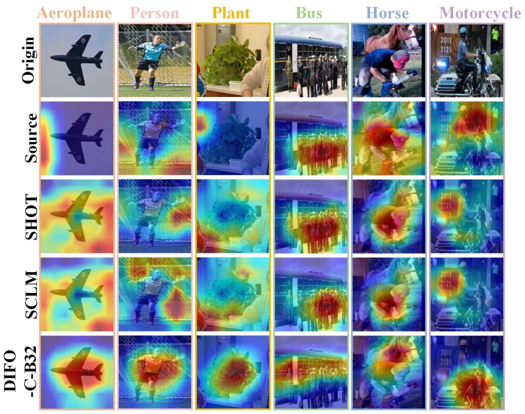

Figure 2. Grad-CAM visualization of **DIFO**-C-B32 and typical comparison methods on toy samples selected from VisDA.

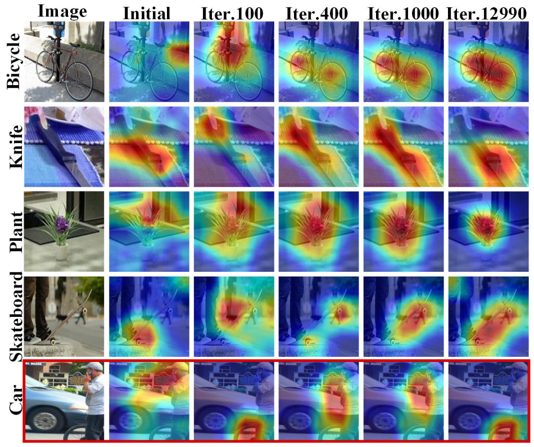

Figure 3. The evolving dynamics of model learning attention based on **DIFO**-C-B32. The red bounding box indicates the failure case.

on a single GPU of NVIDIA RTX.

4. Supplementation of Full Experiment Results

Full results on VisDA. As the supplement of results on VisDA, Tab. 1 presents the full classification details over the 12 categories. It is seen that DIFO-C-RN and DIFO-C-B32 obtain the best results in 7/12 categories compared with SFDA methods. Meanwhile, DIFO-C-RN and DIFO-C-B32 are on top of the second best UDA results in 8/12 categories. Also, we note that the UDA method of ADCLIP beats DIFO-C-RN and DIFO-C-B32 on four transfer tasks. It is understandable that ADCLIP use the labelled source data, whilst our method cannot access the source data. Despite this, DIFO still presents advantages over these source data-required method (see the average accuracy).

Full results of comparison to CLIP. As the supplementation of these domain-grouped results reported in the paper, Fig. 1 gives a comprehensive visualization comparison with CLIP in the perspective of all 31 transfer tasks on the four evaluation datasets. It is seen that the results of DIFO (marked by green circles) are above CLIP (marked by orange circles) on all tasks, whether we use DIFO-C-RN or DIFO-C-B32.

Full results of Partial-set and Open-set SFDA. As the supplementation of these average results in Tab. 5, Tab. 2 gives the full classification accuracy over 12 transfer tasks in the **Office-Home** dataset. As the top in Tab. 2, DIFO-C-B32 obtains best results on 9/12 tasks in the Partial-set SFDA and on the half tasks in the Open-set SFDA.

5. Expanded Model Analysis

Grad-CAM visualization. In Fig. 2, we present the Grad-CAM visualization [18] comparison with the source model and two typical SFDA methods, SHOT and SCLM, based

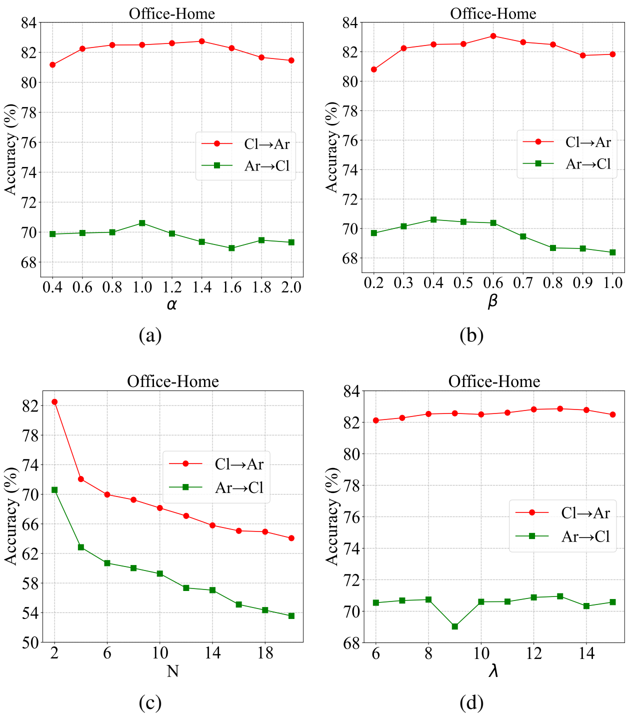

Figure 4. Performance sensitivity of the hyper-parameters. From (a) to (d), the four sub-figures present the accuracy changing as the parameters α, β, N and λ varying, respectively.

on self-supervised learning without ViL model help. For the single object-contained images (see 1∼3 column), DIFO-C-B32's attention focuses on the target object, whilst other methods cover the entire image. Regarding the multi-object-contained images (see 4∼6 column), DIFO-C-B32's attention is more consistent with the target semantics given by the real labels than other methods focusing on the wrong object. These results explain the effectiveness of DIFO-C-B32 integrating the domain generality of the ViL model and the task specificity of the source model.

Attention-based evolving dynamics. To better understand the working of DIFO, this part visualizes the evolving dynamics of model learning attention during the training phase. For a clear view, we display the Grad-CAM visualization results at some typical iterations, as shown in Fig. 3. Among the rightly classified images (the top four rows), the attention smoothly concentrates to the discriminative visual patch. In contrast, the attention of the misclassified image (the last row) converges to the meaningless one.

Sensitivity of hyper-parameter. In the DIFO method, α, β are trade-off parameters in objective L_{PC} (see Eq. (6)) and L_{MKA} (see Eq. (7)). λ is the parameter of Exponential distribution in Eq. (4), whilst N is the number of the most-likely categories. This part discusses their performance sensitivity based on the symmetric transfer tasks Cl→Ar and Ar→Cl in the **Office-Home** dataset. As depicted in Fig. 4 (a), (b) and (d), when these parameters changes, there are no evident drops in the accuracy variation curves. This indicates that DIFO is insensitive to parameters α, β and λ. As for N, the accuracy gradually decreases as N increases. This phenomenon is consistent with our expectation that small N is better and a large value will introduce the semantic noise.

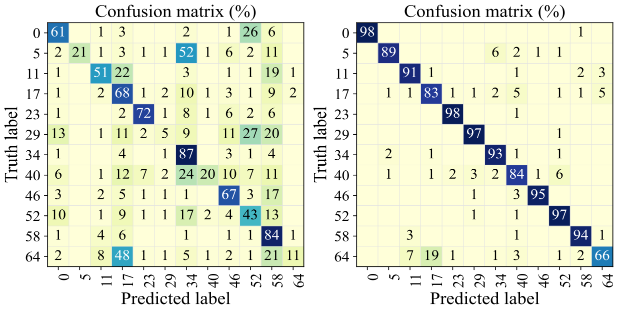

Figure 5. The confusion matrix for 12-way classification on **VisDA**. **Left:** Source model result, **Right:** DIFO-C-B32 result.

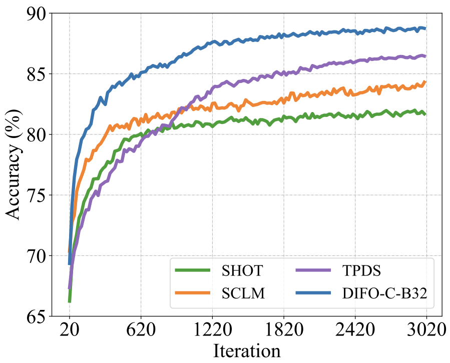

Figure 6. Classification accuracy varying curve comparison on **VisDA** during the adaptation phase.

Confusion matrix. To present a quantitative observation on the category, this part gives the confusion matrix based on the classification results on the **VisDA** dataset. For comparison, we show the confusion matrix of the source model at the left side of Fig. 5. In the no-adaptation case, the misclassified data scatter over the matrix. After adaptation, the misclassified data are evidently corrected by DIFO-C-B32 at the right side of Fig. 5. It is seen that DIFO-C-B32 improves performance on all categories, and on some categories achieving significant growth. For instance, in the second category, the performance promotes by **68**% (from **21**% to **89**%).

Training stability. Training stability is a vital characteristic of supervised learning methods. Based on the large-size dataset **VisDA**, we present the adaptation details of DIFO-

C-B32 using the accuracy varying curves on the target domain. For comparison, the curves of typical self-supervised methods, SHOT, SCLM and TPDS, are also depicted. As shown in Fig. 6, the accuracy gradually increases to the maximum. This result confirms the training stability of DIFO-C-B32. Also, DIFO-C-B32 converges much faster than SHOT, SCLM and TPDS. It indicates that introducing task-specific knowledge from the ViL model is helpful in boosting the source model adaptation.

References

[1] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *CVPR*, 2022. 2

[2] Shinto Eguchi and John Copas. Interpreting kullback–leibler divergence with the neyman–pearson lemma. *Journal of Multivariate Analysis*, 97(9):2034–2040, 2006. 1

[3] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 2

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

[5] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *NeurIPS*, 2021. 3

[6] Zhengfeng Lai, Noranart Vesdapunt, Ning Zhou, Jun Wu, Cong Phuoc Huynh, Xuelu Li, Kah Kuen Fu, and Chen-Nee Chuah. Padclip: Pseudo-labeling with adaptive debiasing in clip for unsupervised domain adaptation. In *ICCV*, 2023. 2

[7] Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *ICML*, 2022. 2, 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] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *CVPR*, 2023. 2

[10] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. 1

[11] R. Müller, S. Kornblith, and G. E Hinton. When does label smoothing help? In *NeurIPS*, 2019. 1

[12] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv:1710.06924*, 2017. 1

[13] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 1

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[15] Subhankar Roy, Martin Trapp, Andrea Pilzer, Juho Kannala, Nicu Sebe, Elisa Ricci, and Arno Solin. Uncertainty-guided source-free domain adaptation. In *ECCV*, 2022. 1

[16] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 1

[17] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019. 1

[18] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 3

[19] Mainak Singha, Harsh Pal, Ankit Jha, and Biplab Banerjee. Ad-clip: Adapting domains in prompt space using clip. In *ICCV Workshop*, 2023. 2

[20] S Tang, Yuji Shi, Zhiyuan Ma, Jian Li, Jianzhi Lyu, Qingdu Li, and Jianwei Zhang. Model adaptation through hypothesis transfer with gradual knowledge distillation. In *IROS*, 2021. 2

[21] Song Tang, Yan Zou, Zihao Song, Jianzhi Lyu, Lijuan Chen, Mao Ye, Shouming Zhong, and Jianwei Zhang. Semantic consistency learning on manifold for source data-free unsupervised domain adaptation. *Neural Networks*, 152, 2022. 1, 2

[22] Song Tang, An Chang, Fabian Zhang, Xiatian Zhu, Mao Ye, and Changshui Zhang. Source-free domain adaptation via target prediction distribution searching. *International Journal of Computer Vision*, pages 1–19, 2023. 2

[23] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 1

[24] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *CVPR*, 2019. 1

[25] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *NeurIPS*, 2021. 1, 2

[26] Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *NeurIPS*, 2022. 2, 3

[27] Li Yi, Gezheng Xu, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Charles Ling, A Ian McLeod, and Boyu Wang. When source-free domain adaptation meets learning with noisy labels. In *ICLR*, 2023. 2

[28] Yixin Zhang, Zilei Wang, and Weinan He. Class relationship embedded learning for source-free unsupervised domain adaptation. In *CVPR*, 2023. 3