Domain-Specificity Inducing Transformers for Source-Free Domain Adaptation

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

Abstract

Conventional Domain Adaptation (DA) methods aim to learn domain-invariant feature representations to improve the target adaptation performance. However, we motivate that domain-specificity is equally important since in-domain trained models hold crucial domain-specific properties that are beneficial for adaptation. Hence, we propose to build a framework that supports disentanglement and learning of domain-specific factors and task-specific factors in a unified model. Motivated by the success of vision transformers in several multi-modal vision problems, we find that queries could be leveraged to extract the domain-specific factors. Hence, we propose a novel Domain-Specificity inducing Transformer (DSiT) framework [1] for disentangling and learning both domain-specific and task-specific factors. To achieve disentanglement, we propose to construct novel Domain-Representative Inputs (DRI) with domain-specific information to train a domain classifier with a novel domain token. We are the first to utilize vision transformers for domain adaptation in a privacy-oriented source-free setting, and our approach achieves state-of-the-art performance on single-source, multi-source, and multi-target benchmarks.

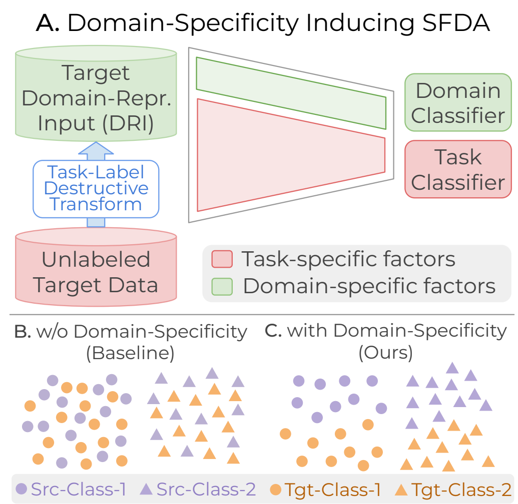

Figure 1. **A.** We induce domain-specificity by disentangling domain- and task-specific factors within the model. A task-label-destructive transform produces novel Domain-Representative Inputs (DRI) to learn domain-specific factors via domain classification. **B.** Conventional DA methods preserve domain-invariance, resulting in only task-oriented clusters in the feature space. **C.** Our proposed disentanglement ensures that different domains are well-clustered.

1. Introduction

Machine learning models often fail to generalize to unseen domains due to the discrepancy between training (source) and test (target) data distributions (*i.e. domain shift*). This results in poor deployment performance and is also critical for applications like autonomous driving [9] or medical imaging [12]. Unsupervised domain adaptation (DA) techniques seek to address this challenging scenario by transferring task-specific knowledge from a labeled source domain to an unlabeled target domain. However, DA works [15] require concurrent access to source and target data. Such a constraint is highly impractical since data tends to be proprietary and cannot be easily shared. Hence, we focus on Source-Free DA [31] that operates under a practical setting of sharing only the source model between a *vendor* and a *client*, without any data-sharing.

Conventional Domain Adaptation works [15] aim to learn task-discriminative features that are domain-invariant, *i.e.* indistinguishable w.r.t. domain shift. Intuitively, if the feature distributions of the source and target domain are similar, then a low error on the source domain translates to the target, shown theoretically by [3]. But domain-invariance does not always result in optimal target performance because a specific domain might be far away from the support of a domain-invariant model [14]. Further, supervised in-domain trained models (where train and test datasets come from the same domain) usually perform better as they hold useful domain-specific properties. Thus, we motivate the concept of *domain-specificity to improve the target adaptation performance*.

In source-free DA, while adapting a model to a new

*Equal Contribution
[1]Project Page: http://val.cds.iisc.ac.in/DSiT-SFDA/

target domain, the major problem is to preserve the task knowledge from the source domain. Prior works address this by aligning the target model feature space with that of the source [37]. However, we argue that it is equally important for a model to learn domain-specific information while preserving task-specific knowledge. Hence, in our work, we seek a solution to an important question, *"How do we develop a framework that enables us to improve domain-specificity while also retaining task-specificity?"*.

Thus, we seek a framework that supports the disentanglement of task-specific factors and domain-specific factors, thereby allowing us better control over them. A well-disentangled framework would allow the learning of both domain-specific and task-specific factors in the model simultaneously, as shown in Fig. 1A. This not only yields better performance but also ensures that the different domains are well-clustered in the feature space of the model (Fig. 1C) compared to the baseline (Fig. 1B). However, the key question remains, *"How do we devise a method for the disentanglement and learning of domain-specific and task-specific factors?"*

Recently, transformers have demonstrated remarkable performance in several vision tasks [26, 39]. They contain a multi-head self-attention mechanism that attends to all image patches and provides a global context. Motivated by this fundamental difference of a global context, we explore the possibility of a disentanglement framework with transformers as domain information is inherently a global, higher-order statistic [8] that may not be adequately captured in CNNs. Inspired by the multi-modal works [21] that utilize queries to extract domain-specific information from a particular modality, we propose to enable the disentanglement of domain-specificity through the query weights as part of our novel *Domain-Specificity inducing Transformer* (DSiT) framework.

Concretely, we induce domain-specificity by updating only the query weights via domain classifier training (Fig. 1A). The remaining weights are updated via task classifier training. To further inculcate the disentanglement, we train the domain-classifier with novel *Domain-Representative Inputs* (DRI) where a task-label-destructive transform removes the task-specific information. We also propose a novel *domain-specificity disentanglement criterion* to evaluate the disentanglement of domain-specific and task-specific factors and demonstrate the disentanglement of our proposed DSiT.

We outline the contributions of our work as follows:

- We investigate and provide insights on how domain-specificity can be leveraged to improve DA. To this end, we propose a novel, unified Domain-Specificity inducing Transformer (DSiT) to disentangle and learn task-specific and domain-specific factors.
- We utilize query weights to enable the disentanglement in DSiT with a novel training algorithm that well supports our insights. We also introduce novel Domain-Representative Inputs (DRI) to further enhance the disentanglement.
- We define a novel domain-specificity disentanglement criterion to determine if the domain-specific and task-specific factors are well-disentangled.
- We achieve state-of-the-art performance on source-free benchmarks across single-source, multi-source, and multi-target DA while also introducing the first source-free DA benchmarks for transformers.

2. Related Works

Domain Adaptation. Prior DA works minimize the domain gap via adversarial distribution matching [18, 20] or minimize statistical distances [42, 47, 48, 64]. However, SFDA works [27, 28, 29, 37, 38] use information maximization and pseudo-labelling to match the target features with the source features. [34, 62] perform neighbourhood clustering and regularization in the target domain. Our work also considers a SFDA setup [30, 31] that focuses on a practical vendor-client setting where vendor and client may use cooperative/same learning strategies, without data sharing.

Transformer-based DA. Vision transformers are self-attention based architectures with state-of-the-art performance on several vision tasks like object recognition [5], semantic segmentation [19], *etc*. The application of transformers in DA scenarios is relatively less explored. CDTrans [59] proposes cross-attention between source and target image pairs for domain alignment. SSRT [50] uses a self-training mechanism while [61] does adversarial training. TransDA [60] incorporates a self-attention layer on top of a ResNet backbone for SFDA, and hence is not a pure Vision Transformer (ViT) based solution. Unlike prior SFDA works, we propose a fully ViT based solution for the first time, with a focus on enhancing domain-specificity.

Domain-specificity for DA. Prior works such as [32, 33] focus on style and content disentanglement and aim to achieve domain invariance by removing the style (domain-specific) information from images while preserving only the content information. However, we argue that domain invariance doesn't guarantee optimal performance. We draw motivation from domain generalization works [14] that propose constructing domain prototypes from unlabeled target samples to learn domain-specific features. Similarly, DiDA [4] proposes feature disentanglement into common and domain-specific features to improve adaptation performance. Furthermore, DMG [7] proposed to balance specificity and invariance via learning domain-specific masks and [6] proposed to learn domain-specific batch-normalization parameters. Inspired by these, we propose to leverage domain-specificity in transformers to improve source-free DA performance.

3. Approach

Problem setup. For closed-set DA, we consider a labeled source dataset $\mathcal{D}_s = \{(x_s, y_s) : x_s \in \mathcal{X}, y_s \in \mathcal{C}_g\}$ where $\mathcal{X}$ denotes the input space and $\mathcal{C}_g$ denotes the goal task label set. We denote the unlabeled target dataset by $\mathcal{D}_t = \{x_t : x_t \in \mathcal{X}\}$. The task is to predict the label for each target sample x_t from the label set $\mathcal{C}_g$. Following [59], we use ViT-B [11] as the backbone feature extractor $h : \mathcal{X} \rightarrow \{\mathcal{Z}_c, \mathcal{Z}_1, \mathcal{Z}_2, ...\mathcal{Z}_{N_P}\}$ where $\mathcal{Z}_c$ represents the class token feature-space and $\mathcal{Z}_1, \mathcal{Z}_2, ...\mathcal{Z}_{N_P}$ are patch token feature-spaces (N_P is the number of patches). A goal task classifier trained on the class token is denoted as $f_g : \mathcal{Z}_c \rightarrow \mathcal{C}_g$. In the source-free vendor-client setup [31], we operate under the practical constraint where data sharing between vendor and client is prohibited. The vendor trains a model on the source domain and shares only the model with the client. The client uses the vendor-side model to adapt to the target domain.

Unlike usual DA works [15] that advocate domain-invariant learning, we investigate how domain-specificity can be leveraged for DA. First, we discuss why domain-specificity is useful through the following insight.

Insight 1. (Domain-specificity leads to improved DA) *In the source-free DA setting, the aim is to achieve good target accuracy for the client-side model. Since an in-domain trained model better represents domain-specific factors as compared to a domain-invariant model, inculcating domain specificity on the client-side plays a major role in improving the target adaptation performance.*

Remarks. In a vendor-client setting, a client wishes to improve the task performance of the model on the target data. Conventional DA methods [15] often devise a strategy to learn domain-invariant features which contain only the task knowledge, which could be used to generalize across multiple domains. However, the optimal task-specific features from a domain-specific model might be far away from the task-specific features of a domain-invariant model [14]. Hence, such approaches are unsuitable for SFDA, where the goal of improved target performance can be better achieved if we inculcate in-domain knowledge to build a domain-specific model. Therefore, we note that it is equally important for a client-side model to learn "domain-specific factors" containing the crucial in-domain characteristic knowledge along with the "task-specific factors" holding the goal task information.

As Insight 1 motivates domain-specificity, a natural question arises, "*How can we enable and control domain-specificity while also learning task-specific factors?*"

Insight 2. (Disentanglement of domain-specific and task-specific factors to control domain specificity) *Disentanglement of domain-specific and task-specific factors provides a way to learn the two orthogonal factors together within the same model, enabling better control over them.*

Remarks. Since source and target domains can be far apart, *i.e.* high variance in domain-specific factors, it is desirable for a domain-specific model to learn a *disentangled* feature space which can adequately capture task-specific as well as domain-specific factors. Prior SFDA works either aim to align the target domain features towards the source domain [37] or update all parameters, risking the loss of source-side task-specific knowledge [62]. Conversely, our proposed disentanglement allows simultaneous and separate learning of the domain-specific and task-specific factors. Moreover, these domain-specific factors could be leveraged to orient the task-specific factors [14] for a new target domain.

Next, we seek an answer to the following question, *"How do we devise a unified architecture to enable disentanglement of domain-specific and task-specific factors?"*

3.1. Exploring the potential of transformers towards domain specificity

We explore the various possibilities of disentangling domain-specific and task-specific factors in the self-attention module of a transformer. Prior vision transformer works have demonstrated superior feature alignment capabilities within the attention module of a transformer, even across multi-modal scenarios such as text-to-vision [21, 53]. Also, transformers are inherently more robust to noise [59] and capture long-range dependencies via self-attention across different patches compared to the localized context in CNNs. Hence, we propose to adopt transformer architectures for domain-specificity and are the first to use them for source-free DA.

a) Query for domain-specificity. Vaswani et al. [54] introduced self-attention as a compatibility function between a Query and a corresponding Key. The final output is computed as a weighted sum of the Value where the weights are assigned as per the self-attention. Prior multi-modal works [21, 36] use modality-specific queries (e.g. from text or image modality) as input to the cross-attention. While we draw inspiration from these works, our novel approach explicitly enforces domain-specificity in the "self-attention query" (*i.e.* inside every transformer layer), different from the cross-attention approach of multi-modal works. This helps to better control the task-specific factors, where a domain-specific Query learns to acquire the domain-specific information from the input Value, making the overall output domain-specific in nature. While there could be more sophisticated ways to inculcate domain-specificity, like via hypernetworks [16] or auxiliary models [23], we propose this simple strategy that validates our insights.

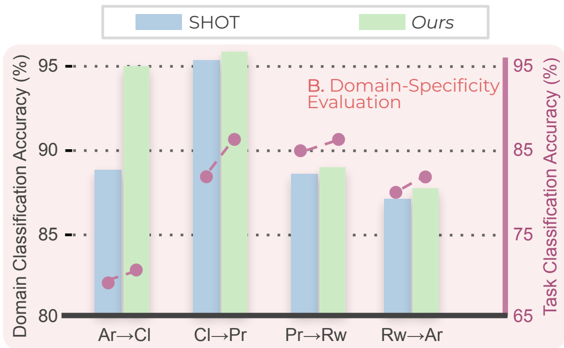

Figure 2. We evaluate the domain-specificity of our adapted model w.r.t. SHOT on Office-Home. Task acc. is goal task accuracy and Domain acc. is the accuracy of the binary source-target domain classifier (as explained in Sec. 3.1b). Our higher task accuracy (in pink) indicates better task-specificity and better disentanglement.

b) Evaluating domain-specificity and task-specificity. To analyze the effectiveness of our proposed strategy, we examine the domain-specificity of the ViT class token by training a linear domain classifier with the source and target domain samples. The accuracy is computed on four settings A2C, C2P, P2R and R2A on Office-Home dataset. From Fig. 2, we observe that our approach achieves a higher domain classification accuracy across all four settings. Note that we access both source and target domain data only for this analysis. Intuitively, we examine the domain-specificity inculcated in the class-tokens as they are used by the task classifier. We also report the task performance to highlight the task-specific knowledge of the model in Fig. 2. We observe that our approach achieves a higher task classification accuracy as well. This implies that our proposed approach captures the domain-specific and task-specific factors well, leading to improved adaptation performance.

3.2. Training algorithm

We propose **D**omain-**S**pecificity **i**nducing **T**ransformers for Source-Free DA (**DSiT**). In the following sections, we describe the proposed approach to train DSiT.

3.2.1 Vendor-side source training

We train DSiT in two steps. First, we perform domain-specificity disentanglement via domain classifier training. The second step involves goal task classifier training. Next, we delve into the details of each step.

a) Domain-specificity disentanglement (Fig. 3B) We first induce domain-specificity by training a domain classifier f_d with a novel domain token $z_d \in \mathcal{Z}_d$ from the backbone h. For this, the vendor first prepares augmented datasets $\mathcal{D}_s^{(i)} = \{(x_s^{[i]}, y_s, y_d) \ \forall \ (x_s, y_s) \in \mathcal{D}_s\} \ \forall \ i \in [N_a]$ by augmenting each source sample x_s (N_a is the number of augmentations). Here, the augmentation $\mathcal{A}_i : \mathcal{X} \to \mathcal{X}$ is applied to get $x_s^{[i]} = \mathcal{A}_i(x_s)$. Each input is assigned a domain label $y_d = i$ where i denotes the augmentation label. We use five label-preserving augmentations that simulate novel domains [30] (see Suppl. for more details).

We motivate the use of augmentations with two supports. First, as we operate in a source-free setting, access to multiple domains cannot be assumed, and these augmented domains can be used to inculcate domain-specificity. Further, this domain-specificity inculcation can be shared between vendor and client by sharing only the augmentation information. Second, as shown in Fig. 4, an augmented source domain may be closer to another augmented target domain (blue lines) w.r.t. the original source-target (red line). We also confirm this numerically (reported in Suppl). Thus, we have access to more diverse domains (with lower domain gaps) which need to be well-separated by the domain classifier f_d, allowing for better domain-specificity.

While we could train f_d with only augmented samples to inculcate domain-specificity, we also need to disentangle the task-specific information as discussed under Insight 2. Thus, we next describe a novel input representation that better represents domain-specific factors by separating the task-specific factors, *i.e.* enabling better disentanglement.

Domain-Representative Input (DRI) is constructed at the input-level to preserve only domain-specific information by applying a task-label-destructive transformation of patch-shuffling [40] as shown in Fig. 3A. Intuitively, domain information is a higher-order statistic [8] that is preserved after patch-shuffling while class information is lost. The shuffling of patches is extremely vital for amplifying the input domain specific factors for the domain-specificity training step. Hence, DRIs are not just used as an augmentation for the goal task training, as shown in prior transformer works [45], but are a means to inculcate domain specificity. We empirically demonstrate in Sec. 4.2b that DRI guides the goal task performance better than only augmented inputs, because of improved disentanglement. We also provide additional baselines in Table 5 to validate the unsuitability of DRI as a general augmentation for the goal task training.

As per our analysis in Sec. 3.1, we train only the transformer query weights, to exclusively hold domain-specific knowledge, with the domain classification loss,

$$\min_{\theta_Q, \theta_{f_d}} \mathop{\mathbb{E}}_{(x, y_d) \in \cup_i \mathcal{D}_s^{(i)}} [\mathcal{L}_{dom}] \text{ where } \mathcal{L}_{dom} = \mathcal{L}_{ce}(f_d(z_d), y_d) \tag{1}$$

where z_d is the domain-token output from backbone h and $\theta_Q = \cup_j W_{Q_j}$; j is an index over the backbone layers. Note that DRI are used for $\mathcal{L}_{dom}$. For simplicity, we re-use $\mathcal{D}_s^{(i)}$ to include the task-label-destructive transform (Fig. 3A).

b) Domain-specific goal task training (Fig. 3C) After one round of domain classifier training, we train DSiT for the goal task. Here, we update key and value weights W_K and W_V along with all other parameters (except query weights W_Q). Hence, the frozen queries are disentangled

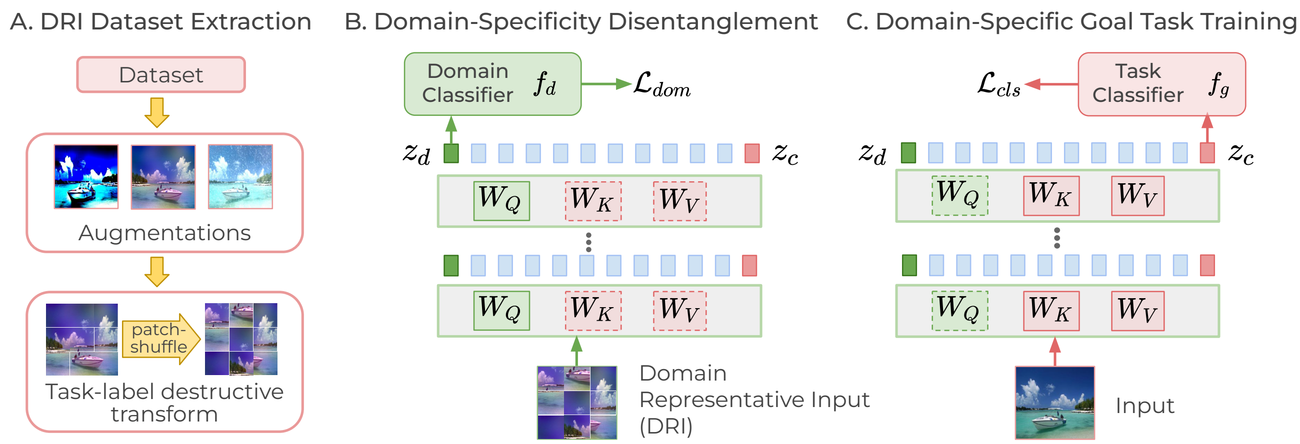

Figure 3. **DSiT Training:** ***A. DRI dataset extraction:*** DRI are obtained via patch-shuffling of augmented inputs. ***B. Domain-Specificity Disentanglement:*** Domain classifier f_d is trained with domain classification loss $\mathcal{L}_{dom}$ using DRI, updating only query weights W_Q via a domain token z_d, while other weights including W_K and W_V are frozen (dotted boundary). ***C. Domain-Specific Goal Task Training:*** The task classifier is trained with task classification loss $\mathcal{L}_{cls}$ updating all weights except W_Q.

and hold crucial domain-specific information. The vendor trains a source model consisting of the backbone h and task classifier f_g using the source dataset $\mathcal{D}_s$ with the task-classification loss as follows,

$$\min_{\theta_h \setminus \theta_Q, \theta_{f_g}} \mathop{\mathbb{E}}_{(x_s, y_s) \in \mathcal{D}_s} [\mathcal{L}_{cls}] \text{ where } \mathcal{L}_{cls} = \mathcal{L}_{ce}(f_g(z_c), y_c) \tag{2}$$

where z_c is the class-token output from backbone h, and $\theta_h \setminus \theta_Q$ are the parameters of the backbone h excluding all W_Q weights while θ_{f_g} are the parameters of classifier f_g. The two steps of domain-specificity disentanglement (Eq. 1) and goal task training (Eq. 2) are performed in tandem, one after the other. See Suppl. for details.

3.2.2 Client-side target adaptation

The vendor shares the source-trained, disentangled DSiT model with the client for target adaptation. The client also follows the same process of domain-specificity disentanglement as the vendor. To disentangle the target-domain-specific factors and task-specific factors for improved target adaptation (as per Insight 1), the client applies the same augmentations to the target data to generate DRIs for domain classifier training with the domain classification loss $\mathcal{L}_{dom}$ (as described in Sec. 3.2.1a and Fig. 3B). For goal task training, the client uses the standard information maximization and diversity losses [37, 62, 63]. The overall client-side objective is as follows:

$$\min_{\theta_h \setminus \theta_Q, \theta_{f_g}} \mathop{\mathbb{E}}_{\mathcal{D}_t} [\mathcal{L}_{im} + \mathcal{L}_{div}] + \min_{\theta_Q, \theta_{f_d}} \mathop{\mathbb{E}}_{\cup_i \mathcal{D}_t^{(i)}} [\mathcal{L}_{dom}] \tag{3}$$

We detail $\mathcal{L}_{im}$ and $\mathcal{L}_{div}$ in the Suppl. as these are commonly used and not our main contributions. Note that only original target data $\mathcal{D}_t$ is used for $\mathcal{L}_{im}$ and $\mathcal{L}_{div}$. As in Eq. 1, DRI are used for $\mathcal{L}_{dom}$ here. For simplicity, we re-use $\mathcal{D}_t^{(i)}$ to include the task-label-destructive transform (Fig. 3A).

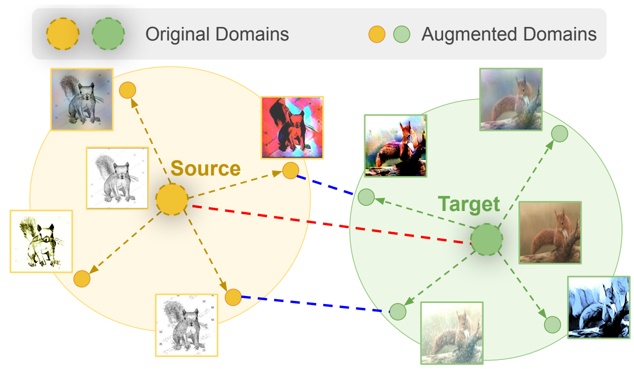

Figure 4. Novel augmented domains can be simulated where the domain gap between an augmented source and an augmented target (blue lines) can be less than the original domain gap (red line). With a domain-classifier for augmented domains, we induce better domain-specificity as multiple domains with lower domain gaps are well-separated.

3.3. Domain-specificity disentanglement criterion

To evaluate the disentanglement of domain-specific and task-specific factors, we propose a criterion considering the following three feature-space cosine similarity metrics. Let γ_{cls} be the intra-class, inter-domain similarity, γ_{dom} be the intra-domain, inter-class similarity, and γ_{all} be the inter-class, inter-domain similarity. These are computed as the similarity between pairs of features averaged over the possible input pairs for each input. For example, possible input pairs for γ_{cls} have to come from the same class but different domains. See Suppl. for more details. Based on these, we define a criterion to determine if the domain-specific and task-specific factors of the model are well-disentangled.

Table 1. Single-Source Domain Adaptation (SSDA) on Office-Home benchmark. SF indicates *source-free* adaptation. ResNet-based methods (top) and Transformer-based methods (bottom). * indicates results taken from CDTrans [59]. **Bold** numbers indicate the best results among SFDA methods. **Green** results indicate our method's improvements over the corresponding methods.

Method	SF	Office-Home												
		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
ResNet-50 [17]	✗	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
SENTRY[44]	✗	61.8	77.4	80.1	66.3	71.6	74.7	66.8	63.0	80.9	74.0	66.3	84.1	72.2
SCDA [35]	✗	60.7	76.4	82.8	69.8	77.5	78.4	68.9	59.0	82.7	74.9	61.8	84.5	73.1
A^{2}Net [58]	✓	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
GSFDA [63]	✓	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
CPGA [46]	✓	59.3	78.1	79.8	65.4	75.5	76.4	65.7	58.0	81.0	72.0	64.4	83.3	71.6
NRC [62]	✓	57.7	80.3	82.0	68.1	79.8	78.6	65.3	56.4	83.0	71.0	58.6	85.6	72.2
SHOT [37]	✓	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++ [38]	✓	57.9	79.7	82.5	68.5	79.6	79.3	68.5	57.0	83.0	73.7	60.7	84.9	73.0
TVT [61]	✗	74.8	86.8	89.4	82.7	87.9	88.2	79.8	71.9	90.1	85.4	74.6	90.5	83.5
SSRT-B [50]	✗	75.1	88.9	91.0	85.1	88.2	89.9	85.0	74.2	91.2	85.7	78.5	91.7	85.4
CDTrans [59]	✗	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
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 **(+2.4)**
DIPE [57]	✓	66.0	80.6	85.6	77.1	83.5	83.4	75.3	63.3	85.1	81.6	67.7	89.6	78.2 **(+2.3)**
Mixup [29]	✓	65.3	82.1	86.5	77.3	81.7	82.4	77.1	65.7	84.6	81.2	**70.1**	88.3	78.5 **(+2.0)**
DSiT-B (*Ours*)	✓	**69.2**	**83.5**	**87.3**	**80.7**	**86.1**	**86.2**	**77.9**	**67.9**	**86.6**	**82.4**	68.3	**89.8**	**80.5**

Definition 1. (Domain-Specificity Disentanglement Criterion) *The domain-specific and task-specific factors of a model are well-disentangled if*

$$\gamma_{cls}, \gamma_{dom} > \gamma_{all} \text{ and } |\gamma_{cls} - \gamma_{dom}| < \tau \tag{4}$$

Remarks. Here, τ is a threshold. In other words, γ_{cls} and γ_{dom} should be higher than γ_{all} and the absolute difference between γ_{cls} and γ_{dom} should be low. Based on the definitions, γ_{cls} indicates task-specificity since the used samples belong to the same class but different domains. Similarly, γ_{dom} represents the domain-specificity. γ_{all} is the similarity between samples from different classes and different domains, which should be lower than γ_{dom} and γ_{cls} with any domain-specific and task-specific knowledge, respectively. Along with the first condition, if the difference between task-specificity and domain-specificity is low, *i.e.* both are equally present, the two will be well-disentangled in the model. In Sec. 4.2c, we empirically verify that our proposed approach satisfies Definition 1.

4. Experiments

We evaluate the effectiveness of our proposed approach on several benchmarks.

Datasets. We demonstrate the efficacy of our work on the following three standard object recognition DA benchmarks. **Office-31** [47] is a benchmark consisting of three domains under office environments - Amazon (**A**), DSLR (**D**), and Webcam (**W**) with 31 classes each. **Office-Home** [56] consists of images of everyday objects divided into four domains – Artistic (**Ar**), ClipArt (**Cl**), Product (**Pr**), and Real-world (**Rw**), each with 65 classes. **VisDA** [43] is a large-scale dataset for adapting algorithms from synthetic domains to real domains. There are 152,397 synthetic images in the source domain and 55,388 real-world images in the target domain. **DomainNet** [42] is the most challenging benchmark with six domains, each with 345 classes: ClipArt (**C**), Real (**R**), Infograph (**I**), Painting (**P**), Sketch (**S**), and Quickdraw (**Q**).

Implementation details. We follow the experimental settings of CDTrans [59] and use DeiT-Base [52] as the backbone ViT for fair comparisons. The DeiT-Base architecture consists of 12 layers, where each layer consists of 12 self-attention heads collectively termed as multi-head self-attention. We use an input size of 224×224 and a patch size of 16×16 for all our experiments, resulting in 14×14 patches (196 total) per input. DeiT-B contains an additional distillation token, however the rest of the architecture is the same as a ViT-B backbone. For optimizing the training objectives, we use Stochastic Gradient Descent (SGD) with momentum 0.9, and weight decay ratio of 1×10^{-4}. Refer to the Suppl. for the complete implementation details.

4.1. Comparisons with prior works

We compare with SOTA methods on single-source, and multi-source DA in Table 1, 2, 3 and 4 and multi-target DA (see Suppl. for details).

a) Single-source domain adaptation (SSDA). Table 1 2, and 3 present comparisons between our proposed approach DSiT and prior SSDA works. For the Office-Home benchmark, our approach achieves state-of-the-art performance

Table 2. Single-Source Domain Adaptation (SSDA) on Office-31 and VisDA benchmarks. SF indicates *source-free* adaptation. ResNet-based methods (top) and Transformer-based methods (bottom). * indicates results taken from CDTrans [59]. **Bold** numbers indicate the best results among SFDA methods. **Green** results indicate our method's improvements over the corresponding methods.

Method	SF	Office-31							VisDA
		A→W	D→W	W→D	A→D	D→A	W→A	Avg.	S→R
ResNet-50 [17]	✗	68.9	68.4	62.5	96.7	60.7	99.3	76.1	52.4
CAN [25]	✗	94.5	99.1	99.8	95.0	78.0	77.0	90.6	87.2
FixBi [41]	✗	96.1	99.3	100.0	95.0	78.7	79.4	91.4	87.2
CDAN+RADA [24]	✗	96.2	99.3	100.0	96.1	77.5	77.4	91.1	76.3
CPGA [46]	✓	94.1	98.4	99.8	94.4	76.0	76.6	89.9	84.1
HCL [22]	✓	91.3	98.2	100.0	90.8	72.7	72.7	87.6	83.5
VDM-DA [51]	✓	94.1	98.0	100.0	93.2	75.8	77.1	89.7	85.1
A^2Net [58]	✓	94.0	99.2	100.0	94.5	76.7	76.1	90.1	84.3
NRC [62]	✓	90.8	99.0	100.0	96.0	75.3	75.0	89.4	85.9
SHOT [37]	✓	90.1	98.4	99.9	94.0	74.7	74.3	88.6	82.9
SHOT++ [38]	✓	90.4	98.7	99.9	94.3	76.2	75.8	89.2	87.3
TVT [61]	✗	96.4	99.4	100.0	96.4	84.9	86.1	93.8	83.9
CGDM-B* [13]	✗	95.3	97.6	99.8	94.6	78.8	81.2	91.2	82.3
CD-Trans [59]	✗	96.7	99.0	100.0	97.0	81.1	81.9	92.6	88.4
SSRT-B [50]	✗	97.7	99.2	100.0	98.6	83.5	82.2	93.5	88.7
DIPE [57]	✓	95.5	98.5	100.0	94.8	77.5	77.1	90.5 **(+2.5)**	82.8 **(+4.8)**
Mixup [29]	✓	96.1	98.6	100.0	95.4	80.2	80.1	91.7 **(+1.3)**	86.3 **(+1.3)**
SHOT-B*	✓	94.3	99.0	100.0	95.3	79.4	80.2	91.4 **(+1.6)**	85.9 **(+1.7)**
DSiT-B (*Ours*)	✓	**97.2**	**99.1**	**100.0**	**98.0**	**81.7**	**81.8**	**93.0**	**87.6**

Table 3. Single-Source Domain Adaptation (SSDA) results on the DomainNet dataset. * indicates results taken from [50]. NSF indicates Non-Source-Free, SF indicates Source-Free and SO indicates Source-Only.

SCDA [65] **(NSF)**	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 [52] **(SO)**	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 [37] **(SF)**	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* [59] **(NSF)**	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* [50] **(NSF)**	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**

DSiT **(*Ours*) (SF)**	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	27.2	51.8	23.1	70.2	54.7	45.4
inf	52.3	-	48.8	12.8	68.3	44.2	45.3
pnt	59.2	26.1	-	14.5	71.5	51.4	44.5
qdr	38.1	8.3	21.2	-	37.2	27.6	26.5
rel	60.4	28.0	57.8	13.1	-	49.7	41.8
skt	66.3	27.5	56.0	24.4	70.2	-	48.9
Avg.	55.3	23.4	47.1	17.6	63.5	45.5	**42.1**

among source-free works (Table 1). DSiT outperforms the source-free prior works SHOT-B* by 2.4%, Mixup by 2% and DIPE by 2.3% on Office-Home and yields comparable performance to the non-source-free transformer-based prior work CDTrans. Similarly, on the Office-31 benchmark (Table 2), our approach improves over the source-free works SHOT-B* by 1.6%, Mixup by 1.3% and DIPE by 2.5% and gives competitive results compared to non-source-free works. On the more challenging VisDA dataset (Table 2), our approach outperforms source-free SHOT-B* baseline by 1.7%. DSiT also outperforms all the existing source-free methods by a significant margin on the most challenging benchmark DomainNet (Table 3), and also achieves a 5.1% improvement over CDTrans, despite the latter being non-source-free. Refer to Suppl. for the complete table.

b) Multi-source domain adaptation (MSDA). In Table 4, we compare our approach with source-free and non-source-free multi-source prior works. We improve over the source-free SHOT-B* by 1% on Office-Home. Despite the source-free constraint, we outperform even non-source-free works. Overall, these benchmark results highlight the efficacy of the proposed disentanglement of domain-specific and task-

Table 4. Multi-Source DA (MSDA) on Office-Home benchmark. SF indicates *source-free* adaptation. ResNet-based methods (top) and Transformer-based methods (bottom).

Method	SF	Office-Home				
		→Ar	→Cl	→Pr	→Rw	Avg.
Source-combine	✗	58.0	57.3	74.2	77.9	66.9
WDA [1]	✗	71.9	61.4	84.1	82.3	74.9
SImpAl [55]	✗	70.8	56.3	80.2	81.5	72.2
CMSDA [49]	✗	71.5	67.7	84.1	82.9	76.6
DECIS [2]	✓	74.5	59.4	84.4	83.6	75.5
SHOT++ [38]	✓	73.1	61.3	84.3	84.0	75.7
CAiDA [10]	✓	75.2	60.5	84.7	84.2	76.2
SHOT [37]	✓	72.2	59.3	82.8	82.9	74.3
SHOT-B*	✓	83.9	71.8	89.7	89.4	83.7
DSiT-B (*Ours*)	✓	**84.4**	**73.8**	**90.7**	**89.7**	**84.7** (+1.0)

Table 5. Ablation study for DRI as a general augmentation on Office-Home SSDA benchmark with the DeiT-B backbone.

Method	Ar→Cl	Cl→Pr	Pr→Rw	Rw→Ar	Avg.
SHOT-B [37]	67.1	83.4	85.3	80.4	79.1
SHOT-B + DRI	65.8	79.9	84.2	79.6	77.4
CDTrans [59]	68.8	87.1	88.2	82.0	81.5
CDTrans + DRI	59.7	81.6	84.2	81.5	76.8

specific factors (Insight 2) across three diverse settings of single-source, multi-source, and multi-target DA.

4.2. Analysis

In Table 6, we provide an ablation study for various steps in the training of our approach.

a) Effect of inculcating domain-specificity. In Table 6, we compare the effect of domain-specificity training (without DRI) on both vendor-side and client-side training. On vendor-side (#1 vs. #2), we observe improvements of 0.8% while the improvements on client-side (#4 vs. #5) are 1.0%. This is in line with our Insight 1 that domain-specificity supports target adaptation performance (even without DRI).

b) Effect of DRI. As discussed in Sec. 3.2, domain-representative inputs (DRI) reduce the impact of task-specific information while preserving and aiding in the learning of domain-specific information. This further amplifies the impact of domain-specificity on the goal task, which results in improvements of 1% on the vendor-side (#2 vs. #3) and 1.1% on the client-side (#5 vs. #6).

c) Empirical analysis of Definition 1. As per the discussion under Definition 1, we report the three similarity metrics for the baseline SHOT and our DSiT in Table 7. The condition of $\gamma_{dom}, \gamma_{cls} > \gamma_{all}$ is satisfied by both models (since γ_{all} denotes similarity between any class and any domain samples which would be lower in most cases). γ_{dom} and γ_{cls} for DSiT are closer in magnitude than SHOT, indicating its better disentanglement. This is further strengthened by the improved DA performance of DSiT w.r.t. SHOT.

Table 6. Ablation study for various stages of training on Office-Home SSDA benchmark with the DeiT-B backbone (*average over 4 settings*). SO: Source-Only, DST: Domain-Specificity Training

Training Phase	Model	#	DST	DRI	Avg.
Vendor-side	SO	1	✗	✗	74.8
	DSiT	2	✓	✗	75.6 (+0.8)
		3	✓	✓	76.6 (+1.8)
Client-side	SHOT	4	✗	✗	79.0
	DSiT	5	✓	✗	80.0 (+1.0)
		6	✓	✓	**81.1** (+2.1)

Table 7. Empirical evaluation of similarity metrics shows that domain-specificity γ_{dom} and task-specificity γ_{cls} are closer for our DSiT along with better DA performance, indicating better disentanglement than the baseline.

Office-Home	γ_{cls}	γ_{dom}	γ_{all}	SSDA
SHOT-B [37]	0.84	0.74	0.73	78.1
DSiT-B (*Ours*)	0.81	0.78	0.71	**80.5**

d) Performance of DRI as a general augmentation. To preserve and enhance domain-specificity, we utilize DRIs only for the domain-classifier training (Eq. 1), which helps to improve the target adaptation performance (as per Insight 1). However, DRIs are unsuitable for task-classifier training as the task label information is destroyed by patch-shuffling. In Table 5, we use DRI-augmented inputs for task classifier training with SHOT-B [37] and CDTrans [59] baselines and observe significant drops of 1.7% and 4.7% on Office-Home (4 settings) with respect to the corresponding baselines, respectively. This drop is expected and validates our insights that DRI is not suitable as a general augmentation for DA, but is extremely crucial for domain-specificity training (+1% improvement using DRI in Table 6).

5. Conclusion

In this work, we study the concept of domain-specificity in source-free DA. We provide insights to analyse how domain-specificity could be leveraged to improve target adaptation performance. We, therefore, propose a novel Domain-Specificity inducing Transformer (DSiT) where we leverage the queries of a vision transformer to induce domain-specificity and train the unified model to enable a disentanglement of task- and domain-specificity. Based on our insights, we also introduce a novel Domain-Specificity Disentanglement criterion to determine if the task-specific and domain-specific factors are well disentangled. The proposed approach outperforms several state-of-the-art benchmarks for single-source, and multi-source.

Acknowledgements. This work was supported by Kotak IISc AI-ML Centre (KIAC).

References

[1] Surbhi Aggarwal, Jogendra Nath Kundu, R. Venkatesh Babu, and Anirban Chakraborty. WAMDA: Weighted alignment of sources for multi-source domain adaptation. In *BMVC*, 2020. 8

[2] Sk. Miraj Ahmed, Dripta S. Raychaudhuri, S. Paul, Samet Oymak, and Amit K. Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *CVPR*, 2021. 8

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 1

[4] Jinming Cao, Oren Katzir, Peng Jiang, Dani Lischinski, Daniel Cohen-Or, Changhe Tu, and Yangyan Li. DiDA: Iterative boosting of disentangled synthesis and domain adaptation. In *IEEE ITME*, 2021. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[6] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, 2019. 2

[7] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *ECCV*, 2020. 2

[8] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. HoMM: Higher-order moment matching for unsupervised domain adaptation. In *AAAI*, 2020. 2, 4

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[10] Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. In *NeurIPS*, 2021. 8

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[12] Qi Dou, Cheng Ouyang, Cheng Chen, Hao Chen, and Pheng-Ann Heng. Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. In *IJCAI*, 2018. 1

[13] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *CVPR*, 2021. 7

[14] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *CVPR*, 2021. 1, 2, 3

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 3

[16] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *ICLR*, 2017. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[18] 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. 2

[19] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 2

[20] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *CVPR*, 2018. 2

[21] Ronghang Hu and Amanpreet Singh. UniT: Multimodal multitask learning with a unified transformer. In *ICCV*, 2021. 2, 3

[22] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *NeurIPS*, 2021. 7

[23] Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *ECCV*, 2018. 3

[24] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Re-energizing domain discriminator with sample relabeling for adversarial domain adaptation. In *ICCV*, 2021. 7

[25] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019. 7

[26] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021. 2

[27] Jogendra Nath Kundu, Suvaansh Bhambri, Akshay Kulkarni, Hiran Sarkar, Varun Jampani, and R Venkatesh Babu. Concurrent subsidiary supervision for unsupervised source-free domain adaptation. In *ECCV*, 2022. 2

[28] Jogendra Nath Kundu, Suvaansh Bhambri, Akshay R Kulkarni, Hiran Sarkar, Varun Jampani, and R Venkatesh Babu. Subsidiary prototype alignment for universal domain adaptation. In *NeurIPS*, 2022. 2

[29] Jogendra Nath Kundu, Akshay Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Kulkarni, Varun Jampani, and R. Venkatesh Babu. Balancing discriminability and transferability for source-free domain adaptation. In *ICML*, 2022. 2, 6, 7

[30] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R. Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, 2021. 2, 4

[31] Jogendra Nath Kundu, Naveen Venkat, M V Rahul, and R. Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, 2020. 1, 2, 3

[32] Seunghun Lee, Sunghyun Cho, and Sunghoon Im. DRANet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In *CVPR*, 2021. 2

[33] Suhyeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. Unsupervised domain adaptation for semantic segmentation by content transfer. In *AAAI*, 2021. 2
[34] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 2
[35] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *ICCV*, 2021. 6
[36] Xiangyu Li, Yonghong Hou, Pichao Wang, Zhimin Gao, Mingliang Xu, and Wanqing Li. Trear: Transformer-based rgb-d egocentric action recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14(1):246–252, 2021. 3
[37] 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. 2, 3, 5, 6, 7, 8
[38] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 6, 7, 8
[39] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *arXiv preprint arXiv:2111.06091*, 2021. 2
[40] Yu Mitsuzumi, Go Irie, Daiki Ikami, and Takashi Shibata. Generalized domain adaptation. In *CVPR*, 2021. 4
[41] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. FixBi: Bridging domain spaces for unsupervised domain adaptation. In *CVPR*, 2021. 7
[42] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 2, 6
[43] 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. 6
[44] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. SENTRY: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *ICCV*, 2021. 6
[45] Yao Qin, Chiyuan Zhang, Ting Chen, Balaji Lakshminarayanan, Alex Beutel, and Xuezhi Wang. Understanding and improving robustness of vision transformers through patch-based negative augmentation. In *NeurIPS Workshop*, 2021. 4
[46] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. In *IJCAI*, 2021. 6, 7
[47] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2, 6
[48] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018. 2
[49] Marin Scalbert, Maria Vakalopoulou, and Florent Couzini'e-Devy. Multi-source domain adaptation via supervised contrastive learning and confident consistency regularization. In *BMVC*, 2021. 8
[50] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *CVPR*, 2022. 2, 6, 7
[51] Jiayi Tian, Jing Zhang, Wen Li, and Dong Xu. VDM-DA: Virtual domain modeling for source data-free domain adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. 7
[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 6, 7
[53] 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. 3
[54] 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. 3
[55] Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and R. Venkatesh Babu. Your classifier can secretly suffice multi-source domain adaptation. In *NeurIPS*, 2020. 8
[56] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 6
[57] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *CVPR*, 2022. 6, 7
[58] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, 2021. 6, 7
[59] 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, 6, 7, 8
[60] Guanglei Yang, Hao Tang, Zhun Zhong, Mingli Ding, Ling Shao, Nicu Sebe, and Elisa Ricci. Transformer-based source-free domain adaptation. In *APIN*, 2022. 2
[61] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. TVT: Transferable vision transformer for unsupervised domain adaptation. In *WACV*, 2023. 2, 6, 7
[62] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *NeurIPS*, 2021. 2, 3, 5, 6, 7
[63] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *ICCV*, 2021. 5, 6
[64] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013. 2
[65] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019. 7