SALAD : Source-free Active Label-Agnostic Domain Adaptation for Classification, Segmentation and Detection

Divya Kothandaraman[1], Sumit Shekhar[3], Abhilasha Sancheti[1,3],
Manoj Ghuhan[2], Tripti Shukla[3], Dinesh Manocha[1]
University of Maryland College Park[1], Carnegie Mellon University[2],
Adobe Research[3]

Abstract

We present a novel method, SALAD, for the challenging vision task of adapting a pre-trained "source" domain network to a "target" domain, with a small budget for annotation in the "target" domain and a shift in the label space. Further, the task assumes that the source data is not available for adaptation, due to privacy concerns or otherwise. We postulate that such systems need to jointly optimize the dual task of (i) selecting fixed number of samples from the target domain for annotation and (ii) transfer of knowledge from the pre-trained network to the target domain. To do this, SALAD consists of a novel Guided Attention Transfer Network (GATN) and an active learning function, H_{AL}. The GATN enables feature distillation from pre-trained network to the target network, complemented with the target samples mined by H_{AL} using transfer-ability and uncertainty criteria. SALAD has three key benefits: (i) it is task-agnostic, and can be applied across various visual tasks such as classification, segmentation and detection; (ii) it can handle shifts in output label space from the pre-trained source network to the target domain; (iii) it does not require access to source data for adaptation. We conduct extensive experiments across 3 visual tasks, viz. digits classification (MNIST, SVHN, VISDA), synthetic (GTA5) to real (CityScapes) image segmentation, and document layout detection (PubLayNet to DSSE). We show that our source-free approach, SALAD, results in an improvement of $0.5\%-31.3\%$ (across datasets and tasks) over prior adaptation methods that assume access to large amounts of annotated source data for adaptation. Code is available here.

1. Introduction

Deep learning solutions for visual applications such as semantic segmentation [43, 40], image classification, and document layout analysis [18, 32] require a large amount of annotated data. Two popular trends to deal with the lack of sufficient annotated data are Domain Adaptation (DA) and Active Learning (AL). In Active Learning (AL) [2, 28, 36, 33], the model mines and annotates samples within a fixed budget (*e.g.* 5% of the available corpus of unlabeled data [7]) to maximize the models performance. Typical active learning strategies include modelling diversity and uncertainty for efficient sampling [7, 4]. Domain adaptation [43] aims at transferring knowledge from a "source" domain to the "target" domain of interest.

An amalgamation of active learning and domain adaptation, Active Domain Adaptation (ADA) [4, 39, 28] has explored the use of annotated "source"-data from a related domain to adapt to the "target"-domain dataset, within a fixed budget of annotating the "target" data. The drawback of ADA is that it requires access to annotated source data, which might be prohibitive due to privacy issues or storage constraints [16, 17]. Recently, there have been explorations on source-free adaptation as well [12, 1]. However, these methods don't generalize across label shifts and tasks, which can be enabled by annotating a small budget of samples from the target domain via active learning.

1.1. Main Contributions:

In this work, we focus on the novel problem of Source-free Active Domain Adaptation, SF-ADA, where we have access to a pre-trained "source" network, but the source data is not available due to privacy concerns or otherwise. Further, an unlabeled target dataset, and a small budget for acquiring labels in the target domain is specified. Moreover, the target domain may also have a shift in the label space from the source domain. We propose **SALAD**, a generic novel framework for SF-ADA, which jointly optimizes sampling target data for annotation and source-free adaptation of the neural network to the target domain.

SALAD holistically addresses two key challenges of the problem setting via two complementary components: the Guided Attention Transfer Network (GATN) for source-free adaptation from the pre-trained network to the target domain and an active learning algorithm, H_{AL}, for mining samples from the target domain for annotation:

- *Source-free Adaptation:* GATN enables adaptation at the feature level from the pre-trained network to the target network (Figure 1). GATN uses a transformation network to modulate the features of the pre-trained network, in alignment with the target domain, followed by guided attention for selective distillation to the target network. The target network is guided in adaptation via samples selected through active learning.

- *Active Learning:* The effectiveness of GATN depends on the samples selected for annotation from the target dataset. While it is important to choose samples that are similar to the source distribution, we need to ensure that the chosen samples are informative to the network w.r.t. the target dataset. To do so, H_{AL} combines adaptability from the pre-trained network, as well as uncertainty w.r.t. the target network.

Our method does not attempt to emulate source data using generative approaches, which is common in task-specific source-free domain adaptation [16, 21]. This makes our neural networks easy to train. SALAD has multiple benefits. (i) The architecture is task-agnostic, and can be applied across various visual tasks such as classification, detection and segmentation. (ii) Adaptation happens at the feature space (output of the network before the decoder). Thus, our architecture is label space-agnostic and can handle shifts in label space, where the source and target domains contain different number and types of classes (iii) The source data is not required for adaptation. Further, the pretrained source network is not required while testing and can be discarded after training.

We evaluate SALAD across three tasks. On classification datasets (MNIST, SVHN), we demonstrate that even without the source data, SALAD performs similar to or better than the prior active domain adaptation methods [39, 28] that use large amounts of annotated source data. Next, we evaluate on MNIST under 2 distinct cases of shift in output label space and show that SALAD is able to achieve 99.4% of the accuracy in case of no shift in the label space, thus establishing the effectiveness of our model in scenarios with label shift. Our experiments on the CityScapes dataset for semantic segmentation improves accuracy by 5.57% over fine-tuning. We also highlight the benefits of SALAD over other adaptation paradigms in Table 7. Finally, we conduct experiments on adaptation for document layout detection from PubLayNet to DSSE, where there is a shift in label space. SALAD imparts a relative improvement of 31.3% over fine-tuning of the target network on the small dataset.

Problem	Src. Data	Src. Model	Lab. Tar.	Un. Tar.
Semi Supervised DA (SSDA) [46]	✓	✓	✓	✓
Unsupervised DA (UDA) [40]	✓	✓	✗	✓
Source-Free DA (SFDA) [16]	✗	✓	✗	✓
Active DA (ADA) [39]	✓	✓	✓	✓
Source Free Active DA (SF-ADA)	✗	✓	✓	✓

Table 1: **Problem Settings:** We highlight various domain adaptation settings. Src. Data, Src. Model, Lab. Tar., and Un. Tar. refer to abundant labeled source data, Source Model, Scarce Labeled Target Data and Unlabeled Abundant Target Data, respectively.

2. Related Work

To the best of our knowledge, there is not much prior work on an approach for source-free active adaptation that can generalize across different tasks like classification and detection.

Active Learning Active Learning (AL) aims to acquire a given small budget of labeled data while maximizing supervised training performance. Uncertainty-based methods select examples with the highest uncertainty [44, 31], using entropy [44], minimum classification margins [31], least confidence, etc. Diversity-based methods choose some points representative of the data, e.g. core-set selection [36, 38]. Recent approaches combine these two paradigms [2, 28, 51].

Domain Adaptation Domain adaptation aims to transfer the knowledge learned by a source domain model to an unlabeled target domain. Some of the existing works align feature spaces of the source and target domains by learning domain invariant feature representations by divergence-based measure minimization [13, 24], adversarial training [35, 37, 41], source or target domain data reconstruction [3, 9], image-to-image translation [26, 11] or normalization statistics [27, 20]. However, domain adaptation methods typically require access to annotated source data.

Active Domain Adaptation Active domain adaptation aims to adapt a model trained on source domain data to target domain by annotating a fixed budget of target domain samples. [30] introduced the task of ADA with applications to sentiment classification for textual data. They proposed a method employing a sampling strategy based on model uncertainty and a learned domain separator. More recently, [39] studied ADA in the context of CNNs and proposed a method wherein samples are selected based on their uncertainty and targetedness, followed by adversarial domain adaptation. [34] proposed an algorithm that identifies uncertain and diverse instances for labeling followed by semi-supervised DA. [52] proposed a three-stage active adversarial training of neural networks using invariant feature space learning, uncertainty and diversity-based criteria for sample selection and re-training. [4] addressed the problem of lack of guarantee of good transfer-ability of features in domain adaptation. However, all of the above works use source domain data, which is prohibitive in terms of data privacy.

Source-free Domain Adaptation [19] introduced the paradigm of domain adaptation where source domain data is not available due to privacy issues and only a model pre-trained on the source domain data is available. Existing works employ generative approaches where the trained model is used to generate source samples using batch normalization [12] or energy-based methods [17] for classification task [48, 14, 1]. Others use a combination of distillation-based approach [23] or information maximization-based approach [21]. However, these methods do not consider using active learning to boost the performance, and typically do not generalize across tasks.

3. SALAD

We propose a novel method, SALAD , for the problem statement where we assume (i) a network, N_S, pre-trained (and frozen) on a source-domain, S (the source data is not available for adaptation) and (ii) an unlabeled target domain T from which we are allowed to annotate B images. The goal is to mine B images in total Tot_{AL} cycles, from the target domain for adapting network N_S to a target network, N_T which learns robust task-specific features for the target domain. For each cycle, $c = 1, 2, ..., Tot_{AL}$ of active learning, the labeled set in target domain is denoted as $\mathcal{I}_{T,L,c}$ and unlabeled set as $\mathcal{I}_{T,UL,c}$. The proposed solution, SALAD (Figure 1), consists of two components: (i) Guided Attention Transfer Network (GATN) for adaptation from the pre-trained network (ii) active learning strategy, H_{AL}, for optimized acquisition of samples from the target domain.

3.1. SALAD Description

SALAD consists of two complementary components: an adaptation strategy, Guided Attention Transfer Network (GATN) and an active learning strategy H_{AL}. GATN combined with H_{AL} enables domain-specific learning for target network using chosen samples, where relevant domain-agnostic knowledge is transferred from the pre-trained source network via guided attention. GATN performs adaptation at the feature level.

As shown in Figure 1, the frozen pre-trained network, N_S is first split into a frozen feature encoder which generates feature maps F_P and a frozen task head TH_P. Similarly, the trainable feature encoder of the target network, N_T is split into target feature map F_T and a trainable task head, TH_T. F_P and F_T are passed through the Guided Attention Transfer Network (GATN), which constrains the target network via a transfer learning loss L_{Tr}. The guided attention enables transfer of the domain-agnostic features from the pre-trained network, and discard domain-specific features. The GATN, as well as the pre-trained network, N_S, can be discarded after training, i.e. they are not required in the evaluation phase.

The active learning strategy, H_{AL} chooses samples using a combined metric of adaptability from source network and uncertainty in the target network, indicating the informativeness w.r.t. the target domain [28].

3.2. Guided Attention Transfer Network (GATN)

In this section, we describe the Guided Attention Transfer Network (GATN) in details. The GATN first transforms F_P to F_{P-tr} through a modulation network τ, which is a 4-layer fully convolutional network. Despite the transformation of the features from the pre-trained network, not all knowledge contained in F_{P-tr} is useful to the target domain. This is because τ is a CNN and has no filtering layers. It is important to transfer only domain agnostic features from F_{P-tr}. This is done by the guided attention networks. The transformed features F_{P-tr} and target network features F_T are then fed to two guided attention modules to compute attention across spatial and channel dimensions.

The guided attention modules consist of Guided Spatial Attention (GSA) and Guided Channel Attention (GCA) [47, 8] to compute attention across spatial and channel dimensions respectively. F_{P-tr}, along with F_T, are passed to spatial guided attention (GSA) and channel guided attention (GCA) modules. The attention modules compute alignment at the spatial and channel levels for the modulated source feature map F_{P-tr} and the target network feature map F_T. The target network contains finite domain-specific knowledge through the labeled target subset, that can be leveraged to guide adaptation. We build a guided attention module to do this, which builds on the math synonymous to self-attention [50]. We wish to reflect that the notion of guided attention is built on the idea described in co-attention [49]. Attention literature [42, 47, 8] describes the attention function as mapping a query and a set of key-value pairs to an output, where key, query, value, and output are all vectors. We denote F_{P-tr} as the query vectors and, key and value are assigned to target network feature maps F_T.

We incorporate the notion of guided spatial and channel attentions [47] using spatial and channel level feature vectors [8]. The goal of the Guided Spatial Attention (GSA) module (which generates attention representations A_{GSA}) is to highlight spatial regions of the transformed pre-trained network feature map, F_{P-tr} that align well with the target feature map, F_T. The goal of the Guided Channel Attention module (which generates attention representations A_{GCA}) is to highlight attributes (or channel level features at each spatial location) of transformed source network features F_{P-tr} at each spatial location that align well with the target network feature maps F_T. Let the dimensions of the feature maps be $C \times H \times W$. In spatial guided attention, $1x1$ convolution layers first transform the key, query, and value feature maps. This is followed by reshaping these feature maps to the shape $C \times (H \times W)$, which are used to compute

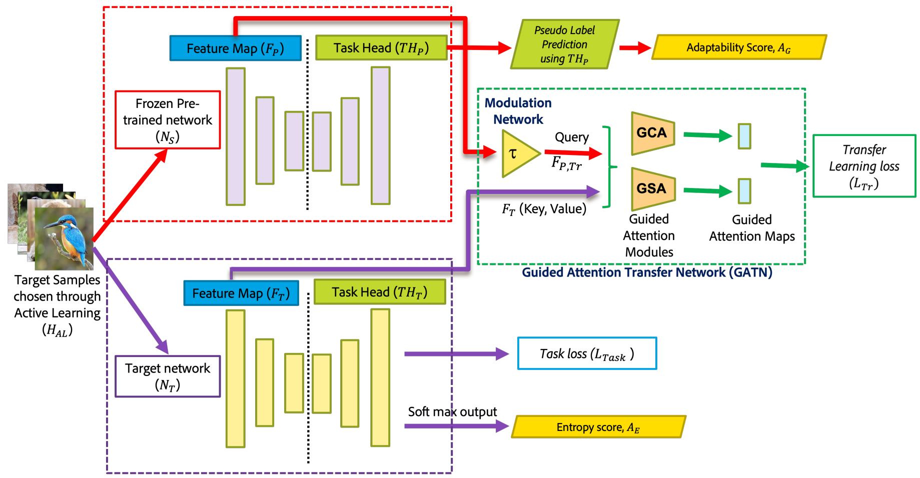

Figure 1: We present a generic method, SALAD, for the task of adapting from a pre-trained source network to a target domain with a small budget for acquiring annotation, under a possible shift in label space. SALAD consists of two complementary components: Guided Attention Transfer Network (GATN) and an active learning strategy H_{AL}.

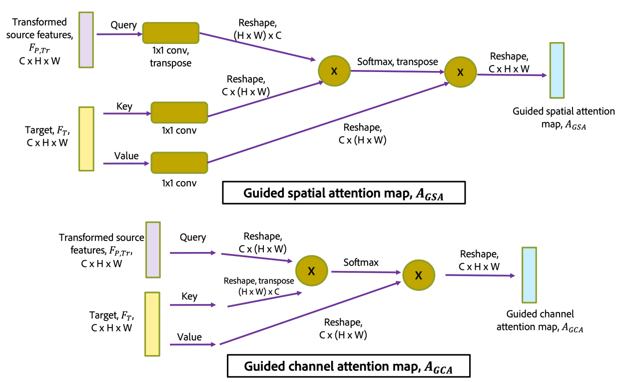

Figure 2: Overview of the computations for guided spatial attention and channel attention representations.

attention weights, A_{GSA}. For channel attention A_{GCA}, reshaped feature maps of dimensions $C \times (H \times W)$ are used to compute attention maps without any convolution. Figure 2 shows the computation of these attention maps. Mathematically,

$$A_{GSA} = S(C_q(F_{P-tr}^{\top}) \odot C_k(F_T))^{\top} \odot C_v(F_T),$$
$$A_{GCA} = S(F_{P-tr} \odot F_T^{\top}))^{\top} \odot F_T. \quad (1)$$

where $\top$, S, and $\odot$ denote transpose operation, softmax and matrix multiplication operations, respectively. C_k, C_q, C_v denote trainable 1×1 convolutions followed by reshaping of the key, query, and value feature maps respectively.

3.3. Training Losses

We jointly train GATN (consisting of τ and the guided attention networks), and the target network, N_T with the following loss terms:

- *Transfer learning loss L_{Tr}:* L_{Tr} is computed as the attention weighted mean squared difference between transformed pre-trained feature maps F_{P-tr} and target network feature maps F_T. Mathematically,

$$L_{Tr} = \sum A_{GSA} * [F_{P-tr} - F_T]^2 + \sum A_{GCA} * [F_{P-tr} - F_T]^2. \quad (2)$$

  This loss is applied to all target images, labeled as well as unlabeled, and scaled by hyper-parameters $\lambda_{Tr,L}$ and $\lambda_{Tr,UL}$, empirically chosen to 0.1 or 0.01 depending on the task. Hence, the total loss for labeled and unlabeled images is:

$$\lambda_{Tr,L} \sum_{\mathcal{I}_{T,L,c}} L_{Tr} + \lambda_{Tr,UL} \sum_{\mathcal{I}_{T,UL,c}} L_{Tr}$$

  .

- *Task-specific loss L_{Task}:* To learn target domain-specific information, we compute the task specific loss for the target domain images, *viz.* multi-class cross entropy for classification and semantic segmentation, and focal loss for object detection. This is computed for both labeled and unlabeled target samples. For unlabeled samples, pseudo label is computed by thresholding the soft max output of N_S. Overall task-specific loss is:

$$\sum_{\mathcal{I}_{T,L,c}} L_{Task} + \lambda_{pseudo} \sum_{\mathcal{I}_{T,UL,c}} L_{Task}$$

  .

- *Overall Loss:* The overall training loss is sum of the transfer and task losses computed above.

3.4. Active Learning Strategy H_{AL}

The training of GATN and the target network involves a task specific loss on the labeled subset of the target domain. Hence, the effectiveness of domain adaptation from the pre-trained network, and learning of domain specific features by the target network depend on the samples mined by the AL strategy H_{AL}. Thus, it is important to for H_{AL} to annotate samples (i) that facilitate adaptation from the pre-trained network as well as (ii) encode all aspects of target domain-specific knowledge.

- *Adaptability from the pre-trained network:* The prediction confidence of the pre-trained network, N_S, in computing the outputs for target domain samples implies the similarity of target samples to the source domain. Samples with high similarity facilitate knowledge transfer from the pre-trained network to the target network. We quantify 'similarity' by defining an 'adaptability score'. For computing the adaptability score, we threshold the final softmax output of the pre-trained network, N_S, to compute the pseudo label map for the target samples. The task-specific loss is computed between the output of the pre-trained network and the pseudo-labels. The gradient score or adaptability score, A_G is the total l_2 norm of the gradient (without any gradient update - since the network is frozen) of the network N_S *wrt* the computed loss. Low gradient implies high confidence and hence high adaptability from N_S.

- *Uncertainty of the target network*: We need to choose target domain samples that provide the target network with exhaustive information about the domain. Hence it is important to choose target domain samples that the network is unsure about, which can be quantified using the entropy of predictions from the target network. The softmax output of the target network provides the class-wise probability score map $\mathbf{p}$. This is used to compute the entropy score AE as $-\sum plogp$. High entropy for a target sample indicates high uncertainty and hence should be selected for labeling.

A combination of the above two measures is maximized greedily to mine the samples for labeling:

$$H_{AL} = -\lambda_G \log A_G + \lambda_E \log A_E \tag{3}$$

where λ_G, λ_E, are binary variables (0/1) that toggle the metrics used for sampling. For the first batch of AL, we set $\lambda_G = 1$, and $\lambda_E = 0$, as the target network, N_T is not trained. For further epochs, we set $\lambda_G, \lambda_E = 1$.

3.5. SALAD Training Overview

We train the target network, N_T and GATN network together by mining samples from the target domain dataset in mini-batches using H_{AL} until the desired budget B is achieved. We use the SGD optimizer with a learning rate of $2.5e-4$, and momentum of 0.9 and weight decay for 0.0005 for the training. The steps in the training routine are as follows:

- Initialize target network, N_T with parameters of the pre-trained source network, N_S

- For active learning cycles, c in range(0, Tot_{AL}):
  - Use the AL strategy H_{AL}, the pre-trained network N_S and the target network N_T to mine B/Tot_{AL} samples from the target dataset, and accumulate labeled target subset $\mathcal{I}_{T,L,c}$ and unlabeled target subset $\mathcal{I}_{T,UL,c}$.
  - Create batches of images combining $\mathcal{I}_{T,L,c}$ and $\mathcal{I}_{T,UL,c}$. Optimize overall training loss for N_T and GATN for given number of epochs or till convergence.
  - Sample for next active learning cycle using H_{AL}, using the current version of N_T and GATN.

4. Experiments and Results

We present results across the classification, detection and segmentation tasks, with standard evaluation metrics. Under classification settings, SALAD, even without access to annotated source data, performs similar to or better than (with a variance of 0.5% in accuracy) prior active domain adaptation methods [39, 28] that use large volume of annotated source data. Furthermore, we conduct experiments on MNIST under 2 distinct cases of shift in output label space, and show that SALAD can achieve 99.4% of the accuracy achieved when trained with all the labels, with only 5% sampling budget. Thus, SALAD can handle shifts in label space. Our experiments on CityScapes for semantic segmentation at various budgets reveal an improvement of 5.57% over fine-tuning (*i.e.* training the model without L_{tr}). Finally, we conduct experiments on document layout adaptation from PubLayNet to DSSE where there is a shift in output label space, SALAD imparts a improvement of 31.3% over fine-tuning (*i.e.* training the model without L_{tr}). We set number of cycles Tot_{AL} to be equal to 3 for MNIST, 1000 for SVHN, 50 for CityScapes, more implementation details can be found in the supplementary material.

4.1. Image Classification

We present our results on digits classification datasets under two settings: (i) shared label space, and (ii) shift in label space. In the shared label space setting, the label space that the pre-trained network was trained on and the label space of the target domains are the same. In the label space

Method	Source Data	B=100	B=200	B=300
Source only accuracy: 62.25				
O-ALDA [33]	✓	79.10	81.40	82.70
CDAN [25] + Entropy [39]	✓	93.10	94.60	95.00
CDAN [25] + BvSB [39]	✓	94.20	95.00	95.90
CDAN [25] + Uniform [28]	✓	90.00	94.00	94.50
CDAN [25] + BADGE [2]	✓	92.90	94.90	96.50
SSDA MME [34]	✓	93.00	95.00	95.50
AADA [39]	✓	94.20	95.20	95.50
CLUE [28]	✓	95.50	96.20	96.50
SALAD	✗	91.64	95.96	**97.16**

(a) **Results on adapting from SVHN to MNIST:** With a budget of 300 images (0.5% of target data MNIST, last column of table), we show that, even without source data, SALAD outperforms prior work on active domain adaptation, using annotated source data. The fully supervised accuracy is 99.2%, we achieve accuracies comparable to fully supervised accuracies using just 0.5% of the dataset.

Method	1000	2000	4000	10000
Source only accuracy: 27.27				
FT+Uniform	68.0	76.2	80.0	84.7
FT+Entropy	68.0	75.1	81.2	87.8
FT+BADGE [2]	70.1	79.2	83.7	88.1
FT+Coreset [36]	70.0	78.8	82.8	88.2
FT+Margin [31]	71.0	78.0	83.2	88.4
FT+CLUE [28]	72.1	76.4	83.0	87.8
SALAD	**74.2**	**82.2**	**86.6**	**88.6**

(b) **Results on adapting from MNIST to SVHN:** We compare with prior methods on active learning, and demonstrate state-of-the-art performance. The fully supervised accuracy is 90.44%, we achieve accuracies comparable to fully supervised accuracies using just 1.8% of the dataset.

Table 2: **Results on digits classification in the shared label space setting**

Method	Budget	Accuracy
SDDA (WACV 2021) [17]	-	75.5
SDDA-P (WACV 2021) [17]	-	76.3
SALAD	0.16%	91.64

(a) SVHN to MNIST

Method	Budget	Accuracy
SDDA (WACV 2021) [17]	-	42.2
SDDA-P (WACV 2021) [17]	-	43.6
SALAD	0.18%	74.2

(b) MNIST to SVHN

Table 3: **Comparing SALAD with prior art on SFDA on digits datasets.** Budget reflects the percentage of total number of target samples used for active learning. We demonstrate that our SF-ADA approach outperforms prior art on SFDA by a large margin using a very small proportion of annotated target samples.

Class/Exp.	mean	0	1	2	3	4	5	6	7	8	9
Case 1: Remove the digits ‘3’ and ‘9’ from source SVHN											
Source only	56.88	69.20	86.80	79.10	0.00	53.80	95.70	41.00	78.20	63.00	0.00
B=100	88.29	97.80	98.90	94.60	83.80	94.30	96.60	84.90	91.00	93.80	48.20
B=200	96.27	98.50	98.60	98.10	94.30	96.70	97.80	97.90	93.60	92.80	94.40
B=300	96.61	99.10	98.70	98.10	95.00	97.50	97.90	98.00	91.30	95.40	97.10
Case 2: Remove the digits ‘7’, ‘5’, ‘4’, ‘1’ from source SVHN											
Source only	41.90	0.00	70.80	65.60	89.60	0.00	0.00	83.00	0.00	66.00	47.60
B=100	87.11	98.80	99.40	93.80	97.30	82.70	49.90	94.20	85.60	66.40	95.40
B=200	92.53	97.60	98.90	94.10	97.40	96.10	60.80	95.90	90.50	94.50	94.60
B=300	97.00	99.20	98.90	97.20	99.10	96.20	95.70	97.50	95.30	94.70	96.30

Table 4: **Results on adaptation from SVHN to MNIST, for label space shift.** We consider two scenarios: case 1 - source data does not contain digits ‘3’ and ‘9’, case 2 - source data does not contain digits ‘7’, ‘5’, ‘4’, ‘1’. SALAD, with H_{AL} and GATN, achieves 99.4% and 99.8% of the accuracy in the scenario with no label shift. The accuracy at a budget of 300 images, without label shift, is 97.16%

shift space setting, the target dataset contains labels not used for training the source network. We used ResNet-101 features for the experiment, consistent with the baselines. We set $\lambda_{Tr} = 0.01$, and λ_G to 1.0 and λ_E to 1.0 after first round of sampling. We set $\lambda_{pseudo} = 1.0$ for the dataset. We use feature heads from the penultimate layer of the ResNet-101 [10] classifier backbone.

Shared label-space setting, SVHN to MNIST: Table 2a contains results of adapting from SVHN to MNIST, at various budgets. Concurrent with our intuition, the accuracy is better at higher budgets. When benchmarked at a budget of 300 samples, which is 0.5% of the total samples that MNIST contains, we observe that SALAD, even without any annotated source data, outperforms prior work on active domain adaptation which use large amounts of annotated source data (600000 images). Moreover, we observe that the accuracy of 97.16% with 300 images is 97.64% of fully supervised accuracy with $60k$ images.

Comparisons with SFDA methods (Table 3 SALAD outperforms prior art on SFDA by a large margin using a very small proportion of annotated target samples.

Comparisons with SFDA + AL methods. SALAD outperforms prior work on ADA. Since ADA (or AL + DA) methods are better than AL + SFDA methods, by transitivity, a holistic solution for SF-ADA, such as SALAD , is more beneficial than a naive combination of SFDA and AL.

Shared label-space setting, MNIST to SVHN: We show results on adaptation from MNIST to SVHN in Table 2b. The complexity of SVHN is higher than that of MNIST, reflected in the source-only accuracy which stands at 27%. As per the prior work, we cap the net budget for mining samples via active learning at 10000 images, which is 1.8% of the total size of the dataset. We demonstrate state-of-the-art performance at varying budget of 1000, 2000, 4000 and 10000 images. Moreover, the accuracy at $10k$ images is 93.44% of fully supervised accuracy, which uses around $500k$ images.

Shift in label space, SVHN to MNIST: In Table 4, we present results on adapting from SVHN to MNIST under a shift in label space. In case 1, we train the source network on SVHN after removing samples corresponding to two classes (class 3 and class 9, randomly chosen). Similarly, in case 2, we remove 4 classes from the source dataset. Direct testing reveals that the accuracy for these classes is 0.

GATN Specification		AL Specification		Budget	Acc
$\lambda_{Tr,L}$	$\lambda_{Tr,UL}$	A_G	A_E		
Baseline source-free accuracy: 62.25%					
0	0	✓	✗	300	88.96
0	0.01	✓	✗	300	89.56
0.01	0.01	✓	✗	300	92.56
0.01	0.01	✓	✓	300	97.16

Table 5: Ablation Experiments for variations in transfer loss parameters ($\lambda_{Tr,L}$, $\lambda_{Tr,UL}$) and active learning strategies (A_G, A_E) from SVHN to MNIST.

Sampling with our AL strategy §3.3 and gradually training with GATN gradually up to a budget of 300 images (again, 0.5% of target samples) restores accuracy to 96.61% and 97.00%, respectively, which is 99.4% and 99.8% of the accuracy achieved in the scenario with no label shift. The term A_E in H_{AL} ensures class balance, i.e., samples corresponding to classes not learnt by the pre-trained network are selected, along with classes with high prediction uncertainty. Thus, our method works well when there is a shift in label space.

Ablation experiments on H_{AL} and GATN. We present ablation experiments on adaptation from SVHN to MNIST in Table 5. We set the net active learning budget at 300 images. Since the network has no prior knowledge about the target domain, the uncertainty metric can be applied only from the second round of active sampling. In the first round of active sampling, we apply only the adaptability metric. In the first experiment, we study the impact of training the target network, without GATN, with all samples mined using just the adaptability score, the accuracy is 88.96%. This proves that even in the absence of adaptation from the pre-trained network, samples mined using the pre-trained networks' confidence score is advantageous since the target network is initialized with the weights of the pre-trained network. Next, we apply the distillation loss dictated by GATN only on the labeled subset and correspondingly set $L_{L,Tr} = 0.01$. We observe that knowledge transfer, in addition to sampling using the adaptability score improves performance by 0.6% points. This reinforces the quality of samples mined by the pre-trained network. Next, we apply the distillation loss to the unlabeled subset as well ($L_{Tr,UL}$), which leads to an absolute improvement of 3%. This is an indicator of selective distillation capabilities of GATN, where only useful features are distilled to improve performance. Finally, we experiment by using the uncertainty score, as well as the diversity score, to achieve an accuracy of 97.16%, an improvement of 34.91% points over the baseline and an improvement of 5.6% points over Experiment 3. Hence, the best adaptation using GATN is obtained when samples are mined intelligently (Experiment 4).

Ablation experiment on the modulation network τ. The accuracy on MNIST $\longrightarrow$ SVHN, with 300 images, without the modulation network is 94.15% while the accuracy with the modulation network is 97.16%.

4.2. Autonomous Driving: Synthetic to Real Segmentation on CityScapes

We conduct experiments on a dense pixel-level task, segmentation, where we adapt from GTA5 (25000 images) to CityScapes. To effectively transfer from GTA5 and address uncertainty of the target network while sampling, we set $\lambda_G = \lambda_E = 1$ from the second round of sampling. We set $\lambda_{Tr} = 0.01$. We use feature heads from the layer 3 of the underlying DeepLabv2 ResNet-101 backbone [5]. We present the results in Table 6. In the first row, we directly test the pre-trained GTA5 model, which gives an mIoU of 34.91. We next apply H_{AL} for batch active learning, and train the network using GATN after each round of sampling. A cumulative budget of 50 images, 100 images, 200 images, and 500 images leads to relative improvements (over the source only mIoU) of 31.5%, 46.03%, 52.5% and 62.1% respectively. Classes like 'Sidewalk', 'Wall', 'Fence', 'Pole', 'Sign', 'Rider', 'Train', 'MBike', 'Bike' have very low mIoU ($\sim 20\%$ or less) when directly transferred from the source model. We show that SALAD improves performance by $3\times$ to $25\times$. Class-wise comparisons with prior art on SFDA reveal that our AL heuristic H_{AL} strategically chooses classes like Bike and MBike with low confidence or high uncertainty to boost performance, while not compromising on performance w.r.t. transferable classes like road and sky.

Comparisons with SFDA methods: Table 7 shows the comparison of SALAD with prior SFDA methods. SALAD performs better with a budget of only 50 images (1.5%).

Ablation experiments We present ablation experiments in Table 8. In Table 8(a), we study the effectiveness of GATN at various budgets. In the second column, we apply L_{Tr} to only the labeled subset, along with active learning using H_{AL}. The third column reflects mIoUs obtained by using our complete model, through training on labeled as well as unlabeled subsets along with active learning using H_{AL}. A comparison of the second and third columns indicates the benefits of selective transfer learning using GATN. Moreover, our model results in improvements of 16.27%, 16.92%, 13.24% and 5.57% over the baseline numbers [46] obtained by naively fine-tuning the target network (with pre-trained weights initialization) without GATN, and by random sampling [46], at budgets of 50 images, 100 images, 200 images, and 500 images respectively. In Table 8(b), we study the effectiveness of the different components of GATN, at a budget of 50 images. In the first experiment, we do not use either GCA or GSA [46]. Without GATN, our system will reduce to simple fine-tuning of the target network with the annotated target samples. GATN forms a

Experiment	mIoU	mAcc	Road	Sidewalk	Building	Wall	Fence	Pole	Light	Sign	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MBike	Bike
Source only	34.91	77.84	70.14	21.6	76.27	18.8	16.27	21.31	27.85	15.40	77.67	31.29	74.83	49.47	3.60	79.45	28.71	31.39	4.70	12.43	2.10
TENT [45] (Source-free)	38.9	-	87.3	39.0	79.8	24.3	19.6	21.2	25.1	16.6	83.8	34.7	77.7	57.9	17.8	85.0	24.9	20.8	2.0	16.6	4.5
S4T [29] (Source-free)	43.98	-	88.2	44.2	84.4	28.9	27.6	38.6	41.5	8.4	86.3	41.0	79.2	58.7	25.3	85.4	20.1	26.4	6.3	10.8	8.4
B=50	45.93	89.22	92.09	52.57	83.43	23.72	18.37	33.33	35.90	44.01	84.24	39.23	85.82	55.39	20.16	84.02	38.57	37.77	2.72	16.09	25.26
B=100	50.98	90.33	93.6	57.79	84.16	23.4	21.98	36.07	38.12	45.8	85.39	41.33	86.34	57.67	30.41	86.1	43.81	45.02	26.02	19.15	46.48
B=200	53.34	91.18	94.82	63.83	85.29	29.01	27.85	36.84	39.84	47.53	86.33	42.16	88.4	60.16	31.85	86.88	48.64	48.45	26.29	20.46	48.91
B=500	56.59	92.09	95.59	68.77	86.41	33.08	34.88	39.49	42.54	52.44	87.30	48.17	89.73	62.96	33.91	88.18	53.67	52.41	29.08	23.46	53.15

Table 6: **GTA5 to CityScapes Adaptation:** We show that SALAD imparts a relative improvement of 31.5%, 46.03%, 52.5% and 62.1% over the baseline source model with budgets of 50, 100, 200, and 500 images, with $3\times -25\times$ improvement on specific classes like "Bike", "Train", "MBike", "Sidewalk", etc.

Method	mIoU
UBNA [15] (WACVW 2022)	36.1
UBNA+ [15] (WACVW 2022)	36.5
TENT [45] (ICLR 2021)	38.86
TENT + MS [45] (ICLR 2021)	36.89
SFDA (w/o IPSM) [23] (CVPR 2021)	41.35
SFDA [23] (CVPR 2021)	43.16
URMA [6] (CVPR 2021)	45.1
S4T [29] (ArXiv 2021)	43.98
S4T + MS [29] (ArXiv 2021)	44.83
SALAD	45.93

Table 7: **Comparisons with the state-of-the-art SFDA approaches for adaptation from GTA to CityScapes.** We show that SALAD achieves state-of-the-art mIoU with a small budget of 50 images, the mIoU obtained by naive finetuning [46] on 50 images is 39.5.

Budget B	($\lambda_{Tr,L}$, (0.1,0)	$\lambda_{Tr,UL}$) (0.1,0.1)
50	44.62	45.93
100	48.72	50.98
200	50.49	53.34
500	53.46	56.59

(a)

GSA	GCA	mIoU
✗	✗	39.50
✓	✗	45.43
✗	✓	45.45
✓	✓	45.93

(b)

H_{AL}	mIoU
A_G	50.29
$A_G + A_E$	56.59

(c)

Table 8: **Ablation experiments for adaptation from GTA5 to CityScapes.** In table (a), we study ablations on transfer loss for GATN, Table (b) shows ablations on GATN components at a budget of 50 images. In table (c), we ablate on the active learning heuristic, H_{AL} at a budget of 500 images.

Experiment	mAP	Text	Caption	Figure	Table	List	Section
FT w/o SALAD	23.11	36.12	13.49	25.60	22.24	29.57	11.64
FT w. SALAD	30.36	44.59	11.61	35.57	24.80	37.48	28.18

Table 9: **Adaptation for document layout detection from PubLayNet to DSSE:** Fine-tuning with SALAD improves performance by 31.3% over fine-tuning without SALAD.

bridge between the pre-trained network and the target network, and removing it would break the adaptation process. In the subsequent experiments, we show the impact of using channel and spatial features. In Table 8(c), we demonstrate the effectiveness of using the fusion AL heuristic comprising of the adaptability score w.r.t. the pre-trained network as well as the uncertainty score w.r.t. the target network, as opposed to using just the adaptability score.

4.3. Document Layout Detection: DSSE

In Table 9, we adapt from the medical documents dataset PubLayNet to documents belonging to DSSE, a dataset containing magazines, receipts and posters. The documents in the two domains are quite different. Medical documents are written in a two-column format, with uniform text, figures and tables. In contrast, the target domain, DSSE, a new unseen dataset, is small (only 150 documents) and is extremely diverse. Moreover, PubLayNet has 5 classes, while DSSE has 6 classes. Hence, there is a *shift* in label space. Direct testing of PubLayNet on DSSE results in a mAP of 15.67. Since the dataset is very small, we do not apply H_{AL}, and instead directly use all 150 images for GATN. We use feature heads from the FPN of the underlying RetinaNet ResNet-101 backbone [22]. Fine-tuning without SALAD results in a mAP of 23.11, and fine-tuning with SALAD improves performance by 31.3%, to 30.36.

5. Conclusions, Limitations and Future Work

We propose a generic source-free method, SALAD, for the task of adapting from a pretrained network to target domain, under a possible shift in label space, with the provision to annotate a small budget of samples in the target domain. SALAD consists of two complementary components: an active learning strategy H_{AL}, and GATN for effective adaptation and sampling. We evaluate the performance across 3 tasks and show improved or on par performance with methods using source data. One drawback of our method is that we use binary weights for the scores in H_{AL}, using learnable weights could be an interesting direction for future work. Moreover, we expect that SALAD can be extended to tasks and modalities where traditional domain adaptation has been useful. These include text classification, neural machine translation, sentiment analysis, cross lingual question answering, and domain stylization.

Acknowledgements: This research has been supported by ARO Grants W911NF2110026 and Army Cooperative Agreement W911NF2120076

References

[1] Peshal Agarwal, Danda Pani Paudel, Jan-Nico Zaech, and Luc Van Gool. Unsupervised robust domain adaptation without source data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2009–2018, 2022.

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29:343–351, 2016.

[4] Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami, and Céline Hudelot. Stochastic adversarial gradient embedding for active domain adaptation. *arXiv preprint arXiv:2012.01843*, 2020.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Francois Fleuret et al. Uncertainty reduction for model adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9613–9623, 2021.

[7] Bo Fu, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Transferable query selection for active domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7272–7281, 2021.

[8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European conference on computer vision*, pages 597–613. Springer, 2016.

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

[11] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[12] Masato Ishii and Masashi Sugiyama. Source-free domain adaptation via distributional alignment by matching batch normalization statistics. *arXiv preprint arXiv:2101.10842*, 2021.

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

[14] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data. *arXiv preprint arXiv:2007.01524*, 2020.

[15] Marvin Klingner, Jan-Aike Termöhlen, Jacob Ritterbach, and Tim Fingscheidt. Unsupervised batchnorm adaptation (ubna): A domain adaptation method for semantic segmentation without using source domain representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 210–220, 2022.

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

[17] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625, 2021.

[18] Kai Li, Curtis Wigington, Chris Tensmeyer, Handong Zhao, Nikolaos Barmpalios, Vlad I Morariu, Varun Manjunatha, Tong Sun, and Yun Fu. Cross-domain document object detection: Benchmark suite and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12915–12924, 2020.

[19] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[20] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[21] 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.

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[23] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2021.

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

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

[26] Xudong Mao and Qing Li. Unpaired multi-domain image generation via regularized conditional gans. *arXiv preprint arXiv:1805.02456*, 2018.

[27] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5067–5075, 2017.

[28] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. Active domain adaptation via clustering uncertainty-weighted embeddings. *arXiv preprint arXiv:2010.08666*, 2020.

[29] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. S4t: Source-free domain adaptation for semantic segmentation via self-supervised selective self-training. *arXiv preprint arXiv:2107.10140*, 2021.

[30] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, 2010.

[31] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, pages 413–424. Springer, 2006.

[32] Diede Rusticus, Lutz Goldmann, Matthias Reisser, and Mauricio Villegas. Document domain adaptation with generative adversarial networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1432–1437. IEEE, 2019.

[33] Avishek Saha, Piyush Rai, Hal Daumé, Suresh Venkatasubramanian, and Scott L DuVall. Active supervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 97–112. Springer, 2011.

[34] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[35] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[36] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[37] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[38] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[39] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 739–748, 2020.

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

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

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[43] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[44] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014.

[45] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[46] Zhonghao Wang, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-Mei Hwu, Thomas S Huang, and Honghui Shi. Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 936–937, 2020.

[47] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[48] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9010–9019, 2021.

[49] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019.

[50] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[51] Fedor Zhdanov. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

[52] Fan Zhou, Changjian Shui, Shichun Yang, Bincheng Huang, Boyu Wang, and Brahim Chaib-draa. Discriminative active learning for domain adaptation. *Knowledge-Based Systems*, 222:106986, 2021.