Adapting Models to Scarce Target Data without Source Samples

JoonHo Lee[1,2] and Gyemin Lee[1]

[1] Seoul National University of Science and Technology, Seoul, Korea
[2] Samsung SDS Technology Research, Seoul, Korea*
{joonholee, gyemin}@seoultech.ac.kr

Abstract. When significant discrepancies exist in data distributions between source and target domains, source-trained models often exhibit suboptimal performance in the target domain. Unsupervised domain adaptation (UDA) effectively addresses this issue without needing labels of target data. More recent source-free UDA methods handle the situations where source data is inaccessible. However, the performance of UDA is substantially compromised when the target domain data is scarce. Despite the challenges in obtaining and storing large target data, this aspect of UDA has not been extensively investigated. Our study introduces a new method to alleviate performance degradation in source-free UDA under target data scarcity. The proposed method retains the architecture and pretrained parameters of the source model, thereby reducing the risk of overfitting. Instead, it incorporates less than 3.3% of trainable parameters that comprise a set of convolution layers with non-linearity and a spatial attention network. Empirical assessments reveal that our approach achieves up to 5.4% performance improvement with limited target data on VisDA benchmark over existing UDA methods. Similar trends are also evident in Office-31 benchmark and multi-source UDA experiments with Office-Home benchmark across different target domains. Our method shows promising enhancement of the adapted model's generalization. These findings highlight the efficacy of our method in improving UDA across diverse domain adaptation scenarios.

Keywords: Unsupervised Domain Adaptation · Source-Free Domain Adaptation · Scarce Target Data

1 Introduction

Deep neural networks trained on large amounts of data have shown remarkable successes across various tasks. Nowadays many such pretrained models are available ready to be used in diverse target applications. However, these models typically suffer from accuracy degradation in reality. This deterioration originates from the discrepancy between the source domain where models are trained and the target domain where trained models are deployed. Fine-tuning demands costly efforts of target domain data collection and labeling.

* This work was conducted as part of the Ph.D. program and is independent of Samsung SDS.

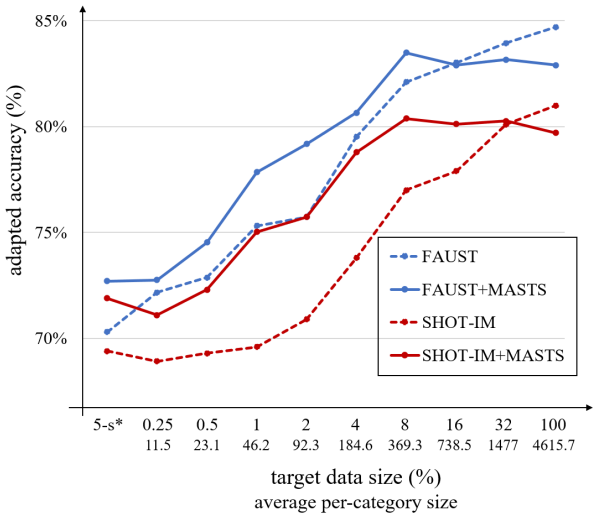


Fig. 1: (Best viewed in color) The performance of a typical unsupervised domain adaptation (UDA) method significantly declines with the reduction of target data as depicted with dashed lines. The proposed Model Adaptation to Limited Target Samples (MALTS) effectively addresses this overfitting issue as presented by solid lines. MALTS freezes the source model and updates only a small number of domain-specific parameters. Our results suggest that limiting the number of trainable parameters is beneficial when target domain samples are scarce.

Domain adaptation provides an alternate approach of addressing this domain shift problem. Unsupervised domain adaptation (UDA) considers a particular scenario where we have access to labeled source data and only unlabeled target data. The goal is to adapt a pretrained source model to the target domain. Many UDA methods assume that source samples are available and leverage them during model adaptation [3, 8, 21, 25]. However, this assumption is often impractical because of data privacy issues and inefficiency in heavy data transmission. In more recently emerged *source-free* UDA methods, only a pretrained source model and unlabeled target samples are given [7, 9, 13, 24].

We propose to go one step further: to achieve the same goal with **limited target data** on hand. Thus, we can save storage (*source-free* and *scarce target*) and avoid expensive data labeling (*unsupervised*). This setup is relevant to the limited storage of edge computing and end-user mobile devices.

However, we encounter many challenges. When target data is scarce, UDA methods suffer from overfitting and their target accuracies drop sharply (Fig. 1). As UDA is primarily concerned with maximizing the performance on the target domain, adapted models tend to forget the source domain (Sec. 5.4) [2, 12].

To address such challenges, we propose a novel Model Adaptation method to Limited Target Samples (MALTS). Typical UDA methods adapt all the parameters of the pretrained source model to the target domain. Instead, MALTS updates only a small number of domain-specific parameters while keeping the entire source model parame-

ters frozen. As MALTS preserves the original source model parameters, both domains share a substantial amount of parameters. This high-degree of parameter sharing enables to learn the target domain without forgetting the source domain.

MALTS involves integrating lightweight networks called CASA into the source model. We design CASA with a convolutional adapter, a dropout layer and a spatial attention module. The networks add a small fraction of parameters ($\sim$3%). In MALTS, we train only these networks using existing UDA methods. Thus, MALTS better controls overfitting when target samples are scarce.

We demonstrate that the proposed method is robust to overfitting across various target-scarce UDA tasks. On VisDA, Office-31 and Office-Home, MALTS consistently outperforms baseline UDA methods by a large margin (up to 5.4%). When the target adapted models are evaluated on the source domain data, the results confirm that MALTS can learn the target domain without forgetting the source domain.

Our key contributions are summarized as follows:

- We investigate UDA methods that their performance sharply diminishes as the target data size decreases and that they are likely to forget the source domain knowledge after adaptation.
- We propose Model Adaptation to Limited Target Samples (MALTS) to address the *target data scarcity* and *source domain forgetting*.
- We design a CASA network that consists of a convolutional adapter and a spatial attention module. MALTS integrates this lightweight network into the frozen pretrained source model and updates only a small number of domain-specific parameters in CASA.
- On VisDA, Office-31 and Office-Home, we empirically evaluate our method across various target-scarce UDA tasks. Our method is *parameter-efficient* and *robust to overfitting* and *learns without forgetting* the source domain.

2 Related Work

2.1 Source-Free Unsupervised Domain Adaptation

UDA aims to generate models that accurately predict labels for new target samples using labeled source and unlabeled target data [1, 3, 8, 10, 21, 25].

Source-free UDA addresses the practical challenges of unavailable source data. Unlike vanilla UDA, source-free UDA operates without source data access [7, 9, 13, 24]. These methods often assume that the target data is sufficient in quantity while unlabeled. Among these, we direct our attention to two approaches: SHOT [13] and FAUST [7]. SHOT is a seminal work in source-free UDA that is still comparable to recent works. FAUST is one of the most recent works. Both methods freeze the head classifier of the pretrained source model and focus on training the remaining components. With the source head classifier fixed, SHOT-IM trains the feature extractor using mutual information maximization objective. FAUST leverages multiple perturbed views of an input. It enforces consistency between the feature embedding from each perturbation and the original feature embedding, and consistency between the prediction for each perturbed input and the prediction of a feature-based classifier. Their details are presented in Sec 3.2.

2.2 Parameter-Efficient Transfer Learning

Parameter-Efficient Transfer Learning (PETL) was initially proposed as an effective alternative to the conventional fine-tuning approach in computer vision. Recently, it has gained prominence in the field of language modeling. This approach typically incorporates lightweight adapter modules into a pretrained model, and updates only these adapters for downstream tasks. [19] addresses versatile visual representation learning for universal image analysis applications. They use a tunable architecture with adapter residual modules. In language models, prior methods such as [5, 6, 11] mainly concentrate on fine-tuning pretrained representations with labeled target domain data. Though PETL is actively explored, its application in UDA has been limited. [15] presents a parameter efficient method for domain adaptation. However, as [15] only adapts BatchNorm statistics, feature distributions from source and target domains can still remain unaligned. Inspired by PETL, our study proposes a UDA method for better feature alignment to tackle scenarios with scarce target data.

3 Background

3.1 Notation

We let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ denote the spaces of inputs, labels and feature tensors, respectively. We consider that a model $f : \mathcal{X} \mapsto \mathcal{Y}$ is a composition of a *feature generator* $g : \mathcal{X} \mapsto \mathcal{Z}$ and a *head classifier* $h : \mathcal{Z} \mapsto \mathcal{Y}$. That is, $f = h \circ g$.

We formulate our UDA problem as follows: a source model is pretrained with a set of n labeled samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $\mathbf{x}_i$ represents a source sample and y_i its corresponding label. A learning algorithm has access to the pretrained source model and a set of m unlabeled samples $\{\mathbf{x}_j\}_{j=1}^{m}$ from target domain. Both domains share the same label space $\mathcal{Y} = \{1, \ldots, K\}$.

3.2 Baseline UDA Methods

For baselines, we choose two source-free UDA methods: SHOT-IM, recognized for its simplicity and effectiveness, and FAUST, a stronger baseline with superior performance. Both methods fix the head of the source classifier h and train the feature generator g to produce the target-adapted feature embedding $z = g(x)$.

SHOT-IM is designed to enhance global diversity and individual certainty in target outputs. It achieves this goal by increasing the marginal entropy $\mathcal{H}(\mathcal{Y})$ to foster a uniform label distribution and by decreasing the conditional entropy $\mathcal{H}(\mathcal{Y}|\mathcal{X})$ to encourage distinct predictions. This strategy leads to a mutual information maximization objective for the training of the feature generator g as follows:

$$\min_{g} \; \mathcal{H}(f(x)) + D_{\text{KL}}\left(\bar{f}(x) \middle\| \frac{1}{K}\mathbf{1}_K\right) - \log K, \tag{1}$$

where $D_{\text{KL}}(\cdot, \cdot)$ denotes the Kullback-Leibler divergence. The term $\bar{f}(x)$ refers to the mean embedding of predictions under the target distribution, and $\mathbf{1}_K$ is a K-dimensional

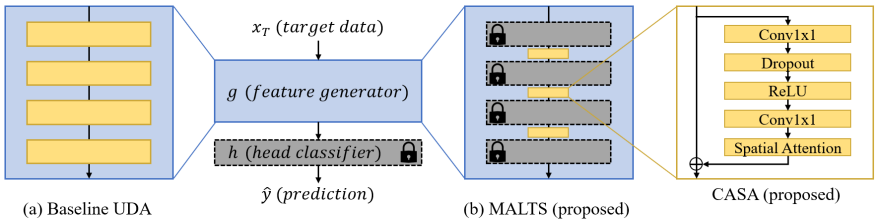


Fig. 2: Overall conceptual comparison between (a) a standard UDA and (b) the proposed method (MALTS). Because a typical UDA method trains a large number of parameters in the source model, it can suffer from overfitting. On the other hand, MALTS keeps the parameters of the source model frozen (indicated by dashed-line blocks). Instead, MALTS updates a small amount of newly added parameters in CASAs. Thus, MALTS is robust to overfitting and less susceptible to domain forgetting. The CASA module consists of a convolutional adapter network with dropout and a subsequent spatial attention network.

vector of ones. While SHOT-IM originally enhances the base model by incorporating additional norm layers and a bottleneck module during source model training, we preserve the base model architecture to avoid the need for re-training.

On the other hand, FAUST addresses domain divergence by capturing data uncertainty through perturbed inputs. It encourages the feature generator g to learn consistent features distant from the decision boundaries of the head classifier h using two consistency objectives. For a set of v perturbed views $x_a^{(1)}, \cdots, x_a^{(v)}$ of an input image x, combining *inter-space* and *intra-space* consistency losses comes to

$$\min_g \frac{1}{v} \sum_{i=1}^{v} \left\{ \mathcal{H}(f^{pl}(x), f(x_a^{(i)})) + D(g(x), g(x_a^{(i)})) \right\}. \tag{2}$$

Here, $\mathcal{H}(\cdot,\cdot)$ and $D(\cdot,\cdot)$ denote the cross-entropy and the cosine dissimilarity, respectively. For the soft pseudo-label $f^{pl}(x)$ of a target input x, FAUST first computes the feature prototype of each class k by $c^k := \sum_{i \in \mathcal{B}} f^k(x_i) g(x_i)$ where x_i is empirical target data in each mini-batch $\mathcal{B}$ and $f^k(x_i)$ is the prediction probability of x_i to class k. Then, $f^{pl}(x)$ is

$$f^{pl}(x) := \text{softmax}(C^T g(x)), \tag{3}$$

where columns of the matrix C are the prototypes.

4 Model Adaptation to Scarce Target Data

The scarcity of target data can cause a UDA method suffer from overfitting and a subsequent sharp decline in target accuracy as demonstrated in Fig. 1. To address these issues, we propose Model Adaptation to Limited Target Samples (MALTS). MALTS freezes the entire source model. Instead, we append a small number of new parameters to the pretrained source model. During UDA, we train only these new parameters whereas the source model remains frozen. Since the number of the newly added trainable parameters is considerably smaller than the source model, this approach can be beneficial to

address forgetting issue on source domain and to prevent model overfitting with few target data.

In MALTS, we introduce a Convolutional Adapter with Spatial Attention (CASA) network to learn domain-specific parameters. The CASA network is appended to the end of each block of stacked layers. For instance, if the backbone network of the source model is ResNet [4], the proposed CASA network is placed after each of four ResNet blocks. This configuration reduces the number of added parameters and dependency on the pretrained source model architecture. For the l-th block g_l in g, the proposed CASA network is sequentially attached to g_l as follows:

$$g_l^{\text{MALTS}} \coloneqq \texttt{casa} \circ g_l, \tag{4}$$

where `casa` is the CASA network composed of a convolutional adapter network with dropout (`ca`) followed by a spatial attention network (`sa`). The `casa` network also contains a skip-connection as illustrated in Fig. 2b. Thus, `casa` can be formulated as follows:

$$\texttt{casa}(z_l) \coloneqq z_l + \texttt{sa} \circ \texttt{ca}(z_l), \tag{5}$$

where z_l is the output of g_l. The details of `ca` and `sa` are explained below.

Convolutional Adapter For `ca`, we implement two 1×1 convolutional layers with an intervening nonlinear activation layer. The proposed `ca` is defined as follows:

$$\texttt{ca}(z_l) \coloneqq \texttt{conv1x1}(\texttt{relu}(\texttt{dropout}(\texttt{conv1x1}(z_l)))). \tag{6}$$

We reduce feature map channels using the first 1×1 convolution and then apply an activation layer to introduce nonlinearity. Subsequently, the second 1×1 convolution restores the number of channels back to the input feature map. This process extracts essential information across channels from the feature map, while preserving the spatial structures of both input and output. We match the number of input and output channels in `ca` with the adjacent blocks. The bottleneck channel is controlled by a ratio factor, r, considering target data size.

Spatial Attention We incorporate a spatial attention network `sa` to enhance the outcome of the convolutional adapter `ca`. Whereas `ca` focuses on capturing channel-wise information, `sa` can apprehend the spatial information among adjacent elements in the feature map. The `sa` module initially compresses the feature map into a single channel and then computes element-wise attention weights.

We implement the channel-wise compression similarly to [23]. Given a feature map $z \in \mathbb{R}^{C\times H\times W}$, we evaluate both max pooling $\texttt{p}_{\texttt{max}}(z) \in \mathbb{R}^{1\times H\times W}$ and average pooling $\texttt{p}_{\texttt{avg}}(z) \in \mathbb{R}^{1\times H\times W}$ across channels. Both pooling features capture the most prominent and the aggregated representations from the feature map. We process the stack of $\texttt{p}_{\texttt{max}}(z)$ and $\texttt{p}_{\texttt{avg}}(z)$ with a 7×7 convolution layer `conv7x7` and a batch normalization `bn` to form a single plane. A sigmoid function σ generates a $H \times W$ attention weight. The proposed `sa` is defined as:

$$\texttt{sa}(z) \coloneqq \sigma(\texttt{bn}(\texttt{conv7x7}([\texttt{p}_{\texttt{max}}(z); \texttt{p}_{\texttt{avg}}(z)]))) \otimes z, \tag{7}$$

where $\otimes$ denotes element-wise multiplication. After scaling the input by this attention weight, the scaled output is re-integrated via the skip-connection as in Eq. 5.

(a) VisDA (b) Office-31 (b) Office-Home

Fig. 3: Sample images from each domain of VisDA (Source, Target), Office-31 (**A**mazon, **D**SLR, **W**ebcam) and Office-Home (**Ar**t, **Cl**ipart, **Pr**oduct, **Re**al World) benchmark datasets.

5 Experiment

5.1 Dataset

We evaluate the proposed method on popular UDA benchmark datasets: VisDA, Office-31 and Office-Home. Fig. 3 shows sample images from each domain of these datasets.

VisDA. VisDA [17] is a synthetic-to-real dataset. The source domain contains 152,397 synthetic images of 3D models of 12 categories rendered from different angles and lighting conditions, whereas the target domain contains 55,388 photo-real images. Its large scale makes VisDA one of the most challenging UDA tasks.

Office-31. Office-31 [20] includes 4,110 images of 31 categories collected from three distinct domains: Amazon images (A), DSLR photos (D), and Webcam photos (W). We consider all the six adaptation tasks.

Office-Home. Office-Home [22] consists of 15,587 images of 65 categories from four visually very different domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Re). With Office-Home, we take each of four domains as target domain, while combining the rest to form source domain. We repeat this multi-source adaptation task four times in turn.

5.2 Implementation Details

Network Architecture. In all experiments, we choose to take ResNets [4] as backbone architectures. In particular, we use the ResNet-101 and ResNet-50 networks pretrained on ImageNet as in prior works [7, 13, 21]. We treat all layers but the final linear layer of ResNet as a feature generator network g. For a head classifier network h, we employ three linear layers with hidden layers of 1,000 neurons. The model accepts images in VisDA, Office-31 and Office-Home as input after each image is resized to 224×224 and normalized using ImageNet statistics.

Both ResNet-101 and ResNet-50 consist of four blocks of multiple residual layer stacks. We append a CASA network to each of four blocks. The number of parameters increased by the four CASA networks is less than 3.3% of the original source model. For the adapter bottleneck of the CASA network, the ratio factor r is set to $\frac{1}{8}$ (ResNet-101) or $\frac{1}{16}$ (ResNet-50). The dropout rate is set to 0.5.

Source Model Training. We present the details of source model training for reproducibility. Note that the source images are only available at the source training period

and are not available for model adaptation. During training source models, we have randomly augmented the source images. We have also applied label smoothing [16] following [13]. The source models are optimized with SGD and validated with randomly spared train samples. For all datasets, the cosine learning rate decay is used with the initial learning rate of 10^{-3}. For each multi-source task with Office-Home, we combine all the source domains to train a source model, similarly to [18, 26].

Target Model Training. During adapting the model to the target domain, each baseline UDA method freezes the head classifier h and learns the feature generator g. On the other hand, MALTS keeps both h and g of the backbone network frozen and updates only the appended CASA networks.

We optimize with SGD (momentum 0.9) for all adaptation tasks. We employ cosine decay schedules with the initial learning rate of 5.0×10^{-4} for VisDA and 2.0×10^{-4} for Office-31 and Office-Home. With FAUST, the sharpening temperature for pseudo-labeling is set to 0.025. The networks are trained for 10K iterations on VisDA and 20K iterations on Office-31 and Office-Home, respectively. The mini-batch size is set to 64. In our experiment, every test dataset across the target domains conforms to the transductive setting whereas 1% of held-out test dataset are used for measuring source data accuracy in Table 4. We report the average accuracy from three independently sampled target domain data.

5.3 Result

VisDA We start by training the baseline UDA methods with the all target samples in VisDA. Both SHOT-IM and FAUST improve the source-only model more than 24%. Whereas the source model accuracy on the target data is 56.6%, they respectively achieves the accuracies of 81.0% and 84.7%.

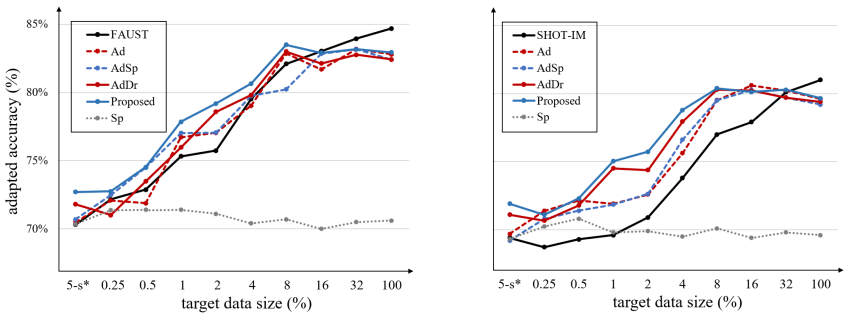
However, the baseline methods underperform if target data is less available. To investigate the impact of target data volume on their performance, we initially reduce the target data size to 32% of the whole dataset and successively halve the size down to 0.25%. The results are presented in Fig. 1 and Table 1. The performance of SHOT-IM and FAUST sharply declines as the target data size decreases. With the reduced target data, the accuracy drops more than 12%: from 81.0% to 68.7% for SHOT-IM, and from 84.7% to 72.2% for FAUST. As the data becomes insufficient to learn the target domain, the baseline UDA methods suffer from overfitting and bad generalization.

Next, we evaluate the performance of the proposed method with varying numbers of target samples. MALTS preserves the source model and only updates the appended CASA networks during model adaptation. We apply MALTS in combination with the baseline UDA methods: SHOT-IM+MALTS and FAUST+MALTS. As compared in Fig. 1 and Table 1, MALTS consistently outperforms the baselines when the target data volume is small. When only 2% of target samples are used, we observe a significant increase in accuracy: 4.8% over SHOT-IM and 3.5% over FAUST. SHOT-IM+MALTS achieves up to 5.4% improvement with 1% target samples. This result shows that with small target sizes MALTS is more robust to overfitting.

Only with sufficiently many target samples is MALTS surpassed by the baselines. In high-resource setting, UDA methods are less susceptible to overfitting and the model

Table 1: Classification accuracy (%) on VisDA (ResNet-101). We report the mean accuracy of three independent runs in each setting. In 5-shot setting, five images are sampled from each category. In other settings, images are sampled uniformly, maintaining class imbalance. The number in parentheses indicates the average number of samples per category. The accuracy of the source-only model on the target data is 56.6%. The best results are in **bold**. (Ad: adapter, Sp: spatial attention, Dr: dropout)

Methods	Ad	Sp	Dr	5-shot	0.25% (11.5)	0.5% (23.1)	1% (46.2)	2% (92.3)	4% (184.6)	8% (369.3)	16% (738.5)	32% (1477.0)	100% (4615.7)
FAUST [7]				70.3	72.2	72.9	75.3	75.7	79.5	82.1	**83.0**	**83.9**	**84.7**
FAUST+MALTS (proposed)	✓	✓	✓	**72.7**	**72.8**	**74.5**	**77.8**	**79.2**	**80.7**	**83.5**	82.9	83.2	82.9
Configuration 1 (Ad)	✓			71.8	72.1	71.9	76.7	77.0	79.0	82.9	81.7	83.2	82.8
Configuration 2 (AdSp)	✓	✓		70.7	72.5	74.5	77.0	77.1	79.8	80.2	82.8	83.2	82.4
Configuration 3 (AdDr)	✓		✓	70.5	71.0	73.5	76.0	78.6	79.8	83.0	82.1	82.8	82.4
Configuration 4 (Sp)		✓		70.4	71.4	71.4	71.4	71.1	70.4	70.7	70.0	70.5	70.6
SHOT-IM [13]				69.4	68.7	69.3	69.6	70.9	73.8	77.0	77.9	80.1	**81.0**
SHOT-IM+MALTS (proposed)	✓	✓	✓	**71.9**	71.1	**72.3**	**75.0**	**75.7**	**78.8**	**80.4**	80.1	**80.3**	79.7
Configuration 1 (Ad)	✓			69.7	**71.4**	72.1	71.9	72.6	75.6	79.5	**80.6**	80.2	79.6
Configuration 2 (AdSp)	✓	✓		69.2	70.8	71.4	71.9	72.6	76.6	79.5	80.3	79.7	79.2
Configuration 3 (AdDr)	✓		✓	71.1	70.7	71.8	74.5	74.4	77.9	80.3	80.2	79.7	79.4
Configuration 4 (Sp)		✓		69.3	70.2	70.8	69.8	69.9	69.5	70.1	69.4	69.8	69.6



(a) FAUST+MALTS on VisDA (b) SHOT-IM+MALTS on VisDA

Fig. 4: (Best viewed in color) FAUST+MALTS and SHOT-IM+MALTS results on VisDA. 5-s* indicates the 5-shot setting.

capacity matters more. Because the baseline UDA methods can update more model parameters, they work better in the full target data setting. On the other hand, MALTS shows a slight decrease in accuracy with more than 8% target samples due to lack of model capacity. Nonetheless, MALTS achieves competitive results with 96.7% less tunable parameters; thus, requiring less training time and GPU memory.

In addition, we explore an extreme case where only five samples per category are available. This poses a challenging low-resource adaptation problem where only 60 target domain images are used to learn the domain-specific task. In this 5-shot setting, MALTS significantly surpasses the baselines. The accuracy goes from 70.3% to 72.7% when MALTS is combined with FAUST, and from 69.4% to 71.9% when combined with SHOT-IM. We also find that the performance of MALTS in the 5-shot setting is similar to (FAUST+MALTS) or even higher than (SHOT-IM+MALTS) that of the

Table 2: Classification accuracy (%) on Office-31 (ResNet-50).

Methods	5-shot	30% (10.0)	60% (20.0)	100% (33.4)
A→D (source only: 68.4)				
FAUST [7]	**87.3**±0.8	86.7±1.5	86.3±0.6	**86.5**±0.6
FAUST+MALTS (proposed)	**87.3**±0.4	**86.8**±0.7	**87.1**±0.3	85.6±0.3
A→W (source only: 68.9)				
FAUST [7]	85.6±0.6	85.4±0.1	86.4±0.3	**88.8**±0.2
FAUST+MALTS (proposed)	**85.8**±0.9	**87.6**±1.0	**89.3**±0.3	87.6±0.3
D→A (source only: 62.5)				
FAUST [7]	66.0±0.6	72.0±0.7	**74.5**±0.5	**73.2**±0.5
FAUST+MALTS (proposed)	**66.7**±0.4	**72.2**±0.3	**74.5**±0.2	71.6±0.0
D→W (source only: 96.7)				
FAUST [7]	95.9±0.2	96.6±0.2	**97.4**±0.5	**98.0**±0.0
FAUST+MALTS (proposed)	**96.4**±0.3	**97.1**±0.4	**97.4**±0.3	97.1±0.0
W→A (source only: 60.7)				
FAUST [7]	64.5±0.9	**71.0**±0.6	72.0±0.4	**71.2**±0.6
FAUST+MALTS (proposed)	**65.7**±0.4	70.9±0.9	**73.1**±0.1	70.4±0.4
W→D (source only: 99.3)				
FAUST [7]	99.6±0.2	99.3±0.3	99.1±0.2	**99.6**±0.3
FAUST+MALTS (proposed)	**99.9**±0.1	**99.5**±0.2	**99.4**±0.2	99.5±0.1
Average (source only: 76.1)				
FAUST [7]	83.2	85.2	85.9	**86.2**
FAUST+MALTS (proposed)	**83.6**	**85.7**	**86.8**	85.3

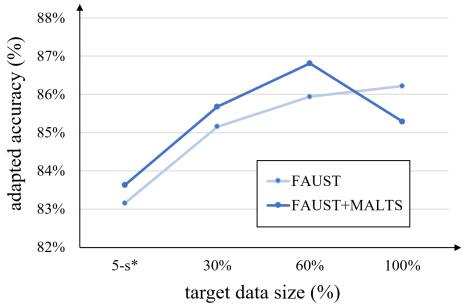


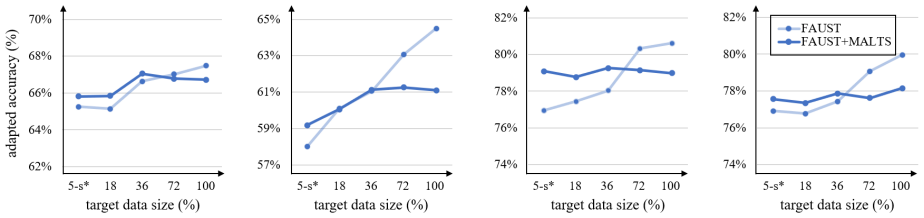
Fig. 5: Average performance of FAUST and FAUST+MALTS on Office-31.

0.25% target size setting. The gap between MALTS and the baseline widens as well. Whereas the 5-shot case uses less than half as many target samples as the 0.25% case, the number of samples across classes is kept the same. This finding suggests that balancing class-wise sample size can positively affect our method when target domain samples are scarce.

Office-31 We also conduct experiments with Office-31. On all six one-to-one domain adaptation tasks, we compare FAUST with FAUST+MALTS. In contrast to VisDA, each domain of Office-31 contains only 33.4 images per class on average. Given the very limited number of samples, we examine three target sizes: 100%, 60%, and 30%. We also assess the 5-shot setting.

Table 3: Classification accuracy (%) on Office-Home (ResNet-101).

Methods	5-shot	18% (10.8)	36% (21.6)	72% (43.2)	100% (60.0)
Cl,Pr,Re→Ar (source only: 64.5)					
FAUST [7]	65.3±0.8	65.1±0.6	66.6±0.4	**67.0**±0.4	**67.5**±0.2
FAUST+MALTS (proposed)	**65.8**±0.4	**65.8**±1.7	**67.1**±0.5	66.8±0.6	66.7±0.3
Ar,Pr,Re→Cl (source only: 53.9)					
FAUST [7]	58.0±0.8	**60.1**±0.4	**61.1**±0.3	**63.1**±0.4	**64.5**±0.4
FAUST+MALTS (proposed)	**58.6**±1.0	**60.1**±0.8	**61.1**±0.7	61.3±0.5	61.1±0.5
Ar,Cl,Re→Pr (source only: 76.3)					
FAUST [7]	77.0±1.4	77.5±1.4	78.0±0.5	**80.3**±0.6	**80.6**±0.9
FAUST+MALTS (proposed)	**79.1**±1.7	**78.8**±1.0	**79.3**±0.7	79.2±0.2	79.0±0.5
Ar,Cl,Pr→Re (source only: 76.2)					
FAUST [7]	76.9±0.5	76.8±0.9	77.4±0.6	**79.1**±0.2	**80.0**±0.1
FAUST+MALTS (proposed)	**77.6**±0.6	**77.4**±1.0	**77.9**±0.4	77.6±0.3	78.2±0.1



(a) Cl, Pr, Re → Ar (b) Ar, Pr, Re → Cl (c) Ar, Cl, Re → Pr (d) Ar, Cl, Pr → Re

Fig. 6: Comparison between FAUST and FAUST+MALTS on four mutli-source tasks using Office-Home.

The average performance clearly shows similar patterns to VisDA as presented in Fig. 5 and Table 2. The performance of FAUST declines as fewer target samples are available. MALTS consistently outperforms the baseline over different target data sizes only to be surpassed by FAUST in the full target data setting. Particularly on the task A→W, MALTS improves the baseline accuracy by 2.9% using 60% target samples.

On A→D and W→D, we notice that MALTS performs better in the 5-shot setting than in the 30% target size setting. Because DSLR exhibits high class-imbalance (see Appendix), we posit that class-wise balance in the 5-shot setting can be beneficial to performance. Plots of individual tasks can be found in Appendix.

Office-Home We extend the proposed method to multi-source domain adaptation tasks. On four multi-source tasks in Office-Home, we compare FAUST with FAUST+MALTS. These tasks are challenging because images from four domains exhibit significantly different visual styles as can be seen in Fig. 3. In addition, each domain has a small number of samples per class, 60 images on average.

Similarly to the experiments with VisDA and Office-31, we investigate the adaptation performance with respect to various target domain data sizes. We consider four different settings: 100%, 72%, 36% and 18%. The respective per class sample sizes are 60, 43.2, 32.6 and 10.8. The 5-shot setting is included as well. Fig. 6 and Table 3 present the experiment results.

(a) Art (b) Clipart (c) Product (d) Real-World

Fig. 7: Four Office-Home domains are visualized using t-SNE. Feature embeddings of 1,000 samples are obtained from a pretrained ResNet-101 backbone network. Clipart exhibits a distinctive feature distribution different from other domains.

Table 4: Source data accuracy (%) of the adapted models. MALTS consistently shows higher accuracy than FAUST.

Model	VisDA	A→D	A→W	D→A	D→W	W→A	W→D	$\mathcal{R}\rightarrow$ Ar	$\mathcal{R}\rightarrow$ Cl	$\mathcal{R}\rightarrow$ Pr	$\mathcal{R}\rightarrow$ Re
Source Model	**94.5**	**88.6**	88.0	**100.0**	**100.0**	**99.3**	**100.0**	**86.4**	86.3	82.9	**85.7**
FAUST [7]	78.0	85.9	85.3	93.3	**100.0**	92.0	99.3	82.1	76.3	77.4	82.4
FAUST+MALTS (proposed)	85.4	88.3	**88.8**	99.6	99.6	**99.3**	**100.0**	**86.4**	**86.6**	**83.2**	85.1

Table 5: The number of trainable parameters, training time and memory usage. FAUST trains the parameters in the ResNet-101 backbone network. FAUST+MALTS trains the parameters in the CASA networks, while fixing the parameters in the backbone network.

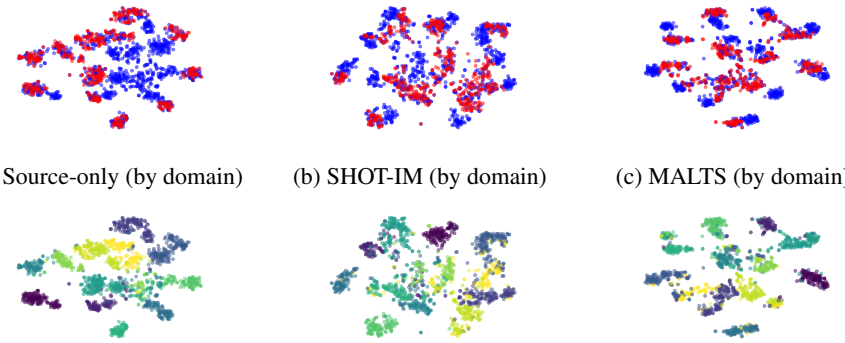
Method	Trainable Param.	Training Time	GPU Memory
FAUST [7]	42.7 million	193 minutes	20.0 GB
FAUST+MALTS (proposed)	1.4 million	157 minutes	16.5 GB

In all multi-source tasks, we observe the trends that are consistent with the VisDA and Office-31 experiments. The accuracy of the baseline method (FAUST) decreases as the target data size is reduced, whereas FAUST+MALTS shows a more gradual decline. FAUST+MALTS is more robust to overfitting and achieves better results when the target data volume is small. As more target samples become available, FAUST outperforms FAUST+MALTS. The crossing occurs when the target data size is between 36% and 72% of the full dataset.

The task when Clipart is the target domain generally shows lower accuracy than other tasks. Clipart exhibits a significant divergence from other domains as illustrated in Fig. 3 and Fig. 7. This severe distributional discrepancy can be addressed by increasing model capacity. In MALTS, we can control the model capacity by changing the ratio factor r of the CASA network. We analyze the effects of r in Sec. 5.4.

5.4 Analysis and Discussion

Source Domain Forgetting We verify that the proposed method is effective to the forgetting issue. After both FAUST and MALTS fully adapt the source model to the target domain, their classification accuracies are evaluated on the source domain test data. In Table 4, source model is also compared. We observe that FAUST forgets the source



(a) Source-only (by domain) (b) SHOT-IM (by domain) (c) MALTS (by domain)

(d) Source-only (by class) (e) SHOT-IM (by class) (f) MALTS (by class)

Fig. 8: (Best viewed in color) Feature embeddings in the VisDA adaptation task with 1% target data are visualized. Source-only model, SHOT-IM and MALTS are compared. **(a), (b) and (c):** Red and blue dots are the source and the target samples of VisDA, respectively. **(d), (e) and (f):** Each color represents a different class.

domain and undergoes a substantial loss in source data accuracy. However, MALTS shows consistently better source accuracy than FAUST after different adaptation tasks. The performance of MALTS is close to or even slightly higher than that of the source model. This result shows that MALTS can preserve the knowledge learned from the source domain while it learns new knowledge from the target domain.

Training Time and Memory Usage The proposed method has much less tunable parameters than a baseline method (3.3%). Performing UDA with MALTS results in 18.7% reduction in training time and 17.5% decrease in GPU memory usage compared to FAUST as shown in Table 5. These results are measured using PyTorch implementation on an A100 GPU. Full VisDA target data is used.

Ablation Study We evaluate the contribution of each component of a CASA network. The detailed evaluation is shown in Table 1 and Fig. 4. Ad and AdSp configurations (red and blue dashed lines) slightly improve the full adaptation baselines with a few exceptions. Their results show some degree of fluctuation. Incorporating dropout (AdDr, red solid line) stabilizes this fluctuation, yet with a limited performance boost. However, adding Sp to this configuration (proposed, blue solid line) leads to a significant performance gain. Employing Sp alone (dotted line) yields a near constant accuracy far below others as the target data size varies.

Feature Visualization We visualize the feature embeddings of VisDA task in 1% target data setting. SHOT-IM is used as a baseline. The output of the last pooling layer of the feature generator g is plotted using t-SNE [14]. SHOT-IM+MALTS shows more compact feature clusters and leaves less number of samples unaligned in both source and target domains (Fig. 8e and Fig. 8f) compared to SHOT (Fig. 8c and Fig. 8d). This observation is consistent with the experiment results.

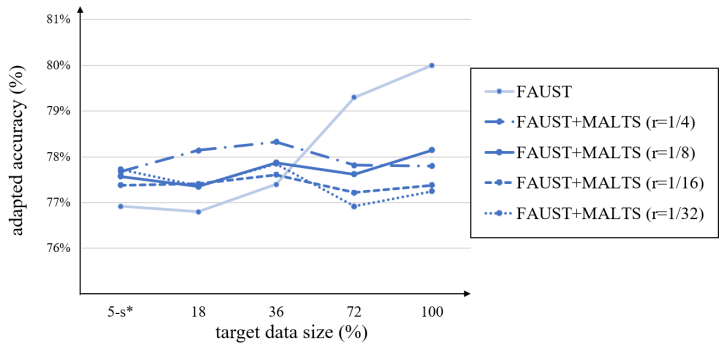


Fig. 9: The effect of bottleneck size r in the CASA module (Ar, Cl, Pr $\rightarrow$ Re).

CASA Parameters Given an input feature map with C channels and a ratio factor r, each CASA network has $(rC^2 + rC) + (rC^2 + C) + 2 \times 7^2 = 2rC^2 + (r+1)C + 98$ target-specific parameters. The 1×1 convolutional layers has a bias. As compared in Table 5, the number of tunable parameters in CASA ($r = \frac{1}{8}$) networks amounts to 3.3% of the entire parameters in the ResNet-101 backbone network g. With ResNet-50, $r = \frac{1}{16}$ comprises a similar proportion of tunable parameters (3.0%).

Ratio Factor We study the impact of the ratio factor r. Fig. 9 shows the multi-source task where Real-World is the target domain. MALTS demonstrates a small fluctuation in performance as the target data size varies. Fig. 9 illustrates that $r = \frac{1}{8}$ generally yields superior performance across different settings except the 5-shot case. In the 5-shot setting, $r = \frac{1}{32}$ performs slightly better. MALTS with varying r values exhibits quite similar patterns, and the performance gap remains within 0.9%. We can increase the model capacity with larger r to improve the adaptation performance.

6 Conclusion

In this paper, we introduce MALTS, a novel approach to address model overfitting in UDA tasks where the source data is unavailable and the target data is scarce. Our method incorporates a network called CASA into the frozen pretrained source model. CASA is a lightweight network with a small number tunable parameters. MALTS adapts only these parameters to the target domain. We compose CASA with a convolutional adapter with dropout followed by a spatial attention network. Empirical evaluations reveal that our approach consistently outperforms existing UDA methods across various target-scarce UDA tasks while demanding less computation resource. Our method also improves generalization of the adapted model by mitigating source domain forgetting issues and regulating the model to better control the overfitting in many data-limited scenarios.

Acknowledgements This work was supported in part by Research Program of Seoul National University of Science and Technology, Republic of Korea.

References

1. Chang, W.G., You, T., Seo, S., Kwak, S., Han, B.: Domain-specific batch normalization for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7354–7362 (2019)
2. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4), 128–135 (1999)
3. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain adversarial training of neural networks. Journal of Machine Learning Research **17(59)**, 1–35 (2016)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
5. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International Conference on Machine Learning. pp. 2790–2799. PMLR (2019)
6. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=nZeVKeeFYf9
7. Lee, J., Lee, G.: Feature alignment by uncertainty and self-training for source-free unsupervised domain adaptation. Neural Networks **161**, 682–692 (2023). https://doi.org/https://doi.org/10.1016/j.neunet.2023.02.009
8. Lee, J., Lee, G.: Unsupervised domain adaptation based on the predictive uncertainty of models. Neurocomputing **520**, 183–193 (2023). https://doi.org/https://doi.org/10.1016/j.neucom.2022.11.070
9. Li, J., Du, Z., Zhu, L., Ding, Z., Lu, K., Shen, H.T.: Divergence-agnostic unsupervised domain adaptation by adversarial attacks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 8196–8211 (2022). https://doi.org/10.1109/TPAMI.2021.3109287
10. Li, S., Xie, M., Gong, K., Liu, C.H., Wang, Y., Li, W.: Transferable semantic augmentation for domain adaptation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11516–11525 (2021)
11. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597 (2021). https://doi.org/10.18653/v1/2021.acl-long.353
12. Li, Z., Hoiem, D.: Learning without forgetting. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 2935–2947 (2016)
13. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of International Conference on Machine Learning (ICML). pp. 6028–6039 (2020)
14. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
15. Mirza, M.J., Micorek, J., Possegger, H., Bischof, H.: The norm must go on: Dynamic unsupervised domain adaptation by normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14765–14775 (2022)
16. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Advances in Neural Information Processing System (NeurIPS). vol. 32, pp. 4694–4703 (2019)

17. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: VisDA: The visual domain adaptation challenge. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop (2016)
18. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Moment matching for multi-source domain adaptation. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1406–1415 (2019)
19. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. In: Advances in neural information processing systems. vol. 30, pp. 506–516 (2017)
20. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domain. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 213–226 (2010)
21. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3723–3732 (2018)
22. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5018–5027 (2017)
23. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
24. Xia, H., Zhao, H., Ding, Z.: Adaptive adversarial network for source-free domain adaptation. In: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9010–9019 (2021)
25. Xu, R., Liu, P., Wang, L., Chen, C., Wang, J.: Reliable weighted optimal transport for unsupervised domain adaptation. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4393–4402 (2020)
26. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain adaptive ensemble learning. IEEE Transactions on Image Processing **30**, 8008–8018 (2021)