# Adaptive Feature Swapping for Unsupervised Domain Adaptation

Anonymous Author(s)
Submission Id: 910*

## ABSTRACT

The bottleneck of visual domain adaptation always lies in the learning of domain invariant representations. In this paper, we present a simple but effective technique named Adaptive Feature Swapping for learning domain invariant features in Unsupervised Domain Adaptation (UDA). Adaptive Feature Swapping aims to select semantically irrelevant features from labeled source data and unlabeled target data and swap these features with each other. Then the merged representations are also utilized for training with prediction consistency constraints. In this way, the model is encouraged to learn representations that are robust to domain-specific information. We develop two swapping strategies including channel swapping and spatial swapping. The former encourages the model to squeeze redundancy out of features and pay more attention to semantic information. The latter motivates the model to be robust to the background and focus on objects. We conduct experiments on object recognition and semantic segmentation in UDA setting and the results show that Adaptive Feature Swapping can promote various existing UDA methods.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

## KEYWORDS

Visual Domain Adaptation, Domain Adaptive Semantic Segmentation, Transfer Learning, Object Recognition

**ACM Reference Format:**

Anonymous Author(s). 2023. Adaptive Feature Swapping for Unsupervised Domain Adaptation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

Domain adaptation improves a target task with insufficient annotations by knowledge transfer from a source domain with rich annotations. To achieve reliable transfer, the domain discrepancy between the source and target domains should be mitigated in domain adaptation. There are two main directions for mitigating domain discrepancy including moment alignment [11, 19, 28, 30, 42, 56] and adversarial training [8, 29, 40, 59], based on popular assumptions such as Covariate Shift [41]. The former aims to minimize domain discrepancy estimated by maximum mean discrepancy (MMD) [28], correlation distance [42] or other distance metric [19, 30] calculated on task-specific features. The latter direction of methods learn features that are indistinguishable from an additional domain discriminator [8, 29] or utilizes adversarial learning between two classifiers for aligning conditional feature distributions [21, 40, 59, 60].

Recently, domain-specific modeling and perturbation-tolerant learning in domain adaptation have drawn considerable attention. Modeling the domain-specific features enhances the learning of domain invariant features [1, 4, 9] for UDA. For example, Bousmalis et al. [1] proposed to learn domain-specific features via time-consuming reconstruction at the pixel level and Cui et al. [4] adopted a single fully connected layer whose responses are restricted to vanish. By introducing auxiliary tasks like reconstruction or vanishing bridge responses, these methods learn the domain-specific features in implicit ways resulting in suboptimal solutions with additional computation overhead.

Perturbation-tolerant learning imposes robustness over input perturbations for enhancing the generalization ability of trained models. Such robustness can be measured by the norm of the input-output Jacobian of the network, and it correlates well with generalization [34]. To enhance the generalization in the target domain, several methods [6, 39, 52] imposed consistency between the predictions of two perturbed samples over a target sample with random data augmentations. With expansion assumption, Wei et al. [50] proved that enforcing locally consistent prediction provides accuracy guarantees on unlabeled target data for unsupervised domain adaptation. Typical consistency constraints include L2 distance minimization [6], agreement maximization [39] and mutual information maximization [52]. These methods usually require two complete time-consuming forward computations of deep network. Besides, the underlying information of samples from the two domains for generating perturbation is not well explored. Perturbation based on usual random augmentation may not capture the structure of cross-domain data and restricts the generalization of trained model.

To alleviate aforementioned issues, we present Adaptive Feature Swapping (AFS) to model the domain-specific features and acquire sample perturbation more efficiently and effectively. The key insight is to swap semantically irrelevant features between source and target data, obtaining perturbed representations that maintain the semantic information. By encouraging the consistency between predictions of the representations after swapping and the original ones, the model attains robustness that benefits better generalization and more reliable knowledge transfer. Enforcing the model to predict the ground truth label over the source representations after swapping encourages the model to pay more attention to the domain invariant features against the domain-specific ones. Besides, AFS supports accuracy guarantees on unlabeled target data as locally consistent prediction is ensured [50].

We develop two kinds of Adaptive Feature Swapping techniques from spatial and channel perspectives to facilitate UDA. For Spatial


*Conference acronym 'XX, June 03–05, 2023, Woodstock, NY*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Feature Swapping, since the irrelevant information like the background surrounding an object should not affect the prediction of the object, we attempt to swap the irrelevant features of source samples with those of target samples. To enable reliable Spatial Feature Swapping, we extract the spatial attention to locating the irrelevant spatial features. We adopt activation-based attention [20] due to its computation efficiency, and the features with less attention are regarded as irrelevant ones. Initially, the model may mistakenly put more attention to some semantically irrelevant features. However, as the model is optimized during training, the attention is promoted to better locate the irrelevant features, which makes Spatial Feature Swapping more effective.

Sharing the same spirit of Spatial Feature Swapping, Channel Feature Swapping is conducted to squeeze the redundancy out of features, so the model tends to pay more attention to the semantic information. Similarly, the accumulation of feature responses of samples in a mini-batch is regarded as channel-wise attention, serving to select irrelevant features for swapping. Since the two feature swapping strategies encourage the model to capture semantic information from different perspectives, they could cooperate well with each other which is verified in our experiments. Moreover, the representation with feature swapping randomly integrate feature from samples of other domain, exhibiting more diverse perturbations for domain adaptation task.

We evaluate our AFS on several benchmarks for object classification and semantic segmentation in UDA setting. Experimental results show that the proposed Adaptive Feature Swapping improves source-only model by a considerable margin. Besides, as a lightweight technique, Adaptive Feature Swapping can be easily plugged into various DA methods and boost their performances.

## 2 RELATED WORK

The unsupervised domain adaptation methods can be mainly divided into moment alignment based methods [19, 28, 30, 42, 56] and adversarial training based methods [8, 29, 40, 59]. Moment alignment based methods try to estimate the domain discrepancy related to distribution moments over deep representations. [45] firstly adopted the MMD [10] in the deep neural network and later DAN [28] utilized the multiple kernel variant of MMD over multiple layer representations for better moment matching. CAN [19] proposed to optimize a new metric that explicitly models the intra-class and the inter-class domain discrepancies. Enhanced Transport Distance (ETD) [24] builds an attention-aware transport distance, which can be viewed as the prediction feedback of the iteratively learned classifier, to measure the domain discrepancy.

Early Adversarial training based methods introduced a domain discriminator [8, 29] that is trained to distinguish source and target representations while the backbone network is trained to generate representations to fool the domain discriminator. DANN [8] performed domain adversarial learning on features while [29] performed domain adversarial learning on the multilinear mapping of feature and classifier response. Recent methods [21, 40, 59, 60] utilized adversarial learning between two classifiers for aligning conditional feature distributions. In [59], the authors used two asymmetrical classifiers to estimate the conditional feature distributions with marginal loss. In [51], the authors propose to make the domain alignment proactively serve classification via feature decomposition and alignment with the prior knowledge induced from the classification task.

Other methods focus on some characteristics of specific layers in a deep neural network for domain adaptation such as adaptively increasing the feature norms in AFN [53], maintaining both discriminability and diversity in BNM [3] and minimizing inter-class confusion of unlabeled target data in MCC [18]. Besides, Bousmalis et al. [1] developed a split model that models the shared domain invariant features, and domain-specific features supporting effective reconstruction. Gong et al. [9] proposed a domain flow generation model to generate a continuous sequence of intermediate domains with various domain-specific information. Both methods require time-consuming reconstruction at the pixel level. Recently, Cui et al. [4] proposed to use a single fully connected layer, named gradually vanishing bridge (GVB) to capture domain-specific features. Dong et al. [5] developed a novel Knowledge Aggregation-induced Transferability Perception (KATP) module to distinguish transferable or untransferable knowledge across domains. He et al. [13] proposed Domain-specific Conditional Jigsaw Adaptation Network which simultaneously encourages the network to extract transferable and discriminative features. These methods require careful module design, while our method learns domain-specific features via simple feature swapping with prediction consistency constraints.

For the perturbation-tolerant learning, two random data augmentations are performed over a target sample and the consistency between the predictions of the two augmented samples is imposed in [6]. In [39], the authors proposed to minimize Min-Entropy Consensus loss that univocally selects a pseudo-label that maximizes the agreement between two perturbed versions of the same target sample. In [52], the authors explicitly maximized the mutual information between the rotated image and the label. Our method differs from these methods by implementing the perturbation via feature swapping. The other related work is [49] which exchanged channel activations between the two convolutional layer activations extracted from two modalities of a sample for better fusing the two modalities. [49] selected feature according to the magnitude of Batch-Normalization scaling factor. On the contrary, our method swaps both **channel and spatial** features from two **independent samples** according to carefully designed **aggregated attention**.

## 3 METHODOLOGY

We illustrate our method in object recognition task under unsupervised domain adaptation (UDA) setting for simplicity. Suppose that there are a source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ of $n_s$ labeled samples and a target domain $D_t = \{(x_i^t)\}_{i=1}^{n_t}$ of $n_t$ unlabeled samples. The two domains distribute differently that the input marginal distributions $p(x)$ vary. In UDA, both the two domains share the same $K$ categories. The goal is to learn a domain invariant model that generalizes well in target domain.

We propose Adaptive Feature Swapping for learning domain invariant representations. As shown in Fig. 1, Adaptive Feature Swapping involves spatial swapping and channel swapping. Spatial swapping aims to select irrelevant features like background for source and target samples and swap these features with each other. Similarly, channel swapping selects irrelevant features along

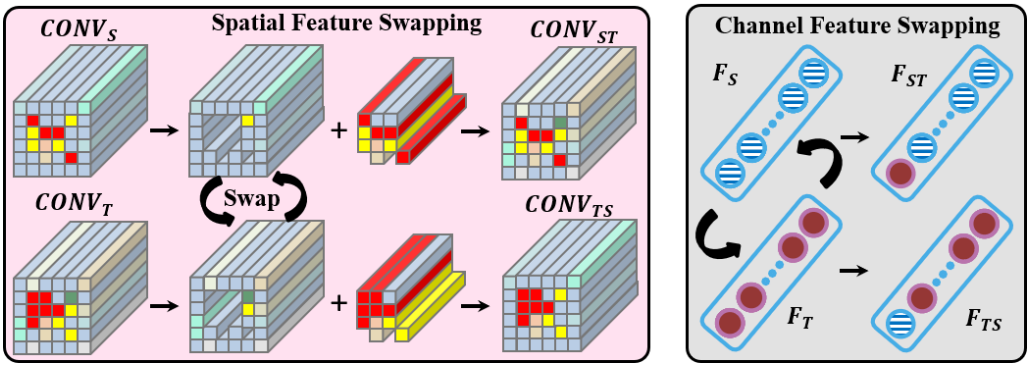


**Figure 1: Illustration of Spatial Feature Swapping and Channel Feature Swapping. For the convolutional activations of a source (target) sample, the semantic features (in red and yellow) are kept and combined with the rest irrelevant features from convolutional activations of a target (source) sample, obtaining new convolutional activations. The Channel Feature Swapping exchanges some irrelevant elements between source and target features.**

the channel and swaps these features between source and target samples. The prediction over the original feature and the one after swapping is enforced to be consistent. Both feature swapping strategies encourage the model to pay more attention to the semantic information that is invariant across domains.

## 3.1 Spatial Feature Swapping

We perform Spatial Feature Swapping on a specific convolutional layer activations. Suppose that for a batch of source samples and a batch of target samples both with size B, their convolutional layer activations are $CONV_S \in \mathcal{R}^{B\times C\times W\times H}$ and $CONV_T \in \mathcal{R}^{B\times C\times W\times H}$. To perform feature swapping, we need to determine which spatial feature is semantically irrelevant. Inspired by activation-based attention [20], we first map the batch of convolutional layer activations into an attention map and then select features with less attention for swapping. Concretely, the spatial attention $M_{sp}(A)$ for an activation tensor $A$ is obtained via

$$(M_{sp}(A))_{i,j} = \sum_{b=1}^{B}\sum_{c=1}^{C} |A_{b,c,i,j}|^p, \tag{1}$$

where $i \in \{1, 2, ..., H\}$, $j \in \{1, 2, ..., W\}$ are spatial indexes and $p$ is set to 1. $b$ and $c$ are the instance index and the channel index, respectively.

To prevent a semantically relevant feature in $CONV_S$ to be swapped to $CONV_T$ that affects the semantic of swapped $CONV_{TS}$, we select features corresponding to the least $r_1$ percentage of aggregated attention $(M_{sp}(CONV_S) + M_{sp}(CONV_T))$ for swapping.

## 3.2 Channel Feature Swapping

In CNN-based classifier, the feature before the final classifier layer is usually of high dimension that involves redundancy. The network may tend to remember irrelevant features for recognition, especially for a small scale labeled source domain. This issue motivates us to adopt Channel Feature Swapping for encouraging the model to ignore semantically irrelevant information.

Suppose that $F_S \in \mathcal{R}^{B\times L}$ and $F_T \in \mathcal{R}^{B\times L}$ are the features before the final classifier for a batch of source samples and a batch of target samples both with size B. We construct channel attention to determine which channel feature is semantically irrelevant. Channel attention can be modeled in several ways like global average pooling (GAP), global max/min pooling, or learned via FC layers [17] or scaled transformation [7]. Among these solutions, GAP is simple yet effective and widely used in NIN [26], GoogLeNet [43], and ResNet [12]. Therefore, we use GAP to extract channel attention for selecting redundant features. Specifically, given an activation tensor $V \in \mathcal{R}^{B\times L}$, the channel attention can be computed as

$$(M_{ch}(V))_c = \sum_{i}^{B} |V_{i,c}|, \tag{2}$$

where $i \in \{1, 2, ..., B\}$ and $c$ is the channel index.

Similar to Spatial Feature Swapping, we need to select channels that contribute little to predicting objects. Specifically, we construct the aggregated attention as $(M_{ch}(F_S) - M_{ch}(F_T))$ assuming that the channel that the model pays nearly equal attention for both domains delivers less semantic information. Then the aggregated attention is sorted and the features corresponding to the least $r_2$ percentage of aggregated attention are selected for swapping.

## 3.3 Learning with Feature Swapping

In unsupervised domain adaptation, a model is trained by minimizing a classification loss on the labeled source domain and an additional transfer loss, to learn both discriminative and domain invariant representations. The classification loss in $\mathcal{D}_S$ can be computed as:

$$\mathcal{L}_{cls} = \frac{1}{N_s}\sum_{i=1}^{N_s} L_{ce}(p_i^s, y_i^s), \tag{3}$$

where $L_{ce}$ is the cross-entropy loss function and $p_i^s$, $y_i^s$ are classification response and label for the source sample $x_i^s$.

The additional transfer loss can be norm increasing loss [53], MMD loss [28], MCC loss [18] or adversarial discriminator score [8].

For the model learning with Adaptive Feature Swapping, the transfer loss can be formulated as:

$$\mathcal{L}_{cst} = \frac{1}{N_s}\sum_{i=1}^{N_s} L_{ce}(p_i^{st}, y_i^s) + \lambda_{cst}\frac{1}{N_t}\sum_{j=1}^{N_t} ||p_j^t - p_j^{ts}||_2^2, \tag{4}$$

where $p_i^{st}$ is the classification response over representation of source sample $x_i^s$ that is swapped with some features from target sample. And $p_j^{ts}$ is the classification response over the representation of target sample $x_j^t$ that is swapped with some features from a source sample. $p_j^t$ is the classification response of the original target sample $x_j^t$.

The transfer loss for Adaptive Feature Swapping encourages consistency between model predictions of the original representation and the representation with feature swapping for a sample. For source samples, the consistency is achieved by cross-entropy loss with corresponding labels. As for unlabeled target samples, the consistency is simply achieved by mean square loss.

The total loss of model learning with Adaptive Feature Swapping is

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{cst}. \tag{5}$$

It is worth noting that the proposed Adaptive Feature Swapping can cooperate with existing methods. Suppose that the loss of existing methods is $\mathcal{L}_{cls} + \lambda_{tran}\mathcal{L}_{tran}$, one can easily add $\mathcal{L}_{cst}$ to the original loss obtaining

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_{tran}\mathcal{L}_{tran} + \mathcal{L}_{cst}. \tag{6}$$

### 3.4 Theoretical Insight

We provide the connections between our method and the theory proposed in [50]. Let $F$ denote a learned scoring functions (a neural network), $G$ be the label predictor that $G(x) = \arg\max_i F(x)_i$, and $G^*$ be the ground-truth classifier. Define $\mathcal{B}$ the set of perturbated versions $x'$ for a sample $x$ where the perturbation is the proposed feature swapping. We can see that $||x' - x|| \leq r_*||x||$, where $r_*$ is swapping ratio. The neighborhood of x can be defined as $\mathcal{N} = \{x' : \mathcal{B}(x) \bigcap \mathcal{B}(x') \neq \emptyset\}$. The consistency loss can be formulated as:

$$R_{\mathcal{B}}(G) = \mathbb{E}_P[1(\exists x' \in \mathcal{B}(x) \; such \; that \; G(x') \neq G(x))] \tag{7}$$

which is strongly reminiscent of our consistency loss $\mathcal{L}_{cst}$.

In accordance with [50], we assume P, a distribution of unlabeled examples over input space $\mathcal{X}$, is $\mathcal{B}$-separated with probability $1 - \mu$ by $G^*$ that $R_{\mathcal{B}}(G^*) \leq \mu$. Besides, let $\mathcal{M}(G_{pl} = \{x : G_{pl}(x) \neq G^*(x)\})$ be the set of examples that mistakenly labeled by a pseudolabeler $G_{pl}$. The expansion assumption stated in [50] requires that $P_i(\mathcal{N}(V)) \leq \min(cP_i(V), 1)$ for all $V \subseteq X$ and all class i with $P_i(V) \leq a$ , where $P_i$ is the class-conditional distribution, $a = \max_i\{P_i(\mathcal{M}(G_{pl}))\}$ and $a$ is required to be less that 1/3 and $c > 3$.

Then it is proved in [50] that for any minimizer $\widehat{G}$ of $\min_G \frac{c+1}{c-1}L_{0\text{-}1}(G, G_{pl}) + \frac{2c}{c-1}R_{\mathcal{B}}(G) - Err(G_{pl})$, we have

$$Err(\widehat{G}) \leq \frac{2}{c-1}Err(G_{pl}) + \frac{2c}{c-1}\mu \tag{8}$$

where $L_{0\text{-}1}(G, G') = \mathbb{E}[1(G(x) \neq G'(x))]$, and $Err(G) = L_{0\text{-}1}(G, G^*)$ is the error rate of $G$.

Therefore, our method could achieve the above bounded target error (Eqn.( 8)) for unsupervised domain adaptation under expansion assumption (this assumption is justified by Wei et al. [50] on real-world datasets with BigGAN). Specifically, during the training process, the current model has relatively low $Err(G_{pl})$, which in turn lower the error rate $Err(\widehat{G})$ of the updated model. With stronger base model like MCC [18] and BNM [3], the $Err(G_{pl})$ is lower. Therefore, the trained model gets lower error rate on target domain, as verified experimentally in Section 4.

## 4 EXPERIMENTS

### 4.1 Datasets and Setup

We use two standard object classification benchmarks and a standard semantic segmentation benchmark for unsupervised domain adaptation to evaluate the proposed methods. Office-Home [46] is a challenging benchmark with 65 categories in four domains: *Art* (Ar), *Clipart* (Cl), *Product* (Pr) and *Real-world* (Rw), which contains 12 transfer tasks. VisDA-C [36] is a large-scale benchmark that contains the images from 12 categories of two very distinct domains, the synthetic domain, and the real-world domain. The synthetic domain contains 152,397 images and the latter contains 55,388 images. Following standard protocol, we focus on the synthetic-to-real transfer task.

For the segmentation task, we evaluate our method by adapting the segmentation from game scenes, GTA5 [38] dataset, to real scenes, the Cityscapes [2] dataset. GTA5 contains 24,966 images with the resolution of 1914×1052. The Cityscapes dataset contains 2,975 training images and 500 images for validation with the resolution of 2048×1024.

Following standard protocol, we use the training set of GTA5 as the labeled source domain and the unlabeled training images from Cityscapes as the target domain. We conduct evaluations on the validation set of Cityscapes and adopt the Intersection-over-Union (IoU) of each class and the mean-Intersection-over-Union (mIoU) as performance metrics. For the task GTA5→Cityscapes, we report the results on the common 19 classes.

### 4.2 Implementation Details

**Classification.** The methods are implemented based on PyTorch [35], and for fair comparisons, we use the ResNet-50/ResNet-101 [12] as the backbone of network, fixing the batch-size of 36 for all experiments. We adopt mini-batch SGD optimizer with momentum 0.9 to train the model. For Office-Home, we use ResNet-50 as the backbone, and the hyperparameters $r_1$, $r_2$, $\lambda_{cst}$ are set to 1/10, 1/15 and 40, respectively. For VisDA-C we adopt ResNet-101 as backbone for fair comparisons. The hyperparameters $r_1$, $r_2$ and $\lambda_{cst}$ are set to 1/4, 1/3 and 40, respectively. The Channel Feature Swapping is performed on the feature after mean pooling over the last residual block of ResNet [12]. The spatial Feature swapping is performed on the activations of the third residual block. We evaluate our approach by applying Adaptive Feature Swapping (AFS) to the source-only model and several mainstream DA methods including DANN [8], CDAN [29], BNM [3], MCC [18], and CST [27] based on their open-source codes. $\lambda_{tran}$ for BNM, CDAN and MCC are kept the same as their original implementations. The reimplemented MCC is marked with † as MCC†. **Codes can be found in the Supplementary.**

**Segmentation.** We evaluate our approach by applying Channel Feature Swapping to the source-only model and the state-of-the-art domain adaptive segmentation method ProDA [57], DAFormer [15], and HRDA [16] based on their open-source codes. Our training is carried out on 2 TITAN RTX GPUs with PyTorch [35]. To train our models using SGD, the initial learning rate is set to 1e-4, and

**Table 1: Accuracy (%) On VisDA-C for UDA using the ResNet-101 backbone. The best accuracy is indicated in bold and the second best one is underlined.**

| Method | plane | bcybl | bus | car | horse | knife | mcyle | persn | plant | sktb | train | trunk | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DAN [28] | 87.1 | 63.0 | 76.5 | 42.0 | 90.3 | 42.9 | 85.9 | 53.1 | 49.7 | 36.3 | 85.8 | 20.7 | 61.1 |
| MCD [40] | 87.0 | 60.9 | 83.7 | 64.0 | 88.9 | 79.6 | 84.7 | 76.9 | 88.6 | 40.3 | 83.0 | 25.8 | 71.9 |
| DRMEA [33] | 92.1 | 75.0 | 78.9 | 75.5 | 91.2 | 81.9 | 89.0 | 77.2 | 93.3 | 77.4 | 84.8 | 35.1 | 79.3 |
| DTA [22] | 93.7 | 82.2 | **85.6** | 83.8 | 93.0 | 81.0 | 90.7 | 82.1 | 95.1 | 78.1 | 86.4 | 32.1 | 81.5 |
| STAR [31] | 95.0 | 84.0 | 84.6 | 73.0 | 91.6 | 91.8 | 85.9 | 78.4 | 94.4 | 84.7 | 87.0 | 42.2 | 82.7 |
| RWOT [54] | 95.1 | 80.3 | 83.7 | 90.0 | 92.4 | 68.0 | **92.5** | 82.2 | 87.9 | 78.4 | **90.4** | **68.2** | 84.0 |
| ResNet [12] | 67.7 | 27.4 | 50.0 | 61.7 | 69.5 | 13.7 | 85.9 | 11.5 | 64.4 | 34.4 | 84.2 | 19.2 | 49.1 |
| +AFS | 95.1 | 59.3 | 83.9 | 72.9 | 95.3 | 66.2 | 91.9 | 63.4 | 93.6 | 62.3 | 84.1 | 17.0 | 73.8 |
| DANN [8] | 87.7 | 34.8 | 84.5 | 44.3 | 66.0 | 62.3 | 73.8 | 48.1 | 59.0 | 34.5 | 67.4 | 18.9 | 56.8 |
| +AFS | 96.7 | 56.1 | 85.2 | 70.0 | 96.0 | 85.7 | 91.8 | 80.0 | 94.5 | 59.0 | 85.0 | 19.0 | 76.6 |
| CDAN [29] | 85.2 | 66.9 | 83.0 | 50.8 | 84.2 | 74.9 | 88.1 | 74.5 | 83.4 | 76.0 | 81.9 | 38.0 | 73.9 |
| +AFS | 93.0 | 73.5 | 80.1 | 69.5 | 91.8 | 97.0 | 82.9 | 79.0 | 83.3 | 72.4 | 84.5 | 39.7 | 78.1 |
| BNM [3] | 95.6 | 86.7 | 77.4 | 48.9 | 92.0 | 83.3 | 76.2 | 76.3 | 88.0 | 64.2 | 84.7 | 46.3 | 76.6 |
| +AFS | 95.9 | 87.0 | 78.1 | 53.6 | 93.7 | 90.5 | 78.8 | 78.8 | 87.4 | 84.9 | 83.4 | 47.0 | 79.9 |
| MCC [18]† | 91.5 | 82.6 | 75.9 | 61.5 | 91.4 | 92.9 | 81.6 | 79.6 | 85.8 | 88.4 | 69.5 | 51.8 | 80.2 |
| +AFS | 93.1 | 78.5 | 81.1 | 79.0 | 93.9 | 94.2 | 88.1 | **85.1** | 91.0 | 85.6 | 85.2 | 41.6 | 83.1 |
| CST+SAM [27]† | 95.2 | **89.0** | 75.8 | **89.6** | **95.7** | **98.5** | 87.8 | 84.1 | 94.4 | 90.9 | 82.4 | 54.5 | 86.5 |
| +AFS | **97.2** | **89.0** | 83.8 | 87.5 | **95.7** | 94.0 | 87.9 | 82.8 | **96.6** | **94.7** | 82.4 | 53.1 | **87.1** |

adjusted according to the 'poly' learning rate scheduler with a power of 0.9. The weight decay is set to 0.0005. These settings are the same as ProDA [57] for fair comparisons. In accordance with [15] and [16], we train our model with AdamW, a base learning rate of 6e-5 for the encoder and 6e-4 for the decoder, a weight decay of 0.01, linear learning rate warmup with 1.5k iteration, and linear decay afterward. The data augmentation and other hyperparameters are kept the same as [15] and [16]. The hyperparameters $r_2$, $\lambda_{cst}$ are set to 1/15 and 15, respectively. The batch size of HRAD is set to 1 rather than 2 as adopted in their open source code due to the limited computation resources (marked with †).

Note that ProDA [57] adopts pseudo label for training target samples, so the consistency loss on target domain is directly implemented as the cross entropy loss of prediction over representations with feature swapping and its corresponding pseudo labels. For DAFormer and HRDA, we swap features from the semantic decoder and obtain consistency loss from both the source domain and target domain. Only Channel Feature Swapping is utilized since swapping spatial features changes their semantic in segmentation task. Besides, due to the limited computation resources, we only implemented ProDA for its first stage with a batch of 2 images (marked with †).

## 4.3 Classification Results

Results on VisDA-C are reported in Tab.1. We can observe that our method consistently improves the generalization ability of all baseline methods. Compared with source only model, integrating our AFS brings 24.7% accuracy improvement on average. The results show that our methods can directly improve the performance without any other tricks. As for CST+SAM [27], our simple model surpasses it by 0.6%. Note that input-consistency via random augmentation is involved in CST+SAM, leaving little room for our method for improvement. However, we can observe that our method could still boost CST+SAM, showing its promising potential to boost existing unsupervised domain adaptation methods. Based on these experiments, we can infer that AFS can stably enhance the transfer ability of classifier.

Results on Office-Home are summarized in Tab. 2. Office-Home is a challenging dataset due to its large domain discrepancy. Utilizing the proposed AFS into source only model brings 19.2% accuracy improvement on average. **Note that Source-only+AFS outperforms SE [6], indicating the potential of AFS to generate more valuable perturbations compared with random augmentation adopted in SE.** Compared with CST [27], our simple model surpasses it by 0.4% on average. Our AFS also boosts the CDAN, BNM and MCC baselines by considerable margins. Based on these promising results, we can infer that AFS can stably explore truly useful semantic information to better adapt the classifier.

**Ablation Study.** In Tab. 3, we validate the influence of the proposed spatial swapping (SS) and channel swapping (CS) in VisDA-C dataset. All the results reported in Tab. 3 show that our methods can stably improve the transferability of the model.

Firstly, we validate the effectiveness of using spatial swapping. Compared with baseline MCC, integrating spatial swapping (MCC+SS) shows a gain of 1.3% accuracy on average. Based on it, we can infer that spatial swapping helps the model to pay more attention to objects instead of the background. Secondly, the large improvement is achieved by adopting channel swapping into baseline MCC (MCC+CS) that encourages the model to focus on semantic information. Finally, we also utilize both two swapping strategies into baseline MCC and it performs better than MCC+SS and MCC+CS. Therefore, the two feature swapping strategies can benefit each other and help the model achieve better performance. **The computation overhead analysis can be referred to the Supplementary.**

**Table 2: Accuracy (%) On Office-Home for UDA using the ResNet-50 backbone. The best accuracy is indicated in bold and the second best one is underlined.**

| Method | Ar→Cl | Ar→Pr | Ar→Rw | Cl→Ar | Cl→Pr | Cl→Rw | Pr→Ar | Pr→Cl | Pr→Rw | Rw→Ar | Rw→Cl | Rw→Pr | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DAN [28] | 43.6 | 57.0 | 67.9 | 45.8 | 56.5 | 60.4 | 44.0 | 43.6 | 67.7 | 63.1 | 51.5 | 74.3 | 56.3 |
| SE [6] | 48.8 | 61.8 | 72.8 | 54.1 | 63.2 | 65.1 | 50.6 | 49.2 | 72.3 | 66.1 | 55.9 | 78.7 | 61.5 |
| SymNet[60] | 47.7 | 72.9 | 78.5 | 64.2 | 71.3 | 74.2 | 64.2 | 48.8 | 79.5 | 74.5 | 52.6 | 82.7 | 67.6 |
| MDD [59] | 54.9 | 73.7 | 77.8 | 60.0 | 71.4 | 71.8 | 61.2 | 53.6 | 78.1 | 72.5 | 60.2 | 82.3 | 68.1 |
| ECT [24] | 51.3 | 71.9 | 85.7 | 57.6 | 69.2 | 73.7 | 57.8 | 51.2 | 79.3 | 70.2 | 57.5 | 82.1 | 67.3 |
| GVB [4] | 57.0 | 74.7 | 79.8 | 64.6 | 74.1 | 74.6 | 65.2 | 55.1 | 81.0 | 74.6 | 59.7 | 84.3 | 70.4 |
| TCM [55] | 58.6 | 74.4 | 79.6 | 64.5 | 74.0 | 75.1 | 64.6 | 56.2 | 80.9 | 74.6 | 60.7 | 84.7 | 70.7 |
| AAA [23] | 56.7 | 78.3 | 82.1 | 66.4 | 78.5 | **79.4** | 67.6 | 53.5 | 81.6 | 74.5 | 58.4 | 84.1 | 71.8 |
| SENTRY [37] | **61.8** | 77.4 | 80.1 | 66.3 | 71.6 | 74.7 | 66.8 | **63.0** | 80.9 | 74.0 | 66.3 | 84.1 | 72.2 |
| ResNet [12] | 34.9 | 50.0 | 58.0 | 37.4 | 41.9 | 46.2 | 38.5 | 31.2 | 60.4 | 53.9 | 41.2 | 59.9 | 46.1 |
| +AFS | 52.5 | 71.4 | 76.7 | 58.0 | 68.6 | 69.6 | 57.5 | 48.1 | 76.1 | 70.2 | 53.9 | 80.7 | 65.3 |
| DANN [8] | 42.5 | 63.5 | 73.5 | 51.4 | 60.7 | 63.5 | 53.6 | 37.2 | 73.0 | 65.2 | 42.6 | 77.1 | 58.7 |
| +AFS | 48.6 | 69.2 | 75.4 | 56.0 | 64.7 | 68.2 | 56.4 | 42.8 | 74.5 | 67.5 | 49.1 | 79.7 | 62.7 |
| CDAN [29] | 50.7 | 70.6 | 76.0 | 57.6 | 70.0 | 70.0 | 57.4 | 50.9 | 77.3 | 70.9 | 56.7 | 81.6 | 65.8 |
| +AFS | 56.3 | 73.8 | 78.4 | 62.7 | 72.4 | 71.4 | 64.3 | 54.4 | 80.0 | 72.9 | 60.2 | 83.0 | 69.2 |
| BNM [3] | 52.3 | 73.9 | 80.0 | 63.3 | 72.9 | 74.9 | 61.7 | 49.5 | 79.7 | 70.5 | 53.6 | 82.2 | 67.9 |
| +AFS | 56.8 | 75.5 | 79.6 | 63.4 | 76.0 | 74.4 | 63.0 | 53.3 | 80.3 | 73.0 | 59.8 | 83.4 | 69.9 |
| MCC [18]† | 55.8 | 77.4 | 80.8 | 66.3 | 75.7 | 75.8 | 65.4 | 53.4 | 80.7 | 74.0 | 59.0 | 84.4 | 70.7 |
| +AFS | 59.0 | 77.5 | 81.4 | 67.2 | 76.8 | 76.0 | 65.8 | 57.3 | 81.9 | 74.0 | 61.8 | 85.0 | 72.0 |
| CST [27]† | 59.0 | 79.6 | **83.4** | **68.4** | 77.1 | 76.7 | **68.9** | 56.4 | **83.0** | **75.3** | 62.2 | 85.1 | 73.0 |
| +AFS | 58.7 | **80.2** | 83.3 | 67.6 | **79.0** | 76.7 | **68.9** | 57.1 | 82.6 | 75.1 | **65.5** | **85.7** | **73.4** |

**Table 3: Ablation Study On VisDA-C for UDA using the ResNet-101 backbone. The best accuracy is indicated in bold and the second best one is underlined.**

| Method | plane | bcybl | bus | car | horse | knife | mcyle | persn | plant | sktb | train | trunk | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MCC [18]† | 91.5 | 82.6 | 75.9 | 61.5 | 91.4 | 92.9 | 81.6 | 79.6 | 85.8 | **88.4** | 69.5 | **51.8** | 80.2 |
| MCC+SS | 92.7 | 83.0 | 78.2 | 69.2 | 91.7 | **94.3** | 85.1 | 80.7 | 88.8 | 86.7 | 81.7 | 46.3 | 81.5 |
| MCC+CS | **93.5** | **85.4** | 78.9 | 70.6 | 93.4 | 93.0 | 86.0 | 81.6 | **92.5** | 85.8 | 80.7 | 46.0 | 82.3 |
| MCC+SS+CS | 93.1 | 78.5 | **81.1** | **79.0** | **93.9** | 94.2 | **88.1** | **85.1** | 91.0 | 85.6 | **85.2** | 41.6 | **83.1** |

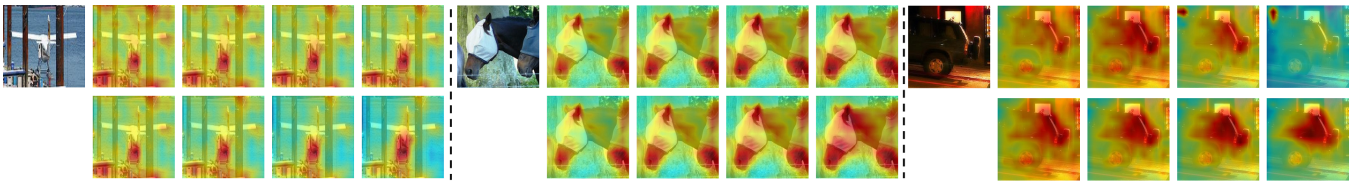

**Figure 2: For each group of images, the 1st column are images from VisDA-C dataset. The 2nd to fifth columns show the attention maps extracted from the activations of third residual block of model trained with 1K, 3K, 5K, and 10K iterations, respectively. The model that produces attention maps of the 1st (2nd), 3rd (4th), and 5th (6th) rows is MCC† baseline (MCC+AFS). The model pays more attention to the regions in red than regions in blue.**

**Attention visualization.** We visualize the attention maps of three images from VisDA-C dataset for baseline MCC and MCC+AFS with Adaptive Feature Swapping in Fig. 2 and Fig. 3. Specifically, the image is input to the trained network, and the activations of a convolutional layer of ResNet-101 is extracted, named A$\in \mathcal{R}^{1024\times W\times H}$. We sum over the first dimension of A and obtain M$\in \mathcal{R}^{W\times H}$. Then we interpolate M into 224×224 to get the final attention map. For Fig. 2, attention map is extracted on the activations of the third residual block, where the spatial feature swapping is performed. As for Fig. 3, attention map is extracted from the activations of the last residual block. These activations are global average pooled and the pooled results are performed with channel feature swapping. Therefore, the attention map of this block could reflect where the model attends when trained with channel feature swapping.

We can observe that, as the model is optimized with increasing iterations, MCC+AFS pays more attention to the semantic region of Plane or Horse. And the background regions attain less attention in MCC+AFS. While in baseline MCC, some background regions

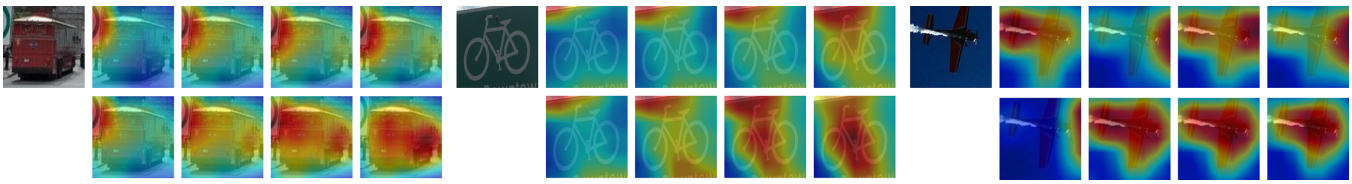

**Figure 3: For each group of images, the 1st column are images from VisDA-C dataset. The 2nd to fifth columns show the attention maps extracted from the activations of the last residual block of model trained with 1K, 3K, 5K, and 10K iterations, respectively. The model that produces attention maps of the 1st (2nd), 3rd (4th), and 5th (6th) rows is MCC† baseline (MCC+AFS). The model pays more attention to the regions in red than regions in blue.**

**Table 4: Comparison results of GTA5→Cityscapes adaptation in terms of mIoU. The best accuracy is indicated in bold and the second best one is underlined.**

| Method | *Road* | *S.walk* | *Build.* | *Wall* | *Fence* | *Pole* | *Tr.light* | *Sign* | *Veget.* | *Terrain* | *Sky* | *Person* | *Rider* | *Car* | *Truck* | *Bus* | *Train* | *M.bike* | *Bike* | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AdaStruct [44] | 86.5 | 25.9 | 79.8 | 22.1 | 20.0 | 23.6 | 33.1 | 21.8 | 81.8 | 25.9 | 75.9 | 57.3 | 26.2 | 76.3 | 29.8 | 32.1 | 7.2 | 29.5 | 32.5 | 41.4 |
| CyCADA [14] | 86.7 | 35.6 | 80.1 | 19.8 | 17.5 | 38.0 | 39.9 | 41.5 | 82.7 | 27.9 | 73.6 | 64.9 | 19.0 | 65.0 | 12.0 | 28.6 | 4.5 | 31.1 | 42.0 | 42.7 |
| CLAN [32] | 87.0 | 27.1 | 79.6 | 27.3 | 23.3 | 28.3 | 35.5 | 24.2 | 83.6 | 27.4 | 74.2 | 58.6 | 28.0 | 76.2 | 33.1 | 36.7 | 6.7 | 31.9 | 31.4 | 43.2 |
| ADVENT [47] | 89.4 | 33.1 | 81.0 | 26.6 | 26.8 | 27.2 | 33.5 | 24.7 | 83.9 | 36.7 | 78.8 | 58.7 | 30.5 | 84.8 | 38.5 | 44.5 | 1.7 | 31.6 | 32.4 | 45.5 |
| BDL [25] | 91.0 | 44.7 | 84.2 | 34.6 | 27.6 | 30.2 | 36.0 | 36.0 | 85.0 | 43.6 | 83.0 | 58.6 | 31.6 | 83.3 | 35.3 | 49.7 | 3.3 | 28.8 | 35.6 | 48.5 |
| FADA [48] | 91.0 | 50.6 | 86.0 | 43.4 | 29.8 | 36.8 | 43.4 | 25.0 | 86.8 | 38.3 | 87.4 | 64.0 | 38.0 | 85.2 | 31.6 | 46.1 | 6.5 | 25.4 | 37.1 | 50.1 |
| MRKLD [62] | 91.0 | 55.4 | 80.0 | 33.7 | 21.4 | 37.3 | 32.9 | 24.5 | 85.0 | 34.1 | 80.8 | 57.7 | 24.6 | 84.1 | 27.8 | 30.1 | 26.9 | 26.0 | 42.3 | 47.1 |
| CAG_UDA [58] | 90.4 | 51.6 | 83.8 | 34.2 | 27.8 | 38.4 | 25.3 | 48.4 | 85.4 | 38.2 | 78.1 | 58.6 | 34.6 | 84.7 | 21.9 | 42.7 | 41.1 | 29.3 | 37.2 | 50.2 |
| Seg-U [61] | 90.4 | 31.2 | 85.1 | 36.9 | 25.6 | 37.5 | 48.8 | 48.5 | 85.3 | 34.8 | 81.1 | 64.4 | 36.8 | 86.3 | 34.9 | 52.2 | 1.7 | 29.0 | 44.6 | 50.3 |
| Source-only | 75.8 | 16.8 | 77.2 | 12.5 | 21.0 | 25.5 | 30.1 | 20.1 | 81.3 | 24.6 | 70.3 | 53.8 | 26.4 | 49.9 | 17.2 | 25.9 | 6.5 | 25.3 | 36.0 | 36.6 |
| +AFS | 76.4 | 21.1 | 74.3 | 19.6 | 23.6 | 33.3 | 36.2 | 26.3 | 82.5 | 22.9 | 72.7 | 61.2 | 32.2 | 73.5 | 32.9 | 28.0 | 0.8 | 28.8 | 27.2 | 40.7 |
| ProDA [57]† | 87.6 | 55.3 | 77.7 | 41.9 | 35.3 | 41.6 | 43.9 | 48.2 | 86.6 | 40.7 | 81.0 | 65.6 | 22.9 | 86.8 | 38.0 | 49.8 | 0.0 | 41.9 | 50.7 | 52.4 |
| +AFS | 90.5 | 55.8 | 80.8 | 41.8 | 36.0 | 41.9 | 45.2 | 48.6 | 86.7 | 41.1 | 81.0 | 66.2 | 22.8 | 87.5 | 39.9 | 49.6 | 0.0 | 42.6 | 51.6 | 53.1 |
| DAFormer [15] | 95.7 | 70.2 | 89.4 | 53.5 | **48.1** | 49.6 | 55.8 | 59.4 | 89.9 | 47.9 | 92.5 | 72.2 | 44.7 | 92.3 | 74.5 | 78.2 | 65.1 | 55.9 | 61.8 | 68.3 |
| +AFS | 96.1 | 71.8 | 89.4 | 53.8 | 44.5 | 49.9 | 56.6 | **62.5** | 90.0 | **51.0** | 90.9 | 71.7 | **45.0** | 92.3 | 72.3 | 81.9 | 72.0 | 56.6 | 63.6 | 69.0 |
| HRDA [16]† | 95.5 | 69.1 | 89.9 | 56.2 | 46.7 | 52.6 | 58.4 | 60.8 | 89.6 | 45.4 | 93.6 | **74.2** | 25.9 | 93.2 | **82.6** | 80.3 | 66.6 | 60.5 | 66.8 | 68.8 |
| +AFS | **96.4** | **74.5** | **90.0** | **57.2** | 45.2 | **54.5** | **60.3** | 60.6 | **90.6** | 49.3 | **94.0** | 73.8 | 22.8 | **93.5** | 81.0 | **84.3** | **73.4** | **62.6** | **67.3** | **70.1** |

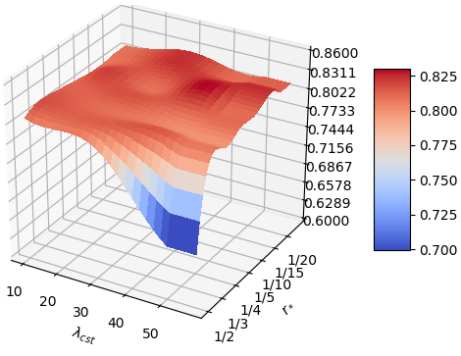


**Figure 4: Hyper-parameter sensitivity analysis of AFS on VisDA-C based on MCC [18].**

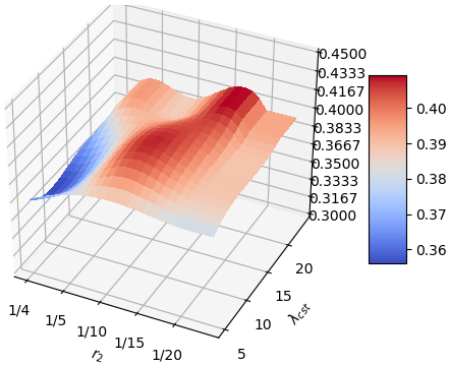


**Figure 5: mIoUs of various Source only+AFS on GTA2Cityscape.**

are still highlighted with considerable attention. The superiority of MCC+AFS over MCC verifies that learning with Adaptive Feature Swapping encourages the model to capture semantic information. **More results including learned representation visualization can be referred to the Supplementary.**

**Hyper-parameter Sensitivity For Classification Task.** Hyper-parameter $r_1$ and $r_2$ control the perturbation strength of the representations with feature swapping and $\lambda_{cst}$ balances the consistency loss with other losses. To evaluate the parameter sensitivity of AFS, we conduct experiments with MCC+AFS on VisDA-C. To reduce the complexity of evaluation, we set $r = r_1 = r_2$ and vary $r \in \{1/2, 1/3, 1/4, 1/5, 1/10, 1/15, 1/20\}$ and $\lambda_{cst} \in \{10, 20, 30, 40, 50\}$. Fig. 4 shows that, except for combinations of large $r = \{1/2, 1/3\}$ and large $\lambda_{cst} = \{40, 50\}$, MCC+AFS is not that sensitive to $r_1$, $r_2$ and $\lambda_{cst}$, and can achieve competitive results under a wide range of hyper-parameter values.

## 4.4 Segmentation Results

We report the quantitative evaluation of our method in Tab. 4. From the results, we can see that with Channel Feature Swapping,

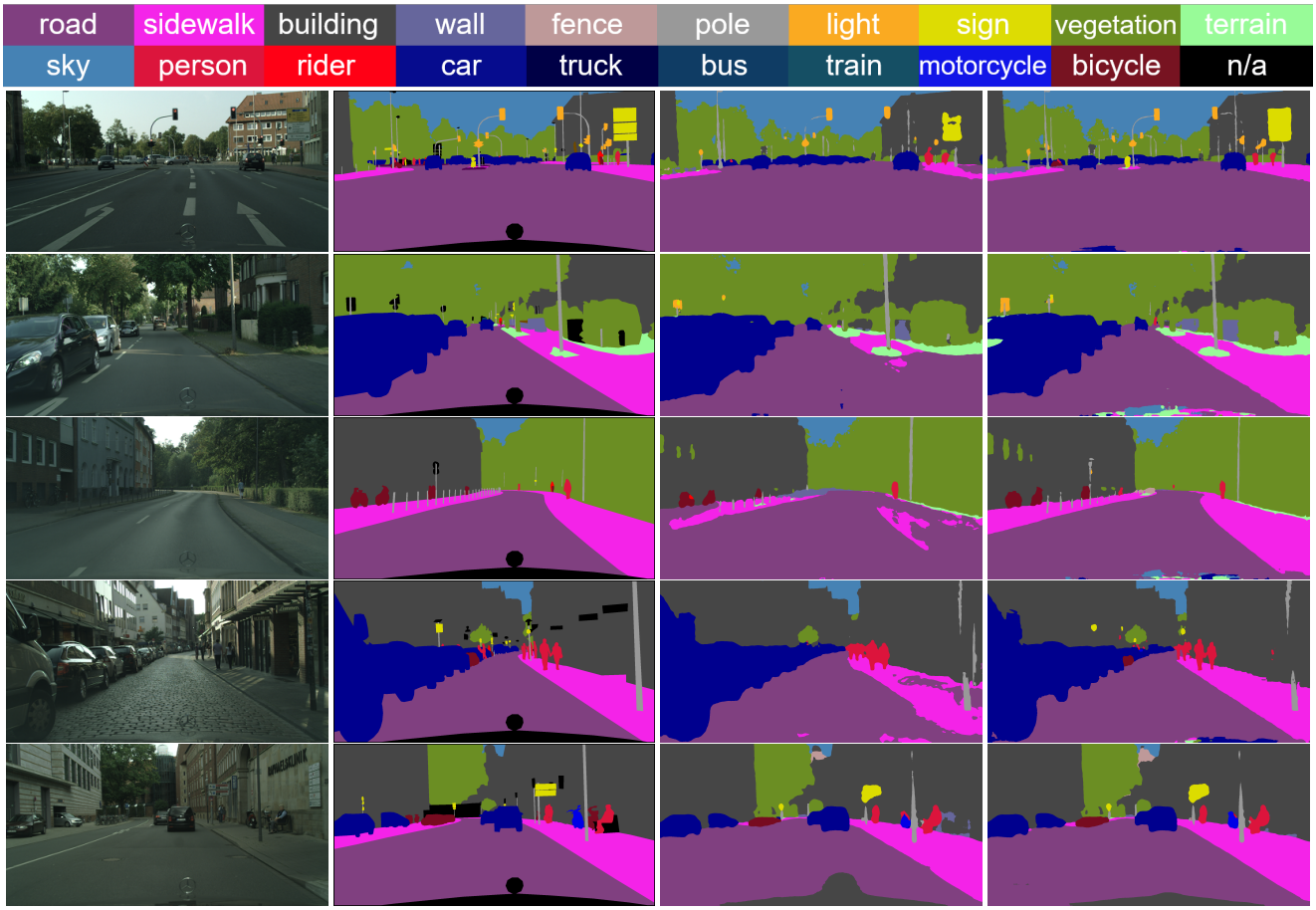


**Figure 6: Qualitative results of validation images from Cityscapes where the source domain is GTA5 dataset. The 1st and 2nd columns are images and ground truth labels from validation set of Cityscape. The 3rd and 4th columns are segmentation results produced by DAFormer [15]† and our method.**

the source-only model is improved by 4.1% in mIoU. Besides, we can observe that applying our Channel Feature Swapping into the ProDA [57]† with ResNet-101 improves the baseline ProDA† by 0.7% in mIoU. Channel Feature Swapping also boosts DAFormer [15] and HRDA [16] by 0.7% and 1.3% in mIoU, respectively. For the challenging domain adaptive segmentation task, the 0.7% improvement in mIoU is not marginal, which validates the effectiveness of Channel Feature Swapping in enhancing the transfer ability of the model.

We provide visual comparison bewteen DAFormer [15] and DAFormer+AFS in Fig. 6. We also visually compare the baseline HRDA [16]† and HRDA†+AFS in Fig. 6. As we can see from these figures, the predictions from our model appear less noisy. We accredit this to the enhanced generalization ability of utilizing Channel Feature Swapping. **More qualitative comparisons can be found in the Supplementary.**

We conduct experiments on semantic segmentation task w.r.t to $r_2$ and $\lambda_{cst}$, since Spatial Feature Swapping is not adopted for it destroys the semantic for segmentation task. We vary $r_2 \in \{1/4, 1/5, 1/10, 1/15, 1/20\}$ and $\lambda_{cst} \in \{5, 10, 15, 20\}$, and the results are shown in Fig. 5. We can observe that Source only+AFS is not that sensitive to $r_2$ and $\lambda_{cst}$, and can achieve competitive results under a wide range of hyper-parameter values. Intuitively, the redundancy of features for segmentation tasks is less since it requires much more information for dense prediction. The swapping ratio should be smaller compared with classification.

## 5 CONCLUSION

Aiming to model domain-specific features and achieve prediction robustness toward sample perturbation in an efficient manner, we propose Adaptive Feature Swapping for learning domain invariant representations in Unsupervised Domain Adaptation (UDA). Specifically, we utilize attention mechanisms to select semantically irrelevant features from labeled source data and unlabeled target data, and swap these features from each other. Then consistency between predictions of original representations and the representations after swapping is enforced such that the model is insensitive to the irrelevant features. We develop two swapping strategies including channel swapping and spatial swapping and extensive experiments on object recognition and semantic segmentation in UDA setting validate the effectiveness of feature swapping. It is promising to investiage Adaptive Feature Swapping in UDA under conditional distribution shift and domain generalization task and we leave these tasks for our future work.

# REFERENCES


[1] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems*. 343–351.

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.

[3] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. 2020. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*. 3941–3950.

[4] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. 2020. Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12455–12464.

[5] Jiahua Dong, Yang Cong, Gan Sun, Zhen Fang, and Zhengming Ding. 2021. Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[6] Geoff French, Michal Mackiewicz, and Mark Fisher. 2018. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*.

[7] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3146–3154.

[8] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. 1180–1189.

[9] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. 2019. Dlow: Domain flow for adaptation and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2477–2486.

[10] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. 2006. A kernel method for the two-sample-problem. *Advances in neural information processing systems* 19 (2006), 513–520.

[11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, Mar (2012), 723–773.

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

[13] Qi He, Zhaoquan Yuan, Xiao Wu, and Jun-Yan He. 2022. Domain-Specific Conditional Jigsaw Adaptation for Enhancing transferability and Discriminability. In *Proceedings of the 30th ACM International Conference on Multimedia*. 6327–6336.

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*. PMLR, 1989–1998.

[15] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. 2021. DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation. *arXiv preprint arXiv:2111.14887* (2021).

[16] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. 2022. HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation. *arXiv preprint arXiv:2204.13132* (2022).

[17] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[18] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. 2020. Minimum class confusion for versatile domain adaptation. In *ECCV*. 464–480.

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

[20] Nikos Komodakis and Sergey Zagoruyko. 2017. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *ICLR*.

[21] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. 2019. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10285–10295.

[22] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. 2019. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 91–100.

[23] Jingjing Li, Zhekai Du, Lei Zhu, Zhengming Ding, Ke Lu, and Heng Tao Shen. 2021. Divergence-agnostic Unsupervised Domain Adaptation by Adversarial Attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[24] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. 2020. Enhanced transport distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13936–13944.

[25] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. 2019. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6936–6945.

[26] Min Lin, Qiang Chen, and Shuicheng Yan. 2014. Network In Network. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[27] Hong Liu, Jianmin Wang, and Mingsheng Long. 2021. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems* 34 (2021).

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

[29] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. 2018. Conditional Adversarial Domain Adaptation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 1647–1657.

[30] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*. 2208–2217.

[31] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. 2020. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9111–9120.

[32] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. 2019. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2507–2516.

[33] You-Wei Luo, Chuan-Xian Ren, Pengfei Ge, Ke-Kun Huang, and Yu-Feng Yu. 2020. Unsupervised domain adaptation via discriminative manifold embedding and alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5029–5036.

[34] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2018. Sensitivity and Generalization in Neural Networks: an Empirical Study. In *International Conference on Learning Representations*.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

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

[37] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. 2021. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8558–8567.

[38] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games. In *European conference on computer vision*. Springer, 102–118.

[39] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. 2019. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9471–9480.

[40] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3723–3732.

[41] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference* 90, 2 (2000), 227–244.

[42] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*. Springer, 443–450.

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

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

[45] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[46] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5018–5027.

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

[48] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. 2020. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *European Conference on Computer Vision*. Springer, 642–659.

[49] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. 2020. Deep multimodal fusion by channel exchanging. *Advances in Neural Information Processing Systems* 33 (2020).

[50] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. 2020. Theoretical Analysis of Self-Training with Deep Networks on Unlabeled Data. In *International Conference on Learning Representations*.

[51] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. 2021. ToAlign: Task-oriented Alignment for Unsupervised Domain Adaptation. *Advances in Neural Information Processing Systems* 34 (2021).

[52] Liang Xiao, Jiaolong Xu, Dawei Zhao, Zhiyu Wang, Li Wang, Yiming Nie, and Bin Dai. 2021. Self-Supervised Domain Adaptation with Consistency Training. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 6874–6880.

[53] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. 2019. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*. 1426–1435.

[54] Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. 2020. Reliable weighted optimal transport for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4394–4403.

[55] Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. 2021. Transporting causal mechanisms for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8599–8608.

[56] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. 2017. Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[57] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. 2021. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12414–12424.

[58] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. 2019. Category Anchor-Guided Unsupervised Domain Adaptation for Semantic Segmentation. *Advances in Neural Information Processing Systems* 32 (2019), 435–445.

[59] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*. 7404–7413.

[60] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. 2019. Domain-Symmetric Networks for Adversarial Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5031–5040.

[61] Zhedong Zheng and Yi Yang. 2021. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision* 129, 4 (2021), 1106–1120.

[62] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. 2019. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5982–5991.