RDA: Robust Domain Adaptation via Fourier Adversarial Attacking

Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu*
Singtel Cognitive and Artificial Intelligence Lab for Enterprises,
School of Computer Science and Engineering, Nanyang Technological University
{Jiaxing.Huang, Dayan.Guan, Aoran.Xiao, Shijian.Lu}@ntu.edu.sg

Abstract

Unsupervised domain adaptation (UDA) involves a supervised loss in a labeled source domain and an unsupervised loss in an unlabeled target domain, which often faces more severe overfitting (than classical supervised learning) as the supervised source loss has clear domain gap and the unsupervised target loss is often noisy due to the lack of annotations. This paper presents RDA, a robust domain adaptation technique that introduces adversarial attacking to mitigate overfitting in UDA. We achieve robust domain adaptation by a novel Fourier adversarial attacking (FAA) method that allows large magnitude of perturbation noises but has minimal modification of image semantics, the former is critical to the effectiveness of its generated adversarial samples due to the existence of ‘domain gaps’. Specifically, FAA decomposes images into multiple frequency components (FCs) and generates adversarial samples by just perturbing certain FCs that capture little semantic information. With FAA-generated samples, the training can continue the ‘random walk’ and drift into an area with a flat loss landscape, leading to more robust domain adaptation. Extensive experiments over multiple domain adaptation tasks show that RDA can work with different computer vision tasks with superior performance.

1. Introduction

Deep convolutional neural networks (CNNs) [39, 70, 21] have defined new state of the arts in various computer vision tasks [7, 44, 60, 20, 39, 70, 21], but their trained models often over-fit to the training data and experience clear performance drops for data from different sources due to the existence of domain gaps. Unsupervised domain adaptation (UDA) has been investigated to address the domain gaps by leveraging unlabeled target data. To this end, most existing UDA works [34, 35, 75, 78, 48, 80, 8, 9] involve supervised losses on source data and unsupervised losses on target data for learning a model that performs well in target domains. However, as illustrated in Fig. 1, these methods often face more severe overfitting (as compared with the classical supervised learning) as supervised source losses in UDA has an extra domain gap (for test data in target domains) and unsupervised target losses are often noisy due to the lack of annotations.

Overfitting exists in almost all deep network training, which is undesired and often degrades the generalization of
the trained deep network models while applied to new data. One way of identifying whether overfitting is happening is to check whether the generalization gap, i.e., the difference between the test loss and the training gap, is increasing or not [13] as shown in Fig. 1. Various strategies have been investigated to alleviate overfitting through weight regularization [19], dropout [71], mixup [93], label smoothing [72], batch normalization [30], etc. However, all these strategies were designed for supervised and semi-supervised learning where training data and test data usually have very similar distributions. For domain adaptive learning, they do not fit in well due to the negligence of domain gaps that widely exist between data of different domains.

We design a robust domain adaptation technique that introduces a novel Fourier adversarial attacking (FAA) technique to mitigate the overfitting in unsupervised domain adaptation. FAA mitigates overfitting by generating adversarial samples that prevent over-minimization of supervised and unsupervisedUDA losses as illustrated in Fig. 1. Specifically, FAA decomposes training images into multiple frequency components (FCs) and only perturbs FCs that capture little semantic information. Unlike traditional attacking that restricts the magnitude of perturbation noises to keep image semantics intact, FAA allows large magnitude of perturbation in its generated adversarial samples but has minimal modification of image semantics. This feature is critical to unsupervised domain adaptation which usually involves clear domain gaps and so requires adversarial sample with large perturbations. By introducing the FAA-generated adversarial samples in training, networks can continue the “random walk” and avoid over-fitting and drift into an area with a flat loss landscape [6, 36, 40], leading to more robust domain adaptation.

The contributions of this work can be summarized in three aspects. First, we identify the overfitting issue in unsupervised domain adaptation and introduce adversarial attacking to mitigate overfitting by preventing training objectives from over-minimization. Second, we design an innovative Fourier Adversarial Attacking (FAA) technique to generate novel adversarial samples to mitigate overfitting in domain adaptation. FAA is generic which can work for both supervised source loss and unsupervised target losses effectively. Third, we conducted extensive experiments over multiple computer vision tasks in semantic segmentation, object detection and image classification. All experiments show that our method mitigates overfitting and improve domain adaption consistently.

2. Related Works

Domain Adaptation: Domain adaptation has been studied extensively for mitigating data annotation constraints. Most existing works can be broadly classified into three categories. The first category is adversarial training based which employs a discriminator to align source and target domains in the feature, output or latent space [46, 78, 75, 67, 24, 80, 76, 25, 88, 95, 18, 28]. The second category is image translation based which adapt image appearance to mitigate domain gaps [41, 92, 90, 94]. The third category is self-training based which predict pseudo labels or minimize entropy to guide iterative learning from target samples [99, 98, 49, 16, 26, 17].

Domain adaptation involves two typical training losses, namely, supervised loss over labeled source data and unsupervised loss over unlabeled target data. State-of-the-art methods tend to over-minimize the two types of losses which directly leads to deviated models with suboptimal adaptation as illustrated in Fig. 1. We design a robust domain adaptation technique that addresses this issue by preventing loss over-minimization.

Overfitting in Network Training: Overfitting is a common phenomenon in deep network training which has been widely studied in the deep learning and computer vision community [2, 5, 54, 62, 84, 82]. Most existing works address overfitting by certain regularization strategies such as weight decay [19], dropout [71], \(l_1\) regularization [74], mixup [93], label smoothing [72], batch normalization [30], virtual adversarial training [51], flooding [31], etc. However, these strategies were mostly designed for supervised or semi-supervised learning which do not fit in well for domain adaptive learning that usually involves domain gaps and unsupervised losses. We design a adversarial attacking technique that mitigates the overfitting in domain adaptive learning effectively.

Adversarial Attacking: Adversarial attacking has been studied in various security problems. For example, [73] shows that adversarial samples can easily confuse CNN models. The following works improved adversarial attacking from different aspects via fast gradient signs [14], minimal adversarial perturbation [53], universal adversarial perturbations [52], gradient-free attacking [1], transferable adversarial sample generation[43], etc. Adversarial attacking has also been applied to other tasks, e.g., [96, 51] employed adversarial samples to mitigate over-fitting in supervised and semi-supervised learning, [79] generated adversarial samples for data augmentation, and [42, 89] augments transferable features for domain divergence minimization.

Most existing adversarial attacking methods commonly constrain the magnitude of perturbation noises for minimal modification of image semantics. However, such generated adversarial samples cannot tackle overfitting in domain adaptive learning well which usually involves a domain gap of fair magnitude. We design an innovative Fourier adversarial attacking technique that allows to generate adversarial samples without magnitude constraint yet with minimal modification of image semantics, more details to be described in the ensuing subsections.
3. Method

We achieve robust domain adaptation via Fourier adversarial attacking as illustrated in Fig. 2. The training consists of two phases, namely, an Attacking Phase and a Defending Phase. Given a training image, the attacking phase learns to identify the right FCs with limited semantic information that allow perturbation noises of large magnitude. It also learns to generate adversarial samples (with perturbable FCs) with minimal modification of image semantics. During the defending phase, the generated adversarial sample is applied to mitigate overfitting by preventing over-minimization of training losses, more details to be described in the ensuing subsections.

3.1. Task Definition

We focus on the problem of unsupervised domain adaptation (UDA). Given labeled source data \( \{X_s, Y_s\} \) and unlabeled target data \( X_t \), our goal is to learn a task model \( F \) that performs well on \( X_t \). The baseline model is trained with the labeled source data only:

\[
\mathcal{L}_{sup}(X_s, Y_s; F) = l(F(X_s), Y_s),
\]

where \( l(\cdot) \) denotes an accuracy-related loss, e.g., the standard cross-entropy loss.

3.2. Fourier Adversarial Attacking

Our proposed Fourier adversarial attacking (FAA) generates adversarial samples to attack the training loss to mitigate overfitting in domain adaptation, as shown in Fig. 2. In adversarial sample generation, it first employs Fourier transformation to decompose input images into multiple frequency components (FCs) and then inject perturbation to non-semantic FCs which allows perturbation of large magnitude but with minimal modification of image semantics.

**Fourier Decomposition:** Inspired by JPEG [81, 58] and frequency-domain learning [27, 33], we transform an image \( x \) into frequency space and decompose it into multiple FCs which allows explicit manipulation of each FC and more controllable perturbations. We employ Fourier transformation to convert \( x \) into frequency space and further decompose it into multiple FCs of equivalent bandwidth:

\[
z = \mathcal{F}(x),
\]

\[
z^{fc} = \mathcal{D}(z; N),
\]

where \( \mathcal{F}(\cdot) \) stands for Fourier transformation [4]; \( z \) denotes the frequency-space representation of \( x \); \( \mathcal{D}(z; N) \) denotes a function that decomposes \( z \) into \( N \) FCs \( z^{fc} = \{z^1, z^2, ..., z^{N-1}, z^N\} \) of equivalent bandwidth.

We consider a gray-scale image \( x \in \mathbb{R}^{H \times H} \) for defining \( \mathcal{D}(z; N) \) formally. We thus have \( z \in \mathbb{C}^{H \times H} \), where \( \mathbb{C} \) denotes complex numbers. We use \( z(i, j) \) to index \( z \) at \( (i, j) \), and \( c_i \) and \( c_j \) to denote the image centroid (i.e., image center at \((H/2, H/2)\)). The equation \( \{z^1, z^2, ..., z^{N-1}, z^N\} = \mathcal{D}(z; N) \) can be formally defined by:

\[
z^{n}(i, j) = \begin{cases} 
    z(i, j), & \text{if } \frac{n-1}{N} < d(l(i, j), (c_i, c_j)) < \frac{n}{N}, \\
    0, & \text{otherwise,}
\end{cases}
\]

where \( d(\cdot, \cdot) \) denotes Euclidean distance, \( N \) denotes how many FCs the input is supposed to be decomposed into and \( n \) denotes the FC index.

To get \( z^{fc} = \{z^1, z^2, ..., z^{N-1}, z^N\} = \mathcal{D}(z; N) \), we perform the equation \( N \) times by changing \( n \) from 1 to \( N \). If \( x \) does not have a square size, we first up-sample the short side to be same as the long side before this process and down-sample it back to the original size after processing. If \( x \) has more than one channel (e.g., RGB image \( x \in \mathbb{R}^{n \times n \times 3} \)), we process each channel independently.

**Adversarial Attacking:** With the decomposed FCs \( z^{fc} = \{z^1, z^2, ..., z^{N-1}, z^N\} \), we attack domain adaption losses by perturbing partial FCs without magnitude constraint. Specifically, we employ a learnable gate module [32, 77, 87, 50] to select certain FCs for perturbation. This gate assigns a binary score to each FC, where ‘1’ indicates this FC is selected for perturbation (multiplied by ‘1’) while ‘0’ shows this FC is discarded (reset to all zero values). \( G \) works with Gumbel-Softmax, a differentiable sampling mechanism for categorical variables that can be trained via standard back-propagation. Please refer to [32, 77, 87, 50] for details.

Given a reference image (an randomly selected target-domain image) in FC representation \( z^{fc}_{ref} = \{z^1_{ref}, z^2_{ref}, ..., z^{N-1}_{ref}, z^N_{ref}\} = \mathcal{D}(F(x_{ref}); N) \), we employ this gate module to block some FCs (all reset to zero), and apply the selected FCs to perturb the corresponding FCs of the input image \( x \):

\[
z^{fc} = (1 - G(z^{fc}))z^{fc} + G(z^{fc})z^{fc}_{ref},
\]

where the learnable gate module \( G \) enables binary channel attention that selects which FCs to perturb.

Specifically, for the identified FCs of the input image \( x \), we extract the corresponding FCs of a reference image \( x_{ref} \) and use them as perturbation noises. We generate adversarial samples in this manner because the perturbation noises (i.e., non-semantic FCs of \( z^{fc}_{ref} \)) from target natural images are more reasonable and meaningful as compared to a random noises/signals as in many existing adversarial sample generation methods. In addition, the use of non-semantic FCs of target samples mitigates inter-domain gaps which helps to improve target-domain performance in domain adaptation.

With the perturbed FCs \( z^{fc} \), we convert them all back (via inverse Fourier transformation) to get the adversarial sample \( x^{FAA} \):

\[
x^{FAA} = \mathcal{F}^{-1}(G(z^{fc})),
\]
3.3. FAA Training

The proposed FAA involves three types of losses including task loss (i.e., UDA training loss in this work) that is to be attacked, the gate related loss that constrains the proportion of perturbable FCs, and the reconstruction loss that aims to minimize the attacking effects over image semantics.

Given input images \( X \subset \mathbb{R}^{H \times W \times 3} \), a task model \( F \) and the Fourier attacker \( A \), we denote the task loss by \( \mathcal{L}(X; A, F) \). Note the task loss refers to UDA training loss in this work but it can be other types of losses such as supervised or unsupervised training losses. The gate related loss and the reconstruction loss are defined by:

\[
\mathcal{L}_{\text{gat}}(X; A) = \left\{ \begin{array}{cl} 
\sum (G(z^{fc}), & \text{if } \sum (G(z^{fc})) > N \times p,
0, & \text{otherwise},
\end{array} \right.
\]
\[
\mathcal{L}_{\text{rec}}(X; A) = |R(X) - R(A(X))|,
\]

where \( G(z^{fc}) \) is a gating process in \( \mathcal{A} \) as described in Eq. 2. \( R \) is a band-pass filter to get the mid-frequency content [27] that captures semantic information (e.g., structures and shapes) and thus the consistency loss can ensure that the selected FCs contains minimal semantic information. \( p \) is a hyper-parameter that constrains the maximum number of perturbable FCs by \( N \times p \).

The overall training objective of FAA is formulated by:

\[
\max_{\mathcal{A}} \mathcal{L}(X; A, F) = \mathcal{L}_{\text{gat}}(X; A) + \mathcal{L}_{\text{rec}}(X; A)
\]

3.4. Robust Domain Adaptation

Sections 3.2 and 3.3 describe the “Attacking Phase” that generates adversarial samples via FAA. This section presents the proposed robust domain adaptation technique where FAA is used to mitigate the overfitting in UDA in a “Defending Phase”. Specifically, we apply the
drift into an area with a flat loss landscape \([6, 36, 40]\), leading over-minimization of UDA losses by forcing them (adversarial training). Please refer to Algorithm 1 for details way and form a bidirectional training framework.

Defending Phase

Attacking Phase

Given the task model \(F\) and the Fourier attacker \(A\), the training objective of the task model \(F\) is formulated by:

\[
\min_F L(X; A, F)
\]  

The optimization functions of both attacking (i.e., Eq. 5) and defending (i.e., Eq. 6) are generic and applicable to various tasks and data. Specifically, the model \(F\) could be for semantic segmentation, object detection or image classification task. The data could be labeled source data with supervised source losses or unlabeled target data with unsupervised target losses such as adversarial loss \([75, 48, 80, 89, 55, 42, 29, 66]\), self-training loss \([99, 98, 41, 83, 37, 90]\) or entropy loss \([80, 15]\), etc. Below are a supervised loss and a self-training-based unsupervised loss here for reference:

\[
\begin{align*}
L_{sup}^s(X_s; A, F) &= l(F(A(X_s))), Y_s) , \\
L_{un}^s(X_t; A, F) &= l(F(A(X_t))), \hat{Y}_t) ,
\end{align*}
\]

where \(l(\cdot)\) denotes an accuracy-related loss, e.g., cross entropy loss; \(A(X_s)\) and \(A(X_t)\) are inputs perturbed by FAA, and \(Y_t\) is the pseudo label of unlabeled target data.

In summary, our robust domain adaptation has a bidirectional training framework including an Attacking Phase and a Defending Phase as shown in Fig. 2. In the Attacking Phase, the task model \(F\) is fixed and the attacker \(A\) generates adversarial samples to increase the training loss. In the Defending Phase, \(A\) is fixed and \(F\) is updated to reduce the training loss. These two phases are conducted in an alternative way way and form a bidirectional training framework (adversarial training). Please refer to Algorithm 1 for details.

This bidirectional training framework uses FAA to prevent over-minimization of UDA losses by forcing them oscillating around a small value. In another word, it ensures that the task model \(F\) can continue the “random walk” and drift into an area with a flat loss landscape [6, 36, 40], leading to robust domain adaptation.

4. Experiments

This section presents experiments including datasets and implementation details, domain adaptation for semantic segmentation (with ablation studies), object detection, and image classification tasks, and discussion, more details to be described in the ensuing subsections.

4.1. Datasets

**Adaptation for semantic segmentation:** We consider two synthesized-to-real segmentation tasks: 1) GTA5 [61] → Cityscapes [10] and 2) SYNTHIA [63] → Cityscapes. GTA5 contains 24,966 synthetic images and shares 19 categories with Cityscapes. For SYNTHIA, we use ‘SYNTHIA-RAND-CITYSCAPES’ which contains 9,400 synthetic images and shares 16 categories with Cityscapes. Cityscapes contains 2975/500 training/validation images. Following [75, 99], we adapt towards the Cityscapes training set and evaluate on the Cityscapes validation set.

**Adaptation for object detection:** We consider two adaptation tasks: 1) Cityscapes → Foggy Cityscapes [68] and 2) Cityscapes → BDD100k [91]. For Cityscapes, we convert instance segmentation annotations to bounding boxes in experiments. Foggy Cityscapes was created by applying synthetic fog on Cityscapes images. BDD100k contains 100k images including 70k for training and 10k for validation. Following [86, 66, 9], we use a BDD100k subset daytime in experiments, which includes 36, 728 training images and 5, 258 validation images.

**Adaptation for image classification:** We adopt two adaptation benchmarks VisDA17 [57] and Office-31 [64]. VisDA17 includes 152, 409 synthetic images of 12 categories as source and 55, 400 real images as target. Office-31 contains images of 31 classes from Amazon (A), Webcam (W) and DSLR (D) that have 2817, 795 and 498 images, respectively. We evaluate on six tasks \(A \rightarrow W, D \rightarrow W, W \rightarrow D, A \rightarrow D, D \rightarrow A,\) and \(W \rightarrow A\) as in [98, 64, 69].

4.2. Implementation Details

**Semantic segmentation:** We use DeepLab-V2 [7] with ResNet101 [21] as the segmentation network as in [75, 99]. We use SGD optimizer [3] with a momentum 0.9 and a weight decay \(1e^{-4}\). The initial learning rate is \(2.5e^{-4}\) and decayed by a polynomial policy of power 0.9 [7].

**Object detection:** As in [86, 66, 9], we use Faster R-CNN [7] with VGG16 [70] as the detection network. We use SGD optimizer [3] with a momentum 0.9 and a weight decay \(5e^{-4}\). The initial learning rate is \(1e^{-3}\) for 50k iterations and reduced to \(1e^{-4}\) for another 20k iterations [86, 66, 9]. In all experiments, we set image shorter side to 600 and employ RoIAlign [20] for feature extraction.

**Image classification:** For fair comparisons, we follow [98, 64, 69] and use ResNet-101/ResNet-50 [21] (pre-
trained with ImageNet [11] as backbones. We use SGD optimizer [3] with a momentum 0.9 and a weight decay 5e−4. The learning rate is $1 \times 10^{-3}$ and the batch size is 32 [98].

We set the parameter $p$ and the number of FCs $N$ at 0.1 and 96. The band-pass filter $R$ follows [27] with mid-pass and low-/high-rejected designs to get the mid-frequency content that captures semantic information (e.g., structures and shapes).

4.3. Domain Adaptive Semantic Segmentation

Table 1 shows experimental results over semantic segmentation task GTA5 → Cityscapes. It can be seen that FAA is generic and can be applied to attack both BaseLine (for preventing overfitting in supervised source loss) and state-of-the-art UDA methods (for preventing overfitting in both supervised source loss and unsupervised target loss). In addition, incorporating FAA improves both BaseLine and UDA methods clearly and consistently.

We perform ablation studies over two representative UDA methods using adversarial alignment [75] and self-training [99], where +FAA-S, +FAA-T and FAA address the overfitting of supervised source losses, unsupervised target losses and both losses, respectively. It can be seen that +FAA-S and +FAA-T both improve domain adaptation by large margins. This shows that both supervised source objective and unsupervised target objective intro-
duce clear overfitting and FAA mitigates the overfitting effectively. In addition, +FAA performs clearly the best. This shows that preventing the two learning objectives from over-minimization is complementary as overfitting in the two learning objectives affects generalization in different manners. Specifically, supervised source loss has domain gap and over-minimizing it guides the model to over-memorize source data whereas unsupervised target loss is noisy and over-minimizing it leads to deviated solutions with accumulated errors.

Table 2 shows experimental results over semantic segmentation task SYNTHIA → Cityscapes. We can observe that FAA improves state-of-the-art UDA methods in the similar manners as in Table 1. Note we applied FAA to a few representative UDA methods only due to space limit.

### 4.4. Domain Adaptive Object Detection

Table 3 shows domain adaptive object detection over the task Cityscapes → Foggy Cityscapes. It can be seen that FAA boosts mAP by over +2.5% for both SWDA [66] and CRDA [86]. Note that we did not apply FAA to other listed UDA methods due to space limit.

Table 4 shows domain adaptive object detection over the task Cityscapes → BDD100k. It can be seen that including FAA outperforms state-of-the-art UDA methods consistently as in Table 3. Note that we did not apply FAA to other listed UDA methods due to space limit.

### 4.5. Domain Adaptive Image Classification

We present experimental results on VisDA17 in Table 5 in per-class accuracy. It can be seen that incorporating FAA outperforms state-of-the-art UDA methods consistently. This applies to UDA methods that employ stronger backbones ResNet-152 [59, 69].

Table 6 shows domain adaptive image classification experiments over Office-31 (all using the same backbone ResNet-50). We can see that incorporating FAA leads to robust domain adaptation and improve the image classification consistently by large margins.

### 4.6. Discussion

**Qualitative illustration of domain adaptation in segmentation** We compare our robust domain adaptation (RDA) with the recent domain adaptation method [75] qualitatively over semantic segmentation task. As Fig. 3 shows, the proposed RDA outperforms the baselines clearly.

**Overfitting Mitigation:** We compared FAA with existing overfitting mitigation methods. Most existing methods address overfitting through certain network regularization by noise injection to hidden units (Dropout), label-dropout (Label smooth), gradient ascent (Flooding), data and label mixing (Mixup), gradient based adversarial attacking (FGSM), virtual-label based adversarial attacking (VAT), etc. Table 7 shows experimental results over the task GTA → Cityscapes. It can be seen that existing regularization does not perform well in the domain adaptation task. The major reason is that existing methods were designed for supervised and semi-supervised learning where training and test data usually have little domain gap. The proposed FAA mitigates overfitting with clear performance gains as it allows large magnitude of perturbation noises which is critical to the effectiveness of its generated adversarial samples due to the existence of ‘domain gaps’ in UDA.

Due to the space limit, we provide more visualization of
Table 5. Experimental results of domain adaptive image classification task on VisDA17 (+FAA means to include FAA to attack supervised source loss and unsupervised target loss in domain adaptation).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>55.1</td>
<td>87.1</td>
<td>81.9</td>
<td>80.3</td>
<td>87.0</td>
<td>87.8</td>
<td>94.3</td>
<td>87.2</td>
<td>90.6</td>
<td>87.1</td>
</tr>
<tr>
<td></td>
<td>53.3</td>
<td>63.0</td>
<td>77.7</td>
<td>75.5</td>
<td>60.9</td>
<td>79.5</td>
<td>82.3</td>
<td>78.2</td>
<td>80.3</td>
<td>65.0</td>
</tr>
<tr>
<td></td>
<td>61.9</td>
<td>76.5</td>
<td>82.8</td>
<td>75.8</td>
<td>83.7</td>
<td>83.7</td>
<td>73.5</td>
<td>71.4</td>
<td>80.3</td>
<td>76.1</td>
</tr>
<tr>
<td></td>
<td>59.1</td>
<td>42.0</td>
<td>81.2</td>
<td>48.3</td>
<td>64.0</td>
<td>65.3</td>
<td>47.2</td>
<td>67.9</td>
<td>72.1</td>
<td>55.4</td>
</tr>
<tr>
<td></td>
<td>80.6</td>
<td>90.3</td>
<td>29.5</td>
<td>77.9</td>
<td>89.9</td>
<td>92.3</td>
<td>78.9</td>
<td>79.2</td>
<td>82.8</td>
<td>55.4</td>
</tr>
<tr>
<td></td>
<td>17.9</td>
<td>42.9</td>
<td>28.6</td>
<td>27.3</td>
<td>79.6</td>
<td>61.8</td>
<td>49.2</td>
<td>73.2</td>
<td>82.8</td>
<td>52.4</td>
</tr>
<tr>
<td></td>
<td>79.7</td>
<td>85.9</td>
<td>51.9</td>
<td>69.7</td>
<td>84.7</td>
<td>88.9</td>
<td>75.1</td>
<td>87.7</td>
<td>60.0</td>
<td>61.1</td>
</tr>
<tr>
<td></td>
<td>31.2</td>
<td>53.1</td>
<td>54.6</td>
<td>40.2</td>
<td>76.9</td>
<td>86.6</td>
<td>79.7</td>
<td>85.3</td>
<td>68.5</td>
<td>57.0</td>
</tr>
<tr>
<td></td>
<td>81.0</td>
<td>49.7</td>
<td>82.8</td>
<td>46.5</td>
<td>86.4</td>
<td>40.3</td>
<td>83.0</td>
<td>79.3</td>
<td>81.1</td>
<td>61.0</td>
</tr>
<tr>
<td></td>
<td>26.5</td>
<td>36.3</td>
<td>28.2</td>
<td>46.6</td>
<td>88.6</td>
<td>25.8</td>
<td>50.3</td>
<td>16.0</td>
<td>50.3</td>
<td>57.0</td>
</tr>
<tr>
<td></td>
<td>73.5</td>
<td>85.8</td>
<td>78.2</td>
<td>79.3</td>
<td>83.0</td>
<td>83.2</td>
<td>72.9</td>
<td>72.9</td>
<td>61.1</td>
<td>79.7</td>
</tr>
<tr>
<td></td>
<td>8.5</td>
<td>20.7</td>
<td>7.8</td>
<td>16.0</td>
<td>25.8</td>
<td>25.8</td>
<td>50.3</td>
<td>57.0</td>
<td>77.1</td>
<td>57.0</td>
</tr>
</tbody>
</table>

Table 6. Experimental results of domain adaptive image classification over Office-31 (+FAA means to include FAA to attack supervised source loss and unsupervised target loss in domain adaptation).

<table>
<thead>
<tr>
<th>Method</th>
<th>A→W</th>
<th>D→W</th>
<th>W→D</th>
<th>A→D</th>
<th>D→A</th>
<th>W→A</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 [21]</td>
<td>68.4</td>
<td>96.7</td>
<td>99.3</td>
<td>68.9</td>
<td>62.5</td>
<td>60.7</td>
<td>76.4</td>
</tr>
<tr>
<td>DAN [45]</td>
<td>80.5</td>
<td>97.1</td>
<td>99.6</td>
<td>78.6</td>
<td>63.6</td>
<td>62.8</td>
<td>80.4</td>
</tr>
<tr>
<td>RTN [46]</td>
<td>84.5</td>
<td>96.8</td>
<td>99.4</td>
<td>77.5</td>
<td>66.2</td>
<td>64.8</td>
<td>81.6</td>
</tr>
<tr>
<td>DANN [12]</td>
<td>82.0</td>
<td>96.9</td>
<td>99.1</td>
<td>79.7</td>
<td>68.2</td>
<td>67.4</td>
<td>82.2</td>
</tr>
<tr>
<td>ADDA [75]</td>
<td>86.2</td>
<td>96.2</td>
<td>98.4</td>
<td>77.8</td>
<td>69.5</td>
<td>68.9</td>
<td>82.9</td>
</tr>
<tr>
<td>JAN [47]</td>
<td>85.4</td>
<td>97.4</td>
<td>99.8</td>
<td>84.7</td>
<td>68.6</td>
<td>70.0</td>
<td>84.3</td>
</tr>
<tr>
<td>GTA [69]</td>
<td>89.5</td>
<td>97.9</td>
<td>99.8</td>
<td>87.7</td>
<td>72.8</td>
<td>71.4</td>
<td>86.5</td>
</tr>
<tr>
<td>CBST [99]</td>
<td>87.8</td>
<td>98.5</td>
<td>100</td>
<td>66.5</td>
<td>71.2</td>
<td>30.9</td>
<td>85.8</td>
</tr>
<tr>
<td>CBST+FAA</td>
<td>90.9</td>
<td>98.9</td>
<td>99.8</td>
<td>92.5</td>
<td>76.7</td>
<td>76.2</td>
<td>89.1</td>
</tr>
<tr>
<td>TAT [42]</td>
<td>92.5</td>
<td>99.3</td>
<td>100.4</td>
<td>97.2</td>
<td>73.1</td>
<td>72.1</td>
<td>88.4</td>
</tr>
<tr>
<td>TAT+FAA</td>
<td>92.4</td>
<td>99.8</td>
<td>99.4</td>
<td>94.7</td>
<td>78.2</td>
<td>77.1</td>
<td>90.3</td>
</tr>
<tr>
<td>CRST [98]</td>
<td>89.4</td>
<td>98.9</td>
<td>100</td>
<td>68.7</td>
<td>72.6</td>
<td>30.9</td>
<td>86.8</td>
</tr>
<tr>
<td>CRST+FAA</td>
<td>92.3</td>
<td>99.2</td>
<td>99.7</td>
<td>94.4</td>
<td>80.5</td>
<td>78.7</td>
<td>90.8</td>
</tr>
</tbody>
</table>

Table 7. Comparison with existing overfitting mitigation methods: For the semantic segmentation task GTA → Cityscapes, FAA performs the best consistently by large margins.

<table>
<thead>
<tr>
<th>Method</th>
<th>mIoU gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST [99]</td>
<td>-42.6</td>
</tr>
<tr>
<td>+Dropout [71]</td>
<td>43.1</td>
</tr>
<tr>
<td>+Label smooth [72]</td>
<td>43.4</td>
</tr>
<tr>
<td>+Mixup [93]</td>
<td>43.6</td>
</tr>
<tr>
<td>+FGSM [14]</td>
<td>43.9</td>
</tr>
<tr>
<td>+VAT [51]</td>
<td>44.3</td>
</tr>
<tr>
<td>+Flooding [31]</td>
<td>44.1</td>
</tr>
<tr>
<td>+FAA</td>
<td>50.1</td>
</tr>
</tbody>
</table>

5. Conclusion

In this work, we presented RDA, a robust domain adaptation technique that mitigates overfitting in UDA via a novel Fourier adversarial attacking (FAA). We achieve robust domain adaptation by a novel Fourier adversarial attacking (FAA) method that allows large magnitude of perturbation noises but has minimal modification of image semantics. With FAA-generated adversarial samples, the training can continue the ‘random walk’ and drift into an area with a flat loss landscape, leading to more robust domain adaptation. Extensive experiments over multiple domain adaptation tasks show that RDA can work with different computer vision tasks (i.e., segmentation, detection and classification) with superior performance. We will explore disentanglement-based adversarial attacking and its applications to other computer vision tasks. We will also study how FAA could mitigate over-fitting in classical supervised learning and semi-supervised learning as well as the recent contrast-based unsupervised representation learning.

Acknowledgement

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU).
References


on Computer Vision and Pattern Recognition, pages 6891–6902, 2021. 3, 4, 6


[61] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In European conference on computer vision, pages 102–118. Springer, 2016. 5


[75] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7472–7481, 2018. 1, 2, 5, 6, 7, 8


