# Open Set Domain Adaptation

Pau Panareda Busto[1,2]
[1]Airbus Group Innovations
Munich, Germany
pau.panareda-busto@airbus.com

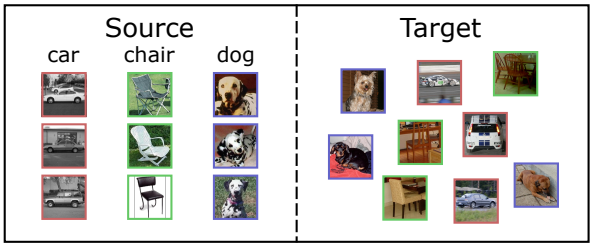
Juergen Gall[2]
[2]Computer Vision Group
University of Bonn, Germany
gall@iai.uni-bonn.de

## Abstract

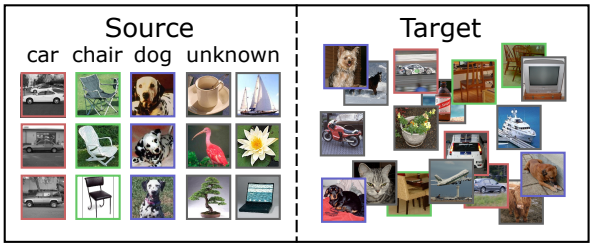
*When the training and the test data belong to different domains, the accuracy of an object classifier is significantly reduced. Therefore, several algorithms have been proposed in the last years to diminish the so called domain shift between datasets. However, all available evaluation protocols for domain adaptation describe a closed set recognition task, where both domains, namely source and target, contain exactly the same object classes. In this work, we also explore the field of domain adaptation in open sets, which is a more realistic scenario where only a few categories of interest are shared between source and target data. Therefore, we propose a method that fits in both closed and open set scenarios. The approach learns a mapping from the source to the target domain by jointly solving an assignment problem that labels those target instances that potentially belong to the categories of interest present in the source dataset. A thorough evaluation shows that our approach outperforms the state-of-the-art.*

## 1. Introduction

For many applications, training data is scarce due to the high cost of acquiring annotated training data. Although there are large annotated image datasets publicly available, the images collected from the Internet often differ from the type of images which are relevant for a specific application. Depending on the application, the type of sensor or the perspective of the sensor, the entire captured scene might greatly differ from pictures on the Internet. The two types of images are therefore in two different domains, namely the source and target domain. In order to classify the images in the target domain using the annotated images in the source domain, the source and target domains can be aligned. In our case, we will map the feature space of the source domain to the feature space of the target domain. Any classifier can then be learned on the transformed data of the source domain to classify the images in the target domain. This process is termed domain adaptation and is further divided in unsupervised and semi-supervised approaches depending on whether the target images are unlabelled or partially labelled.




(a) Closed set domain adaptation




(b) Open set domain adaptation

Figure 1. (a) Standard domain adaptation benchmarks assume that source and target domains contain images only of the same set of object classes. This is denoted as *closed set domain adaptation* since it does not include images of unknown classes or classes which are not present in the other domain. (b) We propose *open set domain adaptation*. In this setting, both source and target domain contain images that do not belong to the classes of interest. Furthermore, the target domain contains images that are not related to any image in the source domain and vice versa.

Besides of the progress we have seen for domain adaptation over the last years [34, 19, 18, 9, 21, 13, 31, 15], the methods have been so far evaluated using a setting where the images of the source and target domain are from the same set of categories. This setting can be termed *closed set domain adaptation* as illustrated in Fig. 1(a). An example of such a *closed set* protocol is the popular *Office* dataset [34]. The assumption that the target domain contains only images of the categories of the source domain is, however, unrealistic. For most applications, the dataset in the target domain contains many images and only a small portion of it might

belong to the classes of interest. We therefore introduce the concept of *open sets* [28, 37, 36] to the domain adaptation problem and propose *open set domain adaptation*, which avoids the unrealistic assumptions of *closed set domain adaptation*. The differences between closed and open set domain adaptation are illustrated in Fig. 1.

As a second contribution, we propose a domain adaptation method that suits both closed and open sets. To this end, we map the feature space of the source domain to the target domain. The mapping is estimated by assigning images in the target domain to some categories of the source domain. The assignment problem is defined by a binary linear program that also includes an implicit outlier handling, which discards images that are not related to any image in the source domain. An overview of the approach is given in Fig. 2. The approach can be applied to the unsupervised or semi-supervised setting, where a few images in the target domain are annotated by a known category.

We provide a thorough evaluation and comparison with state-of-the-art methods on 24 combinations of source and target domains including the *Office* dataset [34] and the *Cross-Dataset Analysis* [44]. We revisit these evaluation datasets and propose a new open set protocol for domain adaptation, both unsupervised and semi-supervised, where our approach achieves state-of-the-art results in all settings.

## 2. Related Work

The interest in studying domain adaptation techniques for computer vision problems increased with the release of a benchmark by Saenko et al. [34] for domain adaptation in the context of object classification. The first relevant works on unsupervised domain adaptation for object categorisation were presented by Golapan et al. [19] and Gong et al. [18], who proposed an alignment in a common subspace of source and target samples using the properties of Grassmanian manifolds. Jointly transforming source and target domains into a common low dimensional space was also done together with a conjugate gradient minimisation of a transformation matrix with orthogonality constraints [3] and with dictionary learning to find subspace interpolations [32, 38, 47]. Sun et al. [40, 39] presented a very efficient solution based on second-order statistics to align a source domain with a target domain. Similarly, Csurka et al. [10] jointly denoise source and target samples to reconstruct data without partial random corruption. Sharing certain similarities with associations between domains, Gong et al. [17] minimise the Maximum Mean Discrepancy (MMD) [20] of two datasets. They assign instances to latent domains and solve it by a relaxed binary optimisation. Hsu et al. [31] use a similar idea allowing instances to be linked to all other samples.

Semi-supervised domain adaptation approaches take advantage of knowing the class labels of a few target samples. Aytar et al. [2] proposed a transfer learning formulation to regularise the training of target classifiers. Exploiting pairwise constraints across domains, Saenko et al. [34] and Kulis et al. [27] learn a transformation to minimise the effect of the domain shift while also training target classifiers. Following the same idea, Hoffman et al. [22] considered an iterative process to alternatively minimise the classification weights and the transformation matrix. In a different context, [7] proposed a weakly supervised approach to refine coarse viewpoint annotations of real images by synthetic images. In contrast to semi-supervised approaches, the task of viewpoint refinement assumes that all images in the target domain are labelled but not with the desired granularity.

The idea of selecting the most relevant information of each domain has been studied in early domain adaptation methods in the context of natural language processing [5]. Pivot features that behave the same way for discriminative learning in both domains were selected to model their correlations. Gong et al. [16] presented an algorithm that selects a subset of source samples that are distributed most similarly to the target domain. Another technique that deals with instance selection has been proposed by Sangineto et al. [35]. They train weak classifiers on random partitions of the target domain and evaluate them in the source domain. The best performing classifiers are then selected. Other works have also exploited greedy algorithms that iteratively add target samples to the training process, while the least relevant source samples are removed [6, 42].

Since CNN features show some robustness to domain changes [11], several domain adaptation approaches based on CNNs have been proposed [39, 31, 45, 48]. Chopra et al. [9] extended the joint training of CNNs with source and target images by learning intermediate feature encoders and combine them to train a deep regressor. The MMD distance has been also proposed as regulariser to learn features for source and target samples jointly [14, 46, 29, 30]. Ganin et al. [13] added a domain classifier network after the CNN to minimise the domain loss together with the classification loss. More recently, Ghifary et al. [15] combined two CNN models for labelled source data classification and for unsupervised target data reconstruction.

Standard object classification tasks ignore the impact of impostors that are not represented by any of the object categories. These *open sets* started getting attention in face recognition tasks, where some test exemplars did not appear in the training database and had to be rejected [28]. Current techniques to detect unrelated samples in multi-class recognition with *open sets* have lately been revisited by Scheirer et al. [37]. [23] and [36] detect unknown instances by learning SVMs that assign probabilistic decision scores instead of class labels. Similarly, [49] and [4] add a regulariser to detect outliers and penalise a misclassification.

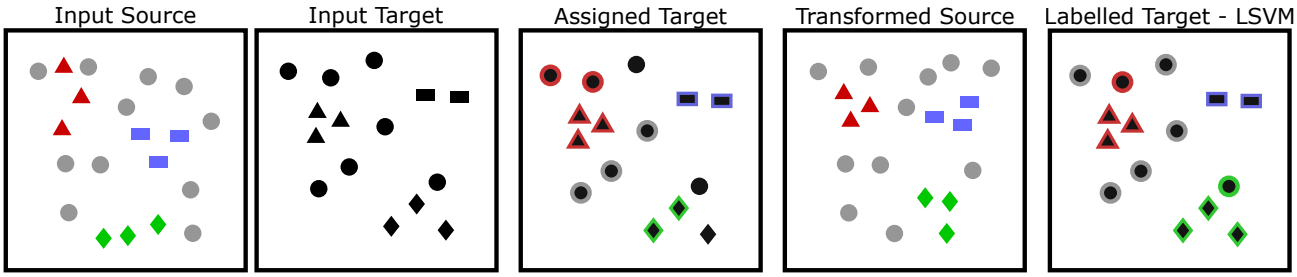



Figure 2. Overview of the proposed approach for unsupervised *open set domain adaptation*. (a) The source domain contains some labelled images, indicated by the colours red, blue and green, and some images belonging to unknown classes (grey). For the target domain, we do not have any labels but the shapes indicate if they belong to one of the three categories or an unknown category (circle). (b) In the first step, we assign class labels to some target samples, leaving outliers unlabelled. (c) By minimising the distance between the samples of the source and the target domain that are labelled by the same category, we learn a mapping from the source to the target domain. The image shows the samples in the source domain after the transformation. This process iterates between (b) and (c) until it converges to a local minimum. (d) In order to label all samples in the target domain either by one of the three classes (red, green, blue) or as unknown (grey), we learn a classifier on the source samples that have been mapped to the target domain (c) and apply it to the samples of the target domain (a). In this image, two samples with unknown classes are wrongly classified as red or green.

## 3. Open Set Domain Adaptation

We present in this paper an approach that iterates between solving the labelling problem of target samples, *i.e.*, associating a subset of the target samples to the known categories of the source domain, and computing a mapping from the source to the target domain by minimising the distances of the assignments. The transformed source samples are then used in the next iteration to re-estimate the assignments and update the transformation. This iterative process is repeated until convergence and is illustrated in Fig. 2.

In Section 3.1, we describe the unsupervised assignment of target samples to categories of the source domain. The semi-supervised case is described in Section 3.2. Section 3.3 finally describes how the mapping from the source domain to the target domain is estimated from the previous assignments. This part is the same for the unsupervised and semi-supervised setting.

### 3.1. Unsupervised Domain Adaptation

We first address the problem of unsupervised domain adaptation, *i.e.*, none of the target samples are annotated, in an open set protocol. Given a set of classes $\mathcal{C}$ in the source domain, including $|\mathcal{C}-1|$ known classes and an additional unknown class that gathers all instances from other irrelevant categories, we aim to label the target samples $\mathcal{T} = \{T_1, \dots, T_{|\mathcal{T}|}\}$ by a class $c \in \mathcal{C}$. We define the cost of assigning a target sample $T_t$ to a class $c$ by $d_{ct} = \|S_c - T_t\|_2^2$ where $T_t \in \mathbb{R}^D$ is the feature representation of the target sample $t$ and $S_c \in \mathbb{R}^D$ is the mean of all samples in the source domain labelled by class $c$. To increase the robustness of the assignment, we do not enforce that all target samples are assigned to a class as shown in Fig. 2(b). The cost of declaring a target sample as outlier is defined by a parameter $\lambda$, which is discussed in Section 4.1.

Having defined the individual assignment costs, we can formulate the entire assignment problem by:

$$\begin{aligned} \underset{x_{ct},o_t}{\text{minimise}} \quad & \sum_t \left( \sum_c d_{ct} x_{ct} + \lambda o_t \right) & \\ \text{subject to} \quad & \sum_c x_{ct} + o_t = 1 & \forall t\,, \\ & \sum_t x_{ct} \geq 1 & \forall c\,, \\ & x_{ct}, o_t \in \{0,1\} & \forall c,t\,. \end{aligned} \tag{1}$$

By minimising the constrained objective function, we obtain the binary variables $x_{ct}$ and $o_t$ as solution of the assignment problem. The first type of constraints ensures that a target sample is either assigned to one class, *i.e.*, $x_{ct} = 1$, or declared as outlier, *i.e.*, $o_t = 1$. The second type of constraints ensures that at least one target sample is assigned to each class $c \in \mathcal{C}$. We use the constraint integer program package SCIP [1] to solve all proposed formulations.

As it is shown in Fig. 2(b), we label the targets also by the unknown class. Note that the unknown class combines all objects that are not of interest. Even if the unknowns in the source and target domain belong to different semantic classes, a target sample might be closer to the mean of all negatives than to any other positive class. In this case, we can confidentially label a target sample as unknown. In our experiments, we show that it makes not much difference if the unknown class is included in the unsupervised setting since the outlier handling discards target samples that are not close to the mean of negatives.

### 3.2. Semi-supervised Domain Adaptation

The unsupervised assignment problem naturally extends to a semi-supervised setting when a few target samples are annotated. In this case, we only have to extend the formulation (1) by additional constraints that enforce that the annotated target samples do not change the label, *i.e.*,

$$x_{\hat{c}_t t} = 1 \qquad \forall (t, \hat{c}_t) \in \mathcal{L}, \quad (2)$$

where $\mathcal{L}$ denotes the set of labelled target samples and $\hat{c}_t$ the class label provided for target sample $t$. In order to exploit the labelled target samples better, one can use the neighbourhood structure in the source and target domain. While the constraints remain the same, the objective function (1) can be changed to

$$\sum_t \left( \sum_c x_{ct} \left( d_{ct} + \sum_{t' \in N_t} \sum_{c'} d_{cc'} x_{c't'} \right) + \lambda o_t \right), \quad (3)$$

where $d_{cc'} = \|S_c - S_{c'}\|_2^2$. While in (1) the cost of labelling a target sample $t$ by the class $c$ is given only by $d_{ct}$, a second term is added in (3). It is computed over all neighbours $N_t$ of $t$ and adds the distance between the classes in the source domain as additional cost if a neighbour is assigned to another class than the target sample $t$.

The objective function (3), however, becomes quadratic and therefore NP-hard to solve. Thus, we transform the *quadratic assignment problem* into a mixed 0-1 linear program using the Kaufman and Broeckx linearisation [25]. By substituting

$$w_{ct} = x_{ct} \left( \sum_{t' \in N_t} \sum_{c'} x_{c't'} d_{cc'} \right), \quad (4)$$

we derive to the linearised problem

$$\begin{aligned}
\underset{x_{ct}, w_{ct}, o_t}{\text{minimise}} \quad & \sum_t \left( \sum_c d_{ct} x_{ct} + \sum_c w_{ct} + \lambda o_t \right) \\
\text{subject to} \quad & \sum_c x_{ct} + o_t = 1 && \forall t\,, \\
& \sum_t x_{ct} \geq 1 && \forall c\,, \\
& a_{ct} x_{ct} + \sum_{t' \in N_t} \sum_{c'} d_{cc'} x_{c't'} - w_{ct} \leq a_{ct} && \forall s, t\,, \\
& x_{ct}, o_t \in \{0, 1\} && \forall c, t\,, \\
& w_{ct} \geq 0 && \forall c, t\,,
\end{aligned} \quad (5)$$

where $a_{ct} = \sum_{t' \in N_t} \sum_{c'} d_{cc'}$.

### 3.3. Mapping

As illustrated in Fig. 2, we iterate between solving the assignment problem, as described in Section 3.1 or 3.2, and estimating the mapping from the source domain to the target domain. We consider a linear transformation, which is represented by a matrix $W \in \mathbb{R}^{D \times D}$. We estimate $W$ by minimising the following loss function:

$$f(W) = \frac{1}{2} \sum_t \sum_c x_{ct} \|W S_c - T_t\|_2^2\,, \quad (6)$$

which we can rewrite in matrix form:

$$f(W) = \frac{1}{2} ||W P_S - P_T||_F^2\,. \quad (7)$$

The matrices $P_S$ and $P_T \in \mathbb{R}^{DxL}$ with $L = \sum_t \sum_c x_{ct}$ represent all assignments, where the columns denote the actual associations. The quadratic nature of the convex objective function may be seen as a linear least squares problem, which can be easily solved by any available QP solver. State-of-the-art features based on convolutional neural networks, however, are high dimensional and the number of target instances is usually very large. We use therefore nonlinear optimisation [41, 24] to optimise $f(W)$. The derivatives of (6) are given by

$$\frac{\partial f(W)}{\partial W} = W(P_S P_S^T) - P_T P_S^T\,. \quad (8)$$

If $L < D$, *i.e.*, the number of samples, which have been assigned to a known class, is smaller than the dimensionality of the features, the optimisation also deals with an underdetermined linear least squares formulation. In this case, the solver converges to the matrix $W$ with the smallest norm, which is still a valid solution.

After the transformation $W$ is estimated, we map the source samples to the target domain. We therefore iterate the process of solving the assignment problem and estimating the mapping from the source domain to the target domain until it converges. After the approach has converged, we train linear SVMs in a one-vs-one setting on the transformed source samples. For the semi-supervised setting, we also include the annotated target samples $\mathcal{L}$ (2) to the training set. The linear SVMs are then used to obtain the final labelling of the target samples as illustrated in Fig. 2(d).

## 4. Experiments

We evaluate our method in the context of domain adaptation for object categorisation. In this setting, the images of the source domain are annotated by class labels and the goal is to classify the images in the target domain. We report the accuracies for both unsupervised and semi-supervised scenarios, where target samples are unlabelled or partially labelled, respectively. For consistency, we use *libsvm* [8] since it has also been used in other works, *e.g.*, [12] and [39]. We set the misclassification parameter $C = 0.001$ in all experiments, which allows for a soft margin optimisation that works best in such classification tasks [12, 39].

### 4.1. Parameter configuration

Our algorithm contains a few parameters that need to be defined. For the outlier rejection, we use

$$\lambda = 0.5\big(\max_{t,c} d_{ct} + \min_{t,c} d_{ct}\big), \tag{9}$$

*i.e.*, $\lambda$ is adapted automatically based on the distances $d_{ct}$, since higher values closer to the largest distance barely discard any outlier and lower values almost reject all assignments. We iterate the approach until the maximum number of 10 iterations is reached or if the distance

$$\sqrt{\sum_c \sum_t x_{ct} \left|S_{c,k} - T_t\right|^2} \tag{10}$$

is below $\epsilon = 0.01$ where $S_{c,k}$ corresponds to the transformed class mean at iteration $k$. In practice, the process converges after 3-5 iterations.

### 4.2. Office dataset

We evaluate and compare our approach on the *Office* dataset [34], which is the standard benchmark for domain adaptation with CNN features. It provides three different domains, namely *Amazon (A)*, *DSLR (D)* and *Webcam (W)*. While the *Amazon* dataset contains centred objects on white background, the other two comprise pictures taken in an office environment but with different quality levels. In total, there are 31 common classes for 6 source-target combinations. This means that there are 4 combinations with a considerable domain shift (A $\rightarrow$ D, A $\rightarrow$ W, D $\rightarrow$ A, W $\rightarrow$ A) and 2 with a minor domain shift (D $\rightarrow$ W, W $\rightarrow$ D).

We introduce an open set protocol for this dataset by taking the 10 classes that are also common in the *Caltech* dataset [18] as shared classes. In alphabetical order, the classes 11-20 are used as unknowns in the source domain and 21-31 as unknowns in the target domain, *i.e.*, the unknown classes in the source and target domain are not shared. For evaluation, each sample in the target domain needs to be correctly classified either by one of the 10 shared classes or as unknown. In order to compare with a closed setting (CS), we report the accuracy when source and target domain contain only samples of the 10 shared classes. Since OS is evaluated on all target samples, we also report the numbers when the accuracy is only measured on the same target samples as CS, *i.e.*, only for the shared 10 classes. The latter protocol is denoted by OS$^*$(10) and provides a direct comparison to CS(10). Additional results for the closed setting with all classes are reported in the supplementary material.

**Unsupervised domain adaptation** We firstly compare the accuracy of our method in the unsupervised set-up with state-of-the-art domain adaptation techniques embedded in the training of CNN models. DAN [29] retrains the AlexNet model by freezing the first 3 convolutional layers, finetuning the last 2 and learning the weights from each fully connected layer by also minimising the discrepancy between both domains. RTN [30] extends DAN by adding a residual transfer module that bridges the source and target classifiers. BP [13] trains a CNN for domain adaptation by a gradient reversal layer and minimises the domain loss jointly with the classification loss. For training, we use all samples per class as proposed in [17], which is the standard protocol for CNNs on this dataset. As proposed in [13], we use for all methods linear SVMs for classification instead of the soft-max layer for a fair comparison.

| | A→D | | | A→W | | |
|---|---|---|---|---|---|---|
| | CS (10) | OS$^*$ (10) | OS (10) | CS (10) | OS$^*$ (10) | OS (10) |
| LSVM | 87.1 | 70.7 | 72.6 | 77.5 | 53.9 | 57.5 |
| DAN [29] | 88.1 | 76.5 | 77.6 | **90.5** | 70.2 | 72.5 |
| RTN [30] | **93.0** | 74.7 | 76.6 | 87.0 | 70.8 | 73.0 |
| BP [13] | 91.9 | 77.3 | 78.3 | 89.2 | 73.8 | 75.9 |
| ATI | 92.4 | 78.2 | 78.8 | 85.1 | **77.7** | **78.4** |
| ATI-$\lambda$ | **93.0** | **79.2** | **79.8** | 84.0 | 76.5 | 77.6 |
| ATI-$\lambda$-N1 | 91.9 | 78.3 | 78.9 | 84.6 | 74.2 | 75.6 |
| | D→A | | | D→W | | |
| | CS (10) | OS$^*$ (10) | OS (10) | CS (10) | OS$^*$ (10) | OS (10) |
| LSVM | 79.4 | 40.0 | 45.1 | 97.9 | 87.5 | 88.5 |
| DAN [29] | 83.4 | 53.5 | 57.0 | 96.1 | 87.5 | 88.4 |
| RTN [30] | 82.8 | 53.8 | 57.2 | 97.9 | 88.1 | 89.0 |
| BP [13] | 84.3 | 54.1 | 57.6 | 97.5 | 88.9 | 89.8 |
| ATI | 93.4 | **70.0** | 71.1 | **98.5** | 92.2 | 92.6 |
| ATI-$\lambda$ | **93.8** | **70.0** | **71.3** | **98.5** | 93.2 | 93.5 |
| ATI-$\lambda$-N1 | 93.3 | 65.6 | 67.8 | 97.9 | **94.0** | **94.4** |

| | W→A | | | W→D | | | AVG. | | |
|---|---|---|---|---|---|---|---|---|---|
| | CS (10) | OS$^*$ (10) | OS (10) | CS (10) | OS$^*$ (10) | OS (10) | CS | OS$^*$ | OS |
| LSVM | 80.0 | 44.9 | 49.2 | **100** | 96.5 | 96.6 | 87.0 | 65.6 | 68.3 |
| DAN [29] | 84.9 | 58.5 | 60.8 | **100** | 97.5 | 98.3 | 90.5 | 74.0 | 75.8 |
| RTN [30] | 85.1 | 60.2 | 62.4 | **100** | 98.3 | **98.8** | 91.0 | 74.3 | 76.2 |
| BP [13] | 86.2 | 61.8 | 64.0 | **100** | 98.0 | 98.7 | 91.6 | 75.7 | 77.4 |
| ATI | 93.4 | 76.4 | 76.6 | **100** | 99.1 | 98.3 | **93.8** | 82.1 | 82.6 |
| ATI-$\lambda$ | **93.7** | **76.5** | **76.7** | **100** | 99.2 | 98.3 | 93.7 | **82.4** | **82.9** |
| ATI-$\lambda$-N1 | 93.4 | 71.6 | 72.4 | **100** | **99.6** | **98.8** | 93.5 | 80.6 | 81.3 |

Table 1. Open set domain adaptation on the unsupervised Office dataset with 10 shared classes (OS) using all samples per class [17]. For comparison, results for closed set domain adaptation (CS) and modified open set (OS$^*$) are reported.

To analyse the formulations that are discussed in Section 3, we compare several variants: ATI (*Assign-and-Transform-Iteratively*) denotes our formulation in (1) assigning a source class to all target samples, *i.e.*, $\lambda = \infty$. Then, ATI-$\lambda$ includes the outlier rejection and ATI-$\lambda$-$N_1$ is the unsupervised version of the locality constrained formulation corresponding to (3) with 1 nearest neighbour. In addition, we denote LSVM as the linear SVMs trained on the source domain without any domain adaptation.

The results of these techniques using the described open set protocol are shown in Table 1. Our approach ATI improves over the baseline without domain adaptation (LSVM) by +6.8% for CS and +14.3% for OS. The improvement is larger for the combinations that have larger domain shifts, *i.e.* with *Amazon*. We also observe that ATI

| | A→D | | | A→W | | |
|---|---|---|---|---|---|---|
| | CS (10) | OS* (10) | OS (10) | CS (10) | OS* (10) | OS (10) |
| LSVM | 84.4±5.9 | 63.7±6.7 | 66.6±5.9 | 76.5±2.9 | 48.2±4.8 | 52.5±4.2 |
| TCA [33] | 85.9±6.3 | 75.5±6.6 | 75.7±5.9 | 80.4±6.9 | 67.0±5.9 | 67.9±5.5 |
| gfk [18] | 84.8±5.1 | 68.6±6.7 | 70.4±6.0 | 76.7±3.1 | 54.1±4.8 | 57.4±4.2 |
| SA [12] | 84.0±3.4 | 71.5±5.9 | 72.6±5.3 | 76.6±2.8 | 57.4±4.2 | 60.1±3.7 |
| CORAL [39] | 85.8±7.2 | 79.9±5.7 | 79.6±5.0 | 81.9±2.8 | 68.1±3.6 | 69.3±3.1 |
| ATI | **91.4±1.3** | 80.5±2.0 | 81.1±2.8 | **86.1±1.1** | 73.4±2.0 | **75.3±1.7** |
| ATI-$\lambda$ | 91.1±2.1 | **81.1±0.4** | **82.2±2.0** | 85.5±2.1 | **73.7±2.6** | **75.3±1.4** |

| | D→A | | | D→W | | |
|---|---|---|---|---|---|---|
| | CS (10) | OS* (10) | OS (10) | CS (10) | OS* (10) | OS (10) |
| LSVM | 75.5±2.1 | 36.1±3.7 | 42.2±3.3 | 96.2±1.0 | 81.5±1.5 | 83.1±1.3 |
| TCA [33] | 88.2±1.5 | 71.8±2.5 | 71.8±2.0 | 97.8±0.5 | 92.0±0.9 | 91.5±1.0 |
| gfk [18] | 79.7±1.0 | 45.3±3.7 | 49.7±3.4 | 96.3±0.9 | 85.1±2.7 | 86.2±2.4 |
| SA [12] | 81.7±0.7 | 52.5±3.0 | 55.8±2.7 | 96.3±0.8 | 86.8±2.5 | 87.7±2.3 |
| CORAL [39] | 89.6±1.0 | 66.6±2.8 | 68.2±2.5 | 97.2±0.7 | 91.1±1.7 | 91.4±1.5 |
| ATI | 93.5±0.3 | 69.8±1.4 | 70.8±2.1 | 97.3±0.5 | 89.6±2.1 | 90.3±1.8 |
| ATI-$\lambda$ | **93.9±0.4** | **71.1±0.9** | **72.0±0.5** | **97.5±1.1** | **92.1±1.3** | **92.5±0.7** |

| | W→A | | | W→D | | | AVG. | | |
|---|---|---|---|---|---|---|---|---|---|
| | CS (10) | OS* (10) | OS (10) | CS (10) | OS* (10) | OS (10) | CS | OS* | OS |
| LSVM | 72.5±2.7 | 34.3±4.9 | 39.9±4.4 | 99.1±0.5 | 89.8±1.5 | 90.5±1.3 | 84.1 | 58.9 | 62.5 |
| TCA | 85.5±3.3 | 68.1±5.1 | 68.6±4.6 | 98.8±0.9 | 94.1±2.9 | 93.6±2.6 | 89.5 | 78.1 | 78.2 |
| gfk | 75.0±2.9 | 43.2±5.1 | 47.6±4.6 | 99.0±0.5 | 92.0±1.5 | 92.2±1.4 | 85.2 | 64.7 | 67.3 |
| SA | 76.5±3.2 | 49.7±5.1 | 53.0±4.6 | 98.8±0.7 | 92.4±2.9 | 92.4±2.8 | 85.7 | 68.4 | 70.3 |
| CORAL | 86.9±1.9 | 63.9±4.9 | 65.6±4.3 | **99.2±0.7** | 96.0±2.1 | 95.0±2.0 | 90.1 | 77.6 | 78.2 |
| ATI | 92.2±1.1 | 75.1±1.7 | 76.0±2.0 | 98.9±1.3 | 95.5±2.3 | 95.4±2.1 | **93.2** | 80.7 | 81.5 |
| ATI-$\lambda$ | **92.4±1.1** | **75.4±1.8** | **76.4±1.8** | 98.9±1.3 | **96.5±2.1** | **95.8±1.8** | **93.2** | **81.5** | **82.3** |

Table 2. Open set domain adaptation on the unsupervised Office dataset with 10 shared classes (OS). We report the average and the standard deviation using a subset of samples per class in 5 random splits [34]. For comparison, results for closed set domain adaptation (CS) and modified open set (OS*) are reported.

outperforms all CNN-based domain adaptation methods for the closed (+2.2%) and open setting (+5.2%). It can also be observed that the accuracy for the open set is lower than for the closed set for all methods, but that our method handles the open set protocol best. While ATI-$\lambda$ does not obtain any considerable improvement compared to ATI in CS, the outlier rejection allows for an improvement in OS. The locality constrained formulation, ATI-$\lambda$-$N_1$, which we propose only for the semi-supervised setting, decreases the accuracy in the unsupervised setting.

Additionally, we report accuracies of popular domain adaptation methods that are not related to deep learning. We report the results of methods that transform the data to a common low dimensionality subspace, including Transfer Component Analysis (TCA) [33], Geodesic Flow Kernel (GFK) [18] and Subspace alignment (SA) [12]. In addition, we also include CORAL [39], which whitens and recolours the source towards the target data. Following the standard protocol of [34], we take 20 samples per object class when *Amazon* is used as source domain, and 8 for *DSLR* or *Webcam*. We extract feature vectors from the fully connected layer-7 (fc7) of the AlexNet model [26]. Each evaluation is executed 5 times with random samples from the source domain. The average accuracy and standard deviation of the five runs are reported in Table 2. The results are similar to the protocol reported in Table 1. Our approach ATI outperforms the other methods both for CS and OS and the additional outlier handling (ATI-$\lambda$) does not improve the accuracy for the closed set but for the open set.

**Impact of unknown class** The linear SVM that we employ in the open set protocol uses the unknown classes of the transformed source domain for the training. Since unknown object samples from the source domain are from different classes than the ones from the target domain, using an SVM that does not require any negative samples might be a better choice. Therefore, we compare the performance of a standard SVM classifier with a specific open set SVM (OS-SVM) [36], where only the 10 known classes are used for training. OS-SVM introduces an inclusion probability and labels target instances as unknown if this inclusion is not satisfied for any class. Table 3 compares the classification accuracies of both classifiers in the 6 domain shifts of the Office dataset. While the performance is comparable when no domain adaptation is applied, ATI-$\lambda$ obtains significantly better accuracies when the learning includes negative instances.

As discussed in Section 3.1, the unknown class is also part of the labelling set $\mathcal{C}$ for the target samples. The labelled target samples are then used to estimate the mapping $W$ (6). To evaluate the impact of including the unknown class, Table 4 compares the accuracy when the unknown class is not included in $\mathcal{C}$. Adding the unknown class improves the accuracy slightly since it enforces that the negative mean of the source is mapped to a negative sample in the target. The impact, however, is very small.

Additionally, we also analyse the impact of increasing the amount of unknown samples in both source and target domain on the configuration *Amazon* $\rightarrow$ *DSLR+Webcam*. Since the domain shift between *DSLR* and *Webcam* is close to zero (same scenario, but different cameras), they can be merged to get more unknown samples. Following the described protocol, we take 20 samples per known category, also in this case for the target domain, and we randomly increase the number of unknown samples from 20 to 400 in both domains at the same time. As shown in Table 5, that reports the mean accuracies of 5 random splits, adding more unknown samples decreases the accuracy if domain adaptation is not used (LSVM), but also for the domain adaption method CORAL [39]. This is expected since the unknowns are from different classes and the impact of the unknowns compared to the samples from the shared classes increases. Our method handles such an increase and the accuracies remain stable between 80.3% and 82.5%.

**Semi-supervised domain adaptation** We also evaluate our approach for open set domain adaptation on the *Office* dataset in its semi-supervised setting. Applying again the standard protocol of [34] with the subset of source samples, we also take 3 labelled target samples per class and leave the rest unlabelled. We compare our method with the

| | A→D | | A→W | | D→A | | D→W | | W→A | | W→D | | AVG. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OS-SVM | LSVM | OS-SVM | LSVM | OS-SVM | LSVM | OS-SVM | LSVM | OS-SVM | LSVM | OS-SVM | LSVM | OS-SVM | LSVM |
| No Adap. | 67.5 | 72.6 | 58.4 | 57.5 | 54.8 | 45.1 | 80.0 | 88.5 | 55.3 | 49.2 | 94.0 | 96.6 | 68.3 | 68.3 |
| ATI-$\lambda$ | 72.0 | **79.8** | 65.3 | **77.6** | 66.4 | **71.3** | 82.2 | **93.5** | 71.6 | **76.7** | 92.7 | **98.3** | 75.0 | **82.9** |

Table 3. Comparison of a standard linear SVM (LSVM) with a specific open set SVM (OS-SVM) [37] on the unsupervised Office dataset with 10 shared classes using all samples per class [17].

| | A→D | A→W | D→A | D→W | W→A | W→D | AVG. |
|---|---|---|---|---|---|---|---|
| | OS(10) | | | | | | |
| ATI-$\lambda$ ($\mathcal{C}$ w/o unknown) | 79.0 | 77.1 | 70.5 | 93.4 | 75.8 | 98.2 | 82.3 |
| ATI-$\lambda$ ($\mathcal{C}$ with unknown) | **79.8** | **77.6** | **71.3** | **93.5** | **76.7** | **98.3** | **82.9** |

Table 4. Impact of including the unknown class to the set of classes $\mathcal{C}$. The evaluation is performed on the unsupervised Office dataset with 10 shared classes using all samples per class [17].

| *number of unknowns* | 20 | 40 | 60 | 80 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|---|---|---|
| *unknown / known* | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 | 1.00 | 1.50 | 2.00 |
| LSVM | 74.2 | 70.0 | 66.2 | 63.4 | 61.4 | 53.9 | 50.4 | 48.2 |
| CORAL [39] | 77.2 | 76.4 | 76.2 | 74.8 | 73.7 | 71.5 | 70.8 | 69.7 |
| ATI-$\lambda$ | 80.3 | 82.4 | 81.2 | 81.7 | 82.5 | 80.9 | 80.7 | 81.9 |

Table 5. Impact of increasing the amount of unknown samples in the domain shift *Amazon* → *DSLR+Webcam* on the unsupervised Office dataset with 10 shared classes using 20 random samples per known class in both domains.

| | A→D | | | A→W | | |
|---|---|---|---|---|---|---|
| | CS (10) | OS* (10) | OS (10) | CS (10) | OS* (10) | OS (10) |
| LSVM (s) | 85.8±3.2 | 62.1±7.9 | 65.9±6.2 | 76.4±2.1 | 45.7±5.0 | 50.4±4.5 |
| LSVM (t) | 92.3±3.9 | 68.2±5.2 | 71.1±4.7 | 91.5±4.9 | 59.6±3.7 | 63.2±3.4 |
| LSVM (st) | 95.7±1.3 | 82.5±3.0 | 84.0±2.6 | 92.4±1.8 | 72.5±3.7 | 74.8±3.4 |
| MMD [46] | 94.1±2.3 | 86.1±2.3 | 86.8±2.2 | 92.4±2.8 | 76.4±1.5 | 78.3±1.3 |
| ATI | 95.4±1.3 | 89.0±1.4 | 89.7±1.3 | 95.9±1.3 | 84.0±1.7 | 85.1±1.5 |
| ATI-$\lambda$ | 97.1±1.1 | **89.5±1.4** | 90.2±1.3 | 96.1±2.0 | 84.1±1.8 | 85.2±1.5 |
| ATI-$\lambda$-N1 | 97.6±1.0 | **89.5±1.3** | **90.3±1.2** | **96.4±1.7** | **84.4±3.6** | **85.5±1.5** |
| ATI-$\lambda$-N2 | **97.9±1.4** | 89.4±1.2 | 90.1±1.0 | 92.8±1.6 | 84.3±2.4 | 85.4±1.5 |
| | D→A | | | D→W | | |
| | CS (10) | OS* (10) | OS (10) | CS (10) | OS* (10) | OS (10) |
| LSVM (s) | 85.2±1.7 | 40.3±4.3 | 45.2±3.8 | 97.2±0.7 | 81.4±2.4 | 83.0±2.2 |
| LSVM (t) | 88.7±2.2 | 52.8±6.0 | 57.0±5.5 | 91.5±4.9 | 59.6±3.7 | 63.2±3.4 |
| LSVM (st) | 91.9±0.7 | 68.7±2.5 | 71.2±2.3 | 98.7±0.9 | 87.3±2.3 | 88.5±2.1 |
| MMD [46] | 90.2±1.8 | 69.0±3.4 | 71.3±3.0 | 98.5±1.0 | 85.5±1.6 | 86.7±1.4 |
| ATI | **93.5±0.2** | 74.4±2.7 | 76.1±2.5 | 98.7±0.7 | 91.6±1.7 | 92.4±1.5 |
| ATI-$\lambda$ | **93.5±0.2** | 74.4±2.5 | 76.2±2.3 | 98.7±0.8 | 91.6±1.7 | 92.4±1.5 |
| ATI-$\lambda$-N1 | 93.4±0.2 | 74.6±2.5 | 76.4±2.3 | 98.9±0.5 | 92.0±1.6 | 92.7±1.5 |
| ATI-$\lambda$-N2 | **93.5±0.1** | **74.9±2.3** | **76.7±2.1** | **99.3±0.5** | **92.2±1.9** | **92.9±1.7** |

| | W→A | | | W→D | | | AVG. | | |
|---|---|---|---|---|---|---|---|---|---|
| | CS (10) | OS* (10) | OS (10) | CS (10) | OS* (10) | OS (10) | CS | OS* | OS |
| LSVM (s) | 78.8±2.9 | 32.4±3.8 | 38.2±3.4 | 99.5±0.3 | 88.7±2.2 | 89.6±1.9 | 87.1 | 58.4 | 62.0 |
| LSVM (t) | 88.7±2.2 | 52.8±6.0 | 57.0±5.5 | 92.3±3.9 | 68.2±5.2 | 71.1±4.7 | 90.9 | 60.2 | 63.8 |
| LSVM (st) | 90.8±1.3 | 66.2±4.4 | 69.0±4.1 | 99.4±0.7 | 93.5±2.7 | 94.0±2.5 | 94.8 | 78.4 | 80.3 |
| MMD [46] | 89.1±3.2 | 65.1±3.8 | 67.8±3.4 | 98.2±1.4 | 93.9±2.9 | 94.4±2.7 | 93.8 | 79.3 | 80.9 |
| ATI | **93.0±0.5** | 71.3±4.6 | 74.3±4.3 | 99.3±0.6 | 96.3±1.8 | 96.6±1.7 | 96.0 | 84.4 | 85.7 |
| ATI-$\lambda$ | **93.0±0.5** | 71.5±4.8 | 73.6±4.4 | **99.5±0.6** | 96.3±1.8 | 96.6±1.7 | 96.3 | 84.6 | 85.7 |
| ATI-$\lambda$-N1 | **93.0±0.6** | 72.2±4.5 | 74.2±4.1 | 99.3±0.6 | **96.7±2.1** | **97.0±1.9** | 96.4 | **84.9** | **86.0** |
| ATI-$\lambda$-N2 | **93.0±0.6** | **72.8±4.2** | **74.8±3.9** | 99.3±0.6 | 95.5±2.2 | 95.9±2.0 | **96.6** | 84.8 | **86.0** |

Table 6. Open set domain adaptation on the semi-supervised Office dataset with 10 shared classes (OS). We report the average and the standard deviation using a subset of samples per class in 5 random splits [34].

deep learning method MMD [46]. As baselines, we report the accuracy for the linear SVMs without domain adaptation (LSVM) when they are trained only on the source samples (s), only on the annotated target samples (t) or on both (st). As expected, the baseline trained on both performs best as shown in Table 6. Our approach ATI outperforms the baseline and the CNN approach [46]. As in the unsupervised case, the improvement compared to the CNN approach is larger for the open set (+4.8%) than for the closed set (+2.2%). While the locality constrained formulation, ATI-$\lambda$-$N$, decreased the accuracy for the unsupervised setting, it improves the accuracy for the semi-supervised case since the formulation enforces that neighbours of the target samples are assigned to the same class. The results with one (ATI-$\lambda$-$N1$) or two neighbours (ATI-$\lambda$-$N2$) are similar.

### 4.3. Dense Cross-Dataset Analysis

In order to measure the performance of our method and the open set protocol across popular datasets with more intra-class variation, we also conduct experiments on the *dense* set-up of the *Testbed for Cross-Dataset Analysis* [44]. This protocol provides 40 classes from 4 well-known datasets, *Bing (B)*, *Caltech256 (C)*, *ImageNet (I)* and *Sun (S)*. While the samples from the first 3 datasets are mostly centred and without occlusions, *Sun* becomes more challenging due to its collection of object class instances from cluttered scenes. As for the Office dataset, we take the first 10 classes as shared classes, the classes 11-25 are used as unknowns in the source domain and 26-40 as unknowns in the target domain. We use the provided DeCAF features (DeCAF7). Following the unsupervised protocol described in [43], we take 50 source samples per class for training and we test on 30 target images per class for all datasets, except *Sun*, where we take 20 samples per class.

The results reported in Table 7 are consistent with the Office dataset. ATI outperforms the baseline and the other methods by +4.4% for the closed set and by +5.3% for the open set. ATI-$\lambda$ obtains the best accuracies for the open set.

### 4.4. Sparse Cross-Dataset Analysis

We also introduce an open set evaluation using the *sparse* set-up from [44] with the datasets *Caltech101 (C)*, *Pas-*

| | B→C | | B→I | | B→S | | C→B | | C→I | | C→S | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) |
| LSVM | 82.4±2.4 | 66.6±4.0 | 75.1±0.4 | 59.0±2.7 | 43.0±2.0 | 24.2±3.0 | 53.5±2.1 | 40.1±1.9 | 76.9±4.3 | 62.5±1.2 | 46.3±2.7 | 28.2±1.4 |
| TCA [33] | 74.9±3.0 | 62.8±3.8 | 68.4±4.0 | 56.6±4.5 | 38.3±1.7 | 29.6±4.2 | 49.2±1.1 | 38.9±1.9 | 73.1±3.6 | 60.2±1.4 | 45.9±3.6 | 29.7±1.6 |
| gfk [18] | 82.0±2.2 | 66.2±4.0 | 74.3±1.0 | 58.3±3.1 | 42.2±1.4 | 23.8±2.0 | 53.2±2.6 | 40.2±1.8 | 77.1±3.3 | 62.2±1.5 | 46.2±3.0 | 28.5±1.0 |
| SA [12] | 81.1±1.8 | 66.0±3.4 | 73.9±0.9 | 57.8±3.2 | 41.9±2.4 | 24.3±2.6 | 53.4±2.5 | 40.3±1.7 | 77.3±4.2 | 62.5±.8 | 46.1±3.3 | 29.0±1.5 |
| CORAL [39] | 80.1±3.5 | 68.8±3.3 | 73.7±2.0 | 60.9±2.6 | 42.2±2.4 | 27.2±3.9 | 53.6±2.9 | 40.7±1.5 | 78.2±5.1 | 64.0±2.6 | 48.2±3.9 | 31.4±0.8 |
| ATI | 86.3±1.6 | **71.4±1.8** | 80.1±0.7 | 68.0±1.9 | **49.2±3.2** | 36.8±1.2 | 53.2±3.4 | 45.4±3.4 | 81.7±3.7 | 66.7±4.2 | 52.0±3.4 | 35.8±1.8 |
| ATI-$\lambda$ | **86.7±1.3** | **71.4±2.3** | **80.6±2.4** | **69.0±2.8** | 48.6±2.5 | **37.4±2.6** | **54.2±1.9** | **45.7±3.0** | **82.2±3.7** | **67.9±4.2** | **53.1±2.8** | **37.5±2.7** |

| | I→B | | I→C | | I→S | | S→B | | S→C | | S→I | | AVG. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) | CS (10) | OS (10) |
| LSVM | **59.1±2.0** | 42.7±2.0 | 86.2±2.6 | 73.3±3.9 | 50.1±4.0 | 32.1±3.2 | 33.1±1.7 | 16.4±1.1 | 53.1±2.6 | 27.9±2.9 | 52.3±1.8 | 25.2±0.5 | 59.3 | 41.5 |
| TCA [33] | 56.1±3.8 | 40.9±2.9 | 83.4±3.2 | 68.6±1.8 | 49.3±2.6 | 34.5±3.8 | 30.6±1.3 | 19.4±2.1 | 47.5±3.5 | 32.0±3.9 | 45.2±1.9 | 31.1±4.6 | 55.2 | 42.0 |
| gfk [18] | 58.7±1.9 | 42.6±2.4 | 86.1±2.7 | 73.3±3.6 | 49.5±3.6 | 32.7±3.6 | 33.3±1.4 | 16.9±1.5 | 53.1±3.0 | 28.6±3.8 | 52.5±2.0 | 26.4±1.1 | 59.0 | 41.6 |
| SA [12] | 58.7±1.8 | 43.1±1.6 | 85.9±2.9 | 72.8±3.1 | 50.0±3.6 | 32.2±3.7 | 34.2±1.1 | 17.5±1.6 | 52.5±3.2 | 29.2±4.2 | 52.6±2.4 | 27.1±1.3 | 59.0 | 41.1 |
| CORAL [39] | 58.5±2.7 | 44.6±2.5 | 85.8±1.5 | 74.5±3.4 | 49.5±4.8 | 35.4±4.4 | 32.9±1.6 | 18.7±1.2 | 52.1±2.8 | 33.6±5.3 | 52.9±1.8 | 31.3±1.3 | 59.0 | 44.2 |
| ATI | 57.9±1.9 | **48.8±2.3** | 89.3±2.2 | 77.1±2.6 | 55.0±5.0 | 42.2±4.0 | **34.9±2.6** | 22.8±3.1 | 59.8±1.3 | 46.9±2.5 | **60.8±3.4** | 32.9±2.2 | 63.4 | 49.5 |
| ATI-$\lambda$ | 58.6±1.4 | 48.7±1.8 | **89.7±2.3** | **77.5±2.2** | **55.3±4.3** | **43.4±4.8** | 34.1±2.4 | **23.2±3.2** | **60.2±2.7** | **47.3±2.9** | 60.3±2.4 | **33.0±1.1** | **63.6** | **50.2** |

Table 7. Unsupervised open set domain adaptation on the Testbed dataset (dense setting) with 10 shared classes (OS). For comparison, results for closed set domain adaptation (CS) are reported.

| | C→O | C→P | O→C | O→P | P→C | P→O | AVG. |
|---|---|---|---|---|---|---|---|
| *shared classes* | 8 | 7 | 8 | 4 | 7 | 4 | |
| *unknown / all (t)* | 0.52 | 0.30 | 0.90 | 0.81 | 0.54 | 0.78 | |
| LSVM | 46.3 | 36.1 | 60.8 | 29.7 | 78.8 | 70.1 | 53.6 |
| TCA [33] | 45.2 | 33.8 | 58.1 | 31.1 | 63.4 | 61.1 | 48.8 |
| gfk [18] | 46.4 | 36.2 | 61.0 | 29.7 | 79.1 | **72.6** | 54.2 |
| SA [12] | 46.4 | 36.8 | 61.1 | 30.2 | 79.8 | 71.1 | 54.2 |
| CORAL [39] | 48.0 | 35.9 | 60.2 | 29.1 | 78.9 | 68.8 | 53.5 |
| ATI | **51.6** | **52.1** | 63.1 | 38.8 | 80.6 | 70.9 | 59.5 |
| ATI-$\lambda$ | 51.5 | 52.0 | **63.4** | **39.1** | **81.1** | 71.1 | **59.7** |

Table 8. Unsupervised open set domain adaptation on the sparse set-up from [44].

*cal07 (P)* and *Office (O)*. These datasets are quite unbalanced and offer distinctive characteristics: *Office* contains centred class instances with barely any background (17 classes, 2300 samples in total, 68-283 samples per class), *Caltech101* allows for more class variety (35 classes, 5545 samples in total, 35-870 samples per class) and *Pascal07* gathers more realistic scenes with partially occluded objects in various image locations (16 classes, 12219 samples in total, 193-4015 samples per class). For each domain shift, we take all samples of the shared classes and consider all other samples as unknowns. Table 8 summarises the amount of shared classes for each shift and the percentage of unknown target samples, which varies from 30% to 90%.

**Unsupervised domain adaptation** For the unsupervised experiment, we conduct a single run for each domain shift using all source and unlabelled target samples. The results are reported in Table 8. ATI outperforms the baseline and the other methods by +5.3% for this highly unbalanced open set protocol. ATI-$\lambda$ improves the accuracy of ATI slightly.

**Semi-supervised domain adaptation** In order to evaluate the semi-supervised setting, we take all source samples and 3 annotated target samples per shared class as it is done in the semi-supervised setting for the Office dataset [34]. The average and standard deviation over 5 random splits are reported in Table 9. While ATI improves over the baseline trained on the source and target samples together (st) by +4.5%, ATI-$\lambda$ and the locality constraints with one neighbour boost the performance further. ATI-$\lambda$-$N_1$ improves the accuracy of the baseline by +5.1%.

| | C→O | C→P | O→C | O→P | P→C | P→O | AVG. |
|---|---|---|---|---|---|---|---|
| LSVM (s) | 46.5±0.1 | 36.2±0.1 | 60.8±0.3 | 29.7±0.0 | 79.5±0.3 | 73.5±0.7 | 54.4 |
| LSVM (t) | 53.1±3.7 | 44.6±2.1 | 73.7±1.5 | 40.5±3.0 | 81.1±2.5 | 70.5±4.3 | 60.6 |
| LSVM (st) | 56.0±1.3 | 44.5±1.2 | 68.9±1.1 | 40.9±2.2 | 80.9±0.6 | 76.7±0.3 | 61.3 |
| ATI | 59.6±1.2 | 55.2±1.3 | 75.8±1.2 | 45.2±1.4 | 81.6±0.2 | **77.1±0.8** | 65.8 |
| ATI-$\lambda$ | 60.3±1.2 | 56.0±1.2 | 75.8±1.1 | **45.8±1.2** | 81.8±0.2 | 76.9±1.3 | 66.1 |
| ATI-$\lambda$-N1 | **60.7±1.2** | **56.3±1.2** | **76.7±1.6** | **45.8±1.4** | **82.0±0.4** | 76.7±1.1 | **66.4** |

Table 9. Semi-supervised open set domain adaptation on the sparse set-up from [44] with 3 labelled target samples per shared class.

## 5. Conclusions

In this paper we have introduced the concept of open set domain adaptation. In contrast to closed set domain adaptation, the source and target domain share only a subset of object classes whereas most samples of the target domain belong to classes not present in the source domain. We proposed new open set protocols for existing datasets and evaluated both CNN methods as well as standard unsupervised domain adaptation approaches. In addition, we have proposed an approach for unsupervised open set domain adaptation. The approach can also be applied to closed set domain adaptation and semi-supervised domain adaptation. In all settings, our approach achieves state-of-the-art results.

**Acknowledgment:** The work has been supported by the ERC Starting Grant ARCA (677650) and the DFG project GA 1927/2-2 as part of the DFG Research Unit FOR 1505 Mapping on Demand (MoD).

## References


[1] T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] Y. Aytar and A. Zisserman. Tabula rasa: model transfer for object category detection. In *IEEE International Conference on Computer Vision*, pages 2252–2259, 2011.

[3] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *IEEE International Conference on Computer Vision*, pages 769–776, 2013.

[4] P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(6):1823–1840, 2008.

[5] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conference on empirical methods in natural language processing*, pages 120–128, 2006.

[6] L. Bruzzone and M. Marconcini. Domain adaptation problems: a DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):770–787, 2010.

[7] P. Busto, J. Liebelt, and J. Gall. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. In *British Machine Vision Conference*, pages 14.1–14.12, 2015.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, 2011.

[9] S. Chopra, S. Balakrishnan, and R. Gopalan. DLID: Deep learning for domain adaptation by interpolating between domains. In *ICML workshop on challenges in representation learning*, 2013.

[10] G. Csurka, B. Chidlowskii, S. Clinchant, and S. Michel. Unsupervised domain adaptation with regularized domain instance denoising. In *IEEE European Conference on Computer Vision*, pages 458–466, 2016.

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pages 647–655, 2014.

[12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision*, pages 2960–2967, 2013.

[13] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[14] M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. In *The Pacific Rim International Conferences on Artificial Intelligence*, pages 898–904, 2014.

[15] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *IEEE European Conference on Computer Vision*, pages 597–613, 2016.

[16] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 222–230, 2013.

[17] B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1286–1294, 2013.

[18] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.

[19] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 999–1006, 2011.

[20] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, pages 513–520, 2006.

[21] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko. Asymmetric and category invariant feature transformations for domain adaptation. *International Journal of Computer Vision*, 109(1-2):28–41, 2014.

[22] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell. Efficient learning of domain-invariant image representations. In *International Conference on Learning Representations*, 2013.

[23] L. P. Jain, W. J. Scheirer, and T. E. Boult. Multi-class open set recognition using probability of inclusion. In *IEEE European Conference on Computer Vision*, 2014.

[24] S. G. Johnson. The NLopt nonlinear-optimization package, 2007–2010.

[25] L. Kaufman and F. Broeckx. An algorithm for the quadratic assignment problem using bender's decomposition. *European Journal of Operational Research*, 2(3):207–211, 1978.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[27] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1785–1792, 2011.

[28] F. Li and H. Wechsler. Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1686–1697, 2005.

[29] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015.

[30] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[31] T. Ming Harry Hsu, W. Yu Chen, C.-A. Hou, Y.-H. Hubert Tsai, Y.-R. Yeh, and Y.-C. Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4121–4129, 2015.

[32] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 692–699, 2013.

[33] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *International Jont Conference on Artifical Intelligence*, pages 1187–1192, 2009.

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *IEEE European Conference on Computer Vision*, pages 213–226, 2010.

[35] E. Sangineto. Statistical and spatial consensus collection for detector adaptation. In *IEEE European Conference on Computer Vision*, pages 456–471, 2014.

[36] W. J. Scheirer, L. P. Jain, and T. E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, 2014.

[37] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[38] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 2013.

[39] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2015.

[40] B. Sun and K. Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *British Machine Vision Conference*, 2014.

[41] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573, 2002.

[42] T. Tommasi and B. Caputo. Frustratingly easy NBNN domain adaptation. In *IEEE International Conference on Computer Vision*, pages 897–904, 2013.

[43] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. A deeper look at dataset bias. *CoRR*, 2015.

[44] T. Tommasi and T. Tuytelaars. A testbed for cross-dataset analysis. In *IEEE European Conference on Computer Vision: Workshop on Transferring and Adapting Source Knowledge in Computer Vision*, pages 18–31, 2014.

[45] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[46] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[47] H. Xu, J. Zheng, and R. Chellappa. bridging the domain shift by domain adaptive dictionary learning. In *British Machine Vision Conference*, pages 96.1–96.12, 2015.

[48] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2142–2150, 2015.

[49] R. Zhang and D. N. Metaxas. Ro-svm: Support vector machine with reject option for image categorization. In *British Machine Vision Conference*, pages 1209–1218, 2006.