

Open Set Domain Adaptation for Image and Action Recognition

Pau Panareda Busto, Ahsan Iqbal, and Juergen Gall, *Member, IEEE*

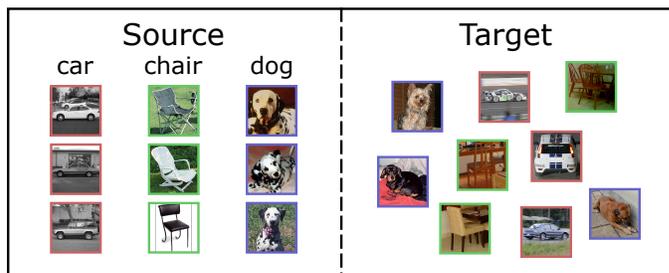
Abstract—Since annotating and curating large datasets is very expensive, there is a need to transfer the knowledge from existing annotated datasets to unlabelled data. Data that is relevant for a specific application, however, usually differs from publicly available datasets since it is sampled from a different domain. While domain adaptation methods compensate for such a domain shift, they assume that all categories in the target domain are known and match the categories in the source domain. Since this assumption is violated under real-world conditions, we propose an approach for open set domain adaptation where the target domain contains instances of categories that are not present in the source domain. The proposed approach achieves state-of-the-art results on various datasets for image classification and action recognition. Since the approach can be used for open set and closed set domain adaptation, as well as unsupervised and semi-supervised domain adaptation, it is a versatile tool for many applications.

Index Terms—Domain Adaptation, Open Set Recognition, Action Recognition.

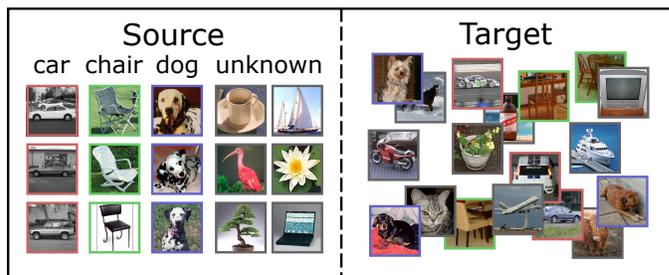
1 INTRODUCTION

IN the last years, impressive results have been achieved on large-scale datasets for image classification or action recognition. Acquiring such large annotated datasets, however, is very expensive and there is a need to transfer the knowledge from existing annotated datasets to unlabelled data that is relevant for a specific application. If the labelled and unlabelled data have different characteristics, they have been sampled from two different domains. In particular, datasets that have been collected from the Internet, e.g., from platforms for sharing videos or images, differ greatly from data that needs to be processed for an application. To address the domain shift between the labelled dataset, which is the source domain, and the unlabelled data from the target domain, various unsupervised domain adaptation approaches have been proposed. If the data from the target source is partially labelled, the problem is termed semi-supervised domain adaptation. In this work, we address unsupervised and semi-supervised domain adaptation in the context of image and action recognition.

Although the methods for domain adaptation have been advanced tremendously in the last years [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], the evaluation protocols were restricted to a scenario where all categories in the target domain are known and match the categories in the source domain. Fig. 1(a) illustrates such a *closed set domain adaptation* setting. The assumption that all images or videos that are in the target domain belong to categories in the source domain, however, is violated in most cases. In particular if the number of potential categories is very large as it is the case for object or action categories, the target domain contains images or videos of categories that are not present in the source domain since they are not of interest for a specific application. We therefore propose a more realistic evalu-



(a) Closed set domain adaptation



(b) Open set domain adaptation

Fig. 1. (a) Standard domain adaptation benchmarks assume that source and target domains contain images or videos only of the same set of categories. This is denoted as *closed set domain adaptation* since it does not include samples of unknown categories or categories 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 or videos that do not belong to the categories of interest. Furthermore, the target domain contains images or videos that are not related to any image or video in the source domain and vice versa.

ation setting for unsupervised or semi-supervised domain adaptation, namely *open set domain adaptation*, which builds on the concept of open sets [11], [12], [13]. As illustrated in Fig. 1, the source and target domains are not anymore restricted in the open set case to share the same categories as in the closed set case, but both domains contain images or videos from categories that are not present in the other

• P. Panareda Busto, M. Iqbal and J. Gall are with the Computer Vision Group, University of Bonn, Germany.
E-mails: s6papana@uni-bonn.de, {iqbal,m,gall}@iai.uni-bonn.de

Manuscript received January 29, 2018.

domain.

To address the problem of open set domain adaptation, we propose a generic approach that learns a linear mapping that maps the feature space of the source domain to the feature space of the target domain. It assigns a subset of images or videos of the target domain to the categories of the source domain and transforms the feature space of the source domain gradually towards the feature space of the target domain. By using a subset instead of the entire set, the approach handles images or videos in the target domain that are not related to any sample in the source domain. The approach can be applied to any feature space, which includes features extracted from images as well as features extracted from videos. The approach works in particular very well for features spaces that are extracted by convolutional networks and outperforms most end-to-end learning approaches for domain adaptation. The good performance of the approach coincides with the observation that deep convolutional networks tend to linearise manifolds of image domains [14], [15]. In this case, a linear mapping is sufficient to map the feature space of the source domain to the feature space of the target domain. In particular, the flexibility of the approach, which can be used for images and videos, for open set and closed set domain adaptation, as well as unsupervised and semi-supervised domain adaptation, makes the approach a versatile tool for applications. An overview of the approach for unsupervised open set domain adaptation is given in Fig. 2.

A preliminary version of this work was presented in [16]. In this work, we introduce open set domain adaptation for action recognition and provide a thorough experimental evaluation, which includes open set domain adaptation from synthetic data to real data and an evaluation of the proposed approach for standard closed set protocols. In total, we evaluate the approach on 26 *open set* and 34 *closed set* combinations of source and target domains including the *Office* dataset [1], its extension with the *Caltech* dataset [3], the *Cross-Dataset Analysis* [17], the *Sentiment dataset* [18], synthetic data [19], and two action recognition datasets, namely the *Kinetics Human Action Video Dataset* [20] and the *UCF101 Action Recognition Dataset* [21]. Our approach achieves state-of-the-art results in all settings both for unsupervised and semi-supervised open set domain adaptation and obtains competitive results compared state-of-the-art deep learning approaches for closed set domain adaptation.

2 RELATED WORK

2.1 Domain Adaptation

The interest in studying domain adaptation techniques for computer vision problems increased with the release of a benchmark by Saenko et al. [1] 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. [2] and Gong et al. [3], 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 [22] and with dictionary learning to find subspace interpolations [23], [24], [25]. Sun et al. [26], [27] presented a very efficient solution based on second-order statistics to align a source domain with a target domain. Herath et al. [28] also match second-order statistics with a joint estimation of latent spaces. To obtain an estimate of the target distribution in the latent space, Gholami et al. [29] introduce a Bayesian approximation to jointly learn a softmax classifier across-domains. Similarly, Csurka et al. [30] jointly denoise source and target samples to reconstruct data without partial random corruption. Zhang et al. [31] also align distributions, but they include geometrical differences in a joint optimisation. Sharing certain similarities with associations between domains, Gong et al. [32] minimise the Maximum Mean Discrepancy (MMD) [33] of two datasets. They assign instances to latent domains and solve it by a relaxed binary optimisation. Hsu et al. [7] 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. [34] proposed a transfer learning formulation to regularise the training of target classifiers. Exploiting pairwise constraints across domains, Saenko et al. [1] and Kulis et al. [35] learn a transformation to minimise the effect of the domain shift while also training target classifiers. Following the same idea, Hoffman et al. [36] considered an iterative process to alternatively minimise the classification weights and the transformation matrix. In a different context, [37] 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 [38]. Pivot features that behave the same way for discriminative learning in both domains were selected to model their correlations. Gong et al. [39] 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. [40]. 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 [41], [42].

During the last years, a large number of domain adaptation methods have been based on deep convolutional neural networks (CNN) [43], which learn more discriminative feature representations than hand-crafted features and substantially reduce the domain bias between datasets in object recognition tasks [44]. Non-adapted classifiers trained with features extracted from CNN layers outperform domain adaptation methods with shallow feature descriptors [27], [44]. Many of these deep domain adaptation architectures are inspired by the traditional methods and seek to minimise the MMD distance as a regulariser to learn features for source and target samples jointly [45], [46], [47],

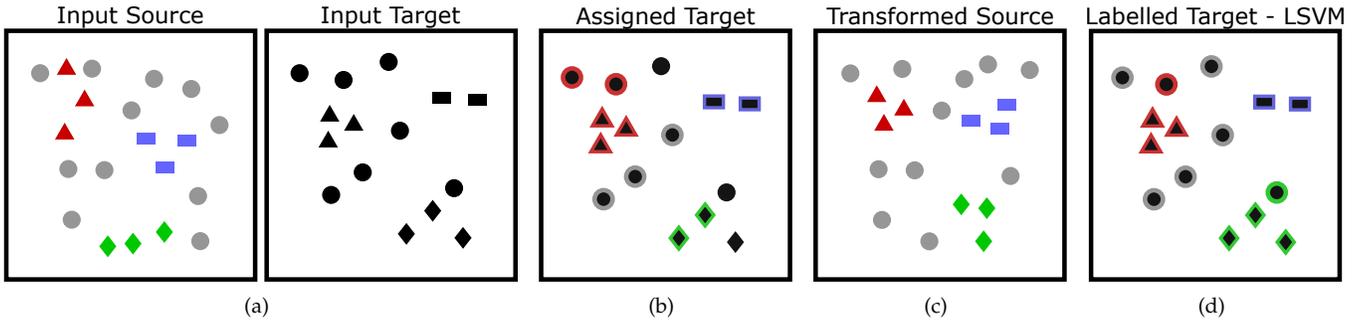


Fig. 2. Overview of the proposed approach for unsupervised *open set domain adaptation*. (a) The source domain contains some labelled images, indicated by the colours red, blue and green, and some images belonging to unknown classes (grey). For the target domain, we do not have any labels but the shapes 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. 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.

[48], [49]. Recently, Carlucci et al. [50] extend this type of networks and use intermediate layers for the alignment of distributions before batch normalisation. They learn a parameter that steers the contribution of each domain at a given layer. Ganin et al. [6] added a domain classifier network after the CNN to maximize the discriminatory loss of both domains while jointly minimising the classification loss using source data. More recently, Tzeng et al. [9] propose a generalized framework for adversarial adaptation. In the semi-supervised setting, Mottian et al. [10] present a deep domain adaptation method that exploits the domain loss minimisation while maximizing the distances between labelled samples from different domains and classes. Other forms of data representation, such as hash codes [51] and scatter tensors [52], [53], have also been combined with deep domain adaptation architectures to further reduce the domain bias.

2.2 Open Set Recognition

The inclusion of *open sets* in recognition tasks appeared in the field of face recognition, where evaluation datasets contain unseen face instances as impostors that have to be rejected [54], [55]. Such open set protocols are nowadays widely used for evaluating face recognition approaches [56].

The generalisation towards an open set scenario for multi-object classification was introduced by Schreier et al. [11], who addressed the more realistic case of a finite set of known objects mixed with many unknown ones. Based on this principle, [57] and [12] propose multi-class classifiers that detect unknown instances by learning SVMs that assign probabilistic decision scores instead of class labels. More recently, Bendale and Boulton [13] adapt traditional neural networks for open set recognition tasks by introducing a new layer that estimates the probability of an object to be labelled as unseen class.

Closely related are also the works [58] and [59] that add a regulariser to detect uninformative data and penalise a misclassification during training. Lately, Gavves et al. [60] present an active learning technique, whose initially trained SVMs on a subset of known classes are used as priors to further train novel object classes from other target datasets.

3 OPEN SET DOMAIN ADAPTATION

We present 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 λ , which is discussed in Section 4.1.

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

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

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 [61] 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.,

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

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 [62]. By substituting

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

we derive to the linearised problem

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

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 can be written in matrix form:

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

The matrices P_S and $P_T \in \mathbb{R}^{D \times L}$ 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 non-linear optimisation [63], [64] 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 image classification and action recognition. In this setting, the images or videos of the source domain are annotated by class labels and the goal is to classify the images or videos 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* [65] since it has also been used in other works, e.g., [66] and [27]. 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 [27], [66]. The source code and the described open set protocols are available at <https://github.com/Heliot7/open-set-da>.

4.1 Parameter configuration

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

$$\lambda = \rho \left(\max_{t,c} d_{ct} + \min_{t,c} d_{ct} \right), \quad (9)$$

i.e., λ is adapted automatically based on the distances d_{ct} and ρ , which is set to 0.5 unless otherwise specified. While higher values of λ closer to the largest distance barely discard any outlier, lower values almost reject all assignments. We iterate the approach until the maximum number of 10 iterations is reached or if the distance

$$\sqrt{\sum_t \sum_c x_{ct} \|W_k S_c - T_t\|_2^2} \quad (10)$$

is below $\epsilon = 0.01$, where W_k denotes the estimated transformation at iteration k . In practice, the process converges after 3-5 iterations.

4.2 Open set domain adaptation

4.2.1 Office dataset

We evaluate and compare our approach on the *Office* dataset [1], 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 ($D \rightarrow W$, $W \rightarrow D$). Following the standard protocol and for a fair comparison with the other methods, we extract feature vectors from the fully connected layer-7 (fc7) of the AlexNet model [43].

We introduce an open set protocol for this dataset by taking the 10 classes that are also common in the *Caltech* dataset [3] 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

	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 [47]	88.1	76.5	77.6	90.5	70.2	72.5			
RTN [48]	93.0	74.7	76.6	87.0	70.8	73.0			
BP [6]	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-λ	93.0	79.2	79.8	84.0	76.5	77.6			
ATI-λ-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 [47]	83.4	53.5	57.0	96.1	87.5	88.4			
RTN [48]	82.8	53.8	57.2	97.9	88.1	89.0			
BP [6]	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-λ	93.8	70.0	71.3	98.5	93.2	93.5			
ATI-λ-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 [47]	84.9	58.5	60.8	100	97.5	98.3	90.5	74.0	75.8
RTN [48]	85.1	60.2	62.4	100	98.3	98.8	91.0	74.3	76.2
BP [6]	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-λ	93.7	76.5	76.7	100	99.2	98.3	93.7	82.4	82.9
ATI-λ-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 [32]. For comparison, results for closed set domain adaptation (CS) and modified open set (OS*) are reported.

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).

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 [47] 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 [48] extends DAN by adding a residual transfer module that bridges the source and target classifiers. BP [6] 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 [32], which is the standard protocol for CNNs on this dataset. As proposed in [6], we use for all methods linear SVMs for classification instead of the soft-max layer for a fair comparison.

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-λ includes the outlier rejection and ATI-λ-N₁ 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.

	A→D		A→W		D→A		D→W		W→A		W→D	
	assign- λ	LSVM										
initial		72.6		57.5		45.1		88.5		49.2		96.6
iteration 1	78.4	76.8	74.5	69.8	73.6	68.1	90.4	90.3	71.9	70.0	89.6	97.8
iteration 2	77.7	79.1	80.1	77.6	80.4	71.3	91.5	93.5	77.2	75.9	84.7	98.3
iteration 3	75.3	79.8							77.8	76.7		

TABLE 2

Evolution of the percentage of correct assignments (assign- λ) when taking into account the selected target samples and the average class accuracy of all target samples using linear SVMs (LSVM). The approach converges after 2 or 3 iterations.

	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 [67]	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 [3]	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 [66]	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 [27]	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- λ	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 [67]	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 [3]	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 [66]	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 [27]	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- λ	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- λ	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 3

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 [1]. For comparison, results for closed set domain adaptation (CS) and modified open set (OS*) are reported.

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., the combinations that include the *Amazon* dataset. We also observe that ATI 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- λ 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- λ - N_1 , which we propose only for the semi-supervised setting, decreases the accuracy in the unsupervised setting.

The evolution of the percentage of correct assignments and the intermediate classification accuracies are shown in Table 2. The approach converges after two or three iterations. While the accuracy of the LSVMs that are trained on the transformed source samples increases with each

iteration, the accuracy of the assignment can even decrease in some cases.

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 dimensional subspace, including Transfer Component Analysis (TCA) [67], Geodesic Flow Kernel (GFK) [3] and Subspace alignment (SA) [66]. In addition, we also include CORAL [27], which whitens and recolors the source towards the target data. Following the standard protocol of [1], we take 20 samples per object class when *Amazon* is used as source domain, and 8 for *DSLRL* or *Webcam*. As in the previous comparison with the CNN-based methods, we extract feature vectors from the last convolutional layer (fc7) from the AlexNet model [43]. 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 3. 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- λ) 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) [12], 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 4 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- λ 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 5 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 *DSLRL+Webcam*.

	A→D		A→W		D→A		D→W		W→A		W→D		AVG.	
	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-λ	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 4

Comparison of a standard linear SVM (LSVM) with a specific open set SVM (OS-SVM) [11] on the unsupervised Office dataset with 10 shared classes using all samples per class [32].

	A→D	A→W	D→A	D→W	W→A	W→D	AVG.
	OS(10)						
ATI-λ (C w/o unknown)	79.0	77.1	70.5	93.4	75.8	98.2	82.3
ATI-λ (C with unknown)	79.8	77.6	71.3	93.5	76.7	98.3	82.9

TABLE 5

Impact of including the unknown class to the set of classes C . The evaluation is performed on the unsupervised Office dataset with 10 shared classes using all samples per class [32].

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 6, 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 adaptation method CORAL [27]. 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%.

<i>Amazon</i> → <i>DSLR+Webcam</i>								
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 [27]	77.2	76.4	76.2	74.8	73.7	71.5	70.8	69.7
ATI-λ	80.3	82.4	81.2	81.7	82.5	80.9	80.7	81.9

TABLE 6

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.

Subsampling of target samples. In order to evaluate the robustness of our method when having a reduced amount of target samples for domain adaptation, we subsample the target data. Fig. 3 shows the results for ATI-λ on the 6 domain shifts of the Office dataset with the standard open set protocol (OS). We vary the number of target samples from 50 to the total number of instances. For a fixed number of target samples, we randomly sample 5 times from the target data and plot the lowest, highest and average accuracy of the 5 runs. The accuracy is always measured on the whole target dataset. The results show that between 300 and 400 target instances are sufficient to achieve similar accuracies than our method with all target samples. When the domain shifts are smaller, e.g., $D \rightarrow W$ and $W \rightarrow D$, even less target samples are required.

Scalability analysis of target samples. The number of sampled target samples has an impact on the execution time of the assignment and the transformation steps of the iterative

process. Therefore, we also test the scalability of the two steps of our method with respect to the number of target samples. The average execution times of both techniques in the domain shift *Amazon* → *DSLR+Webcam* for all the random splits and unknown sets of the previous evaluation are shown in Fig. 4. We observe that the assignment problem takes less than a second to be solved for any size of target data from the evaluated settings. Most of the computation time is required for estimating the transformation W , which requires at least 120 seconds. The computation time of this step, however, increases only moderately with respect to the number of target samples.

Impact of parameter ρ . The cost that determines whether a target sample is considered as outlier during the assignment process is defined by λ (9), which is based on the current minimum and maximum distance between the source clusters and target samples. Thus, λ is updated at each iteration. The value of λ , however, also depends on the parameter ρ . For all experiments, we use $\rho = 0.5$ as default value, aiming for a moderate outlier rejection. Fig. 5 shows the impact of ρ on the accuracy. Using $\rho = 0.5$, which rejects around 10-20% of the target samples, achieves the best results in 5 out of the 6 domain shifts on the Office dataset. When ρ gets closer to 0 the accuracy drops substantially since too many samples are discarded.

Impact of constraint $\sum_t x_{ct} \geq 1$. Our formulation in (1) ensures that at least one target sample is assigned to an object category. Therefore, all classes contribute to the estimation of the transformation matrix W . In order to measure its impact on the adaptation problem, we run experiments with $\sum_t x_{ct} \geq 1$ and without the constraint, i.e., when a class might not be assigned to any target sample at all. As illustrated in Fig. 5, the inclusion of this constraint provides higher accuracies when $\rho < 0.3$. For greater values of ρ , the constraint can be omitted since it does not influence the accuracy.

Impact of wrong assignments. During the iterative process of our method, wrong assignments take part in the optimisation of W , introducing false associations between the source and the target domain that negatively affect the final transformation. A general assumption in our method is that the correct assignments largely compensate the wrong ones and, thus, the transformed source data allows for better classification accuracies in the target domain. Therefore,

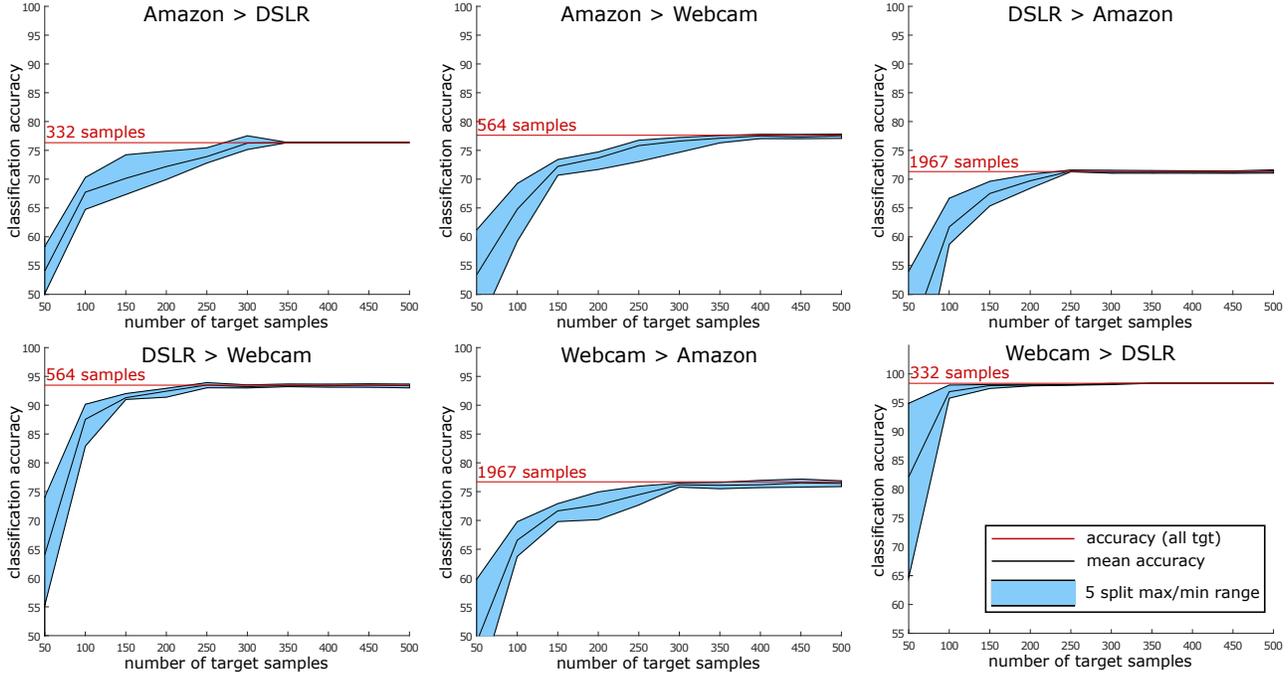


Fig. 3. Impact of using a random subset of target samples. The blue region shows the difference between the best and worst result of the 5 randomly sampled subsets for a given number of target samples and the black line within the region is the mean accuracy of the 5 subsets. The red line indicates the classification accuracy when using all target samples. The results are reported for ATI- λ using the open set protocol on the unsupervised Office dataset with 10 shared classes using all samples per class.

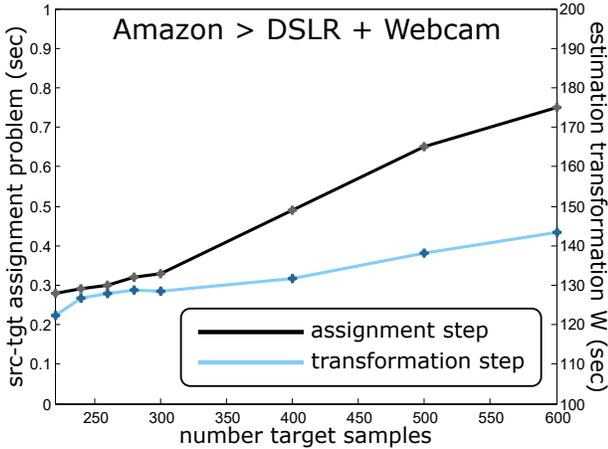


Fig. 4. Execution time in seconds for the assignment and transformation estimation steps of a single iteration with respect to the number of target samples.

we artificially generate assignments in the first iteration by assigning a random subset of target samples to the correct class in the source domain and the remaining target samples to random classes. We then run our approach without any additional modifications until it converges. We report in Table 7 the average percentage of correct assignments of 5 random splits for the domain shift $Amazon \rightarrow DSLR+Webcam$ with 400 unknown samples. While the first iteration represents the accuracy of correct and random assignments that we generate, the last row shows the accuracies after the approach has converged. As it can be observed, the approach ends in a local optimum, but the accuracies increase for all cases except if we initialise the

approach with 100% correct assignments. It is expected that the assignment accuracy does not remain at 100% since the image manifolds are not perfectly linearised and even for the best estimate of W wrong assignments can occur.

<i>Amazon</i> \rightarrow <i>DSLR+Webcam</i> (400 unknown samples)											
%gt (+rnd)	10	20	30	40	50	60	70	80	90	100	std
iteration 1	18.2	27.0	36.1	45.2	54.3	63.5	72.7	81.7	90.7	100.0	85.1
final	24.4	40.1	54.7	65.4	72.8	79.2	83.6	88.8	93.1	96.7	88.6

TABLE 7

Impact of limiting the amount of correct assignments in the first iteration. We report the average percentage of correct assignments over 5 random splits and increase the percentage of correctly selected assignments from 10% to 100%, leaving the rest randomly selected. The last column shows the percentage of correct assignments of the method without modifying the initial assignments.

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 [1] 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 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 8. 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- λ - N , decreased the accuracy for the unsupervised setting, it

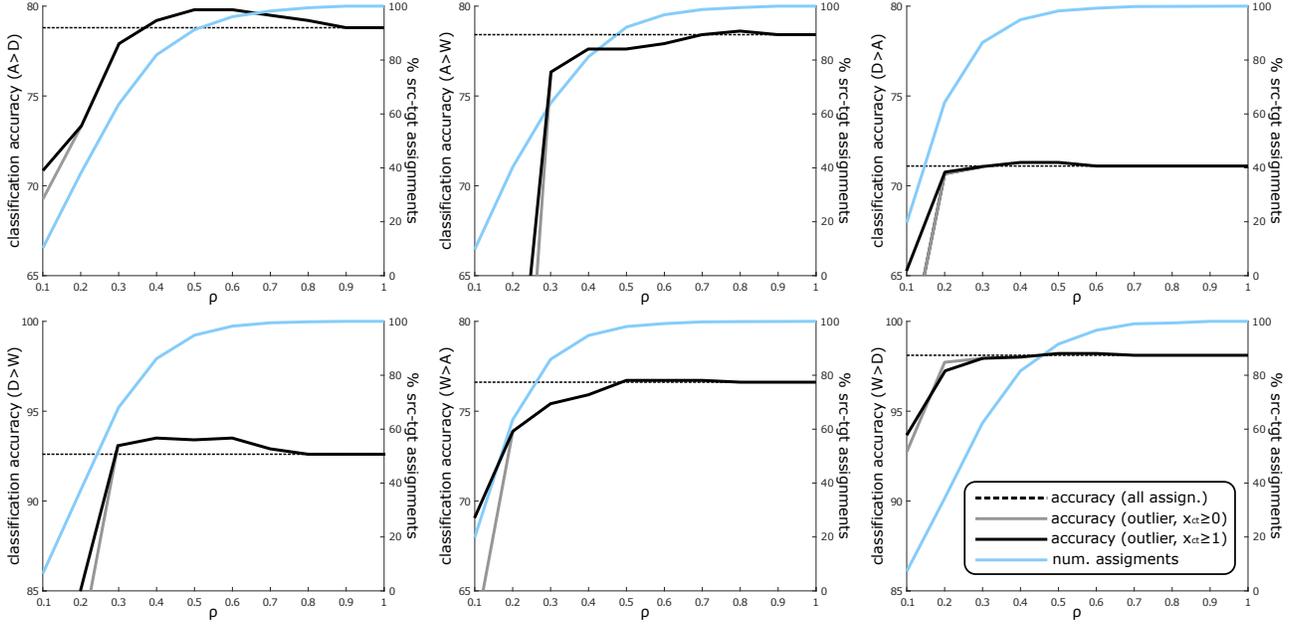


Fig. 5. The black and grey curves show the classification accuracies for varying values of ρ when including or not the constraint $\sum_i x_{ct} \geq 1$, respectively. $\rho = 0.5$ obtains the best accuracies in 5 out of 6 domain shifts. The blue curve shows the percentage of selected assignments to compute the transformation matrix W in the first iteration. The results are reported for ATI- λ using the open set protocol on the unsupervised Office dataset with 10 shared classes using all samples per class.

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- λ -N1) or two neighbours (ATI- λ -N2) are similar.

	A \rightarrow D			A \rightarrow W					
	CS (10)	OS* (10)	OS (10)	CS (10)	OS* (10)	OS (10)			
LSVM (s)	85.8 \pm 3.2	62.1 \pm 7.9	65.9 \pm 6.2	76.4 \pm 2.1	45.7 \pm 5.0	50.4 \pm 4.5			
LSVM (t)	92.3 \pm 3.9	68.2 \pm 5.2	71.1 \pm 4.7	91.5 \pm 4.9	59.6 \pm 3.7	63.2 \pm 3.4			
LSVM (st)	95.7 \pm 1.3	82.5 \pm 3.0	84.0 \pm 2.6	92.4 \pm 1.8	72.5 \pm 3.7	74.8 \pm 3.4			
MMD [46]	94.1 \pm 2.3	86.1 \pm 2.3	86.8 \pm 2.2	92.4 \pm 2.8	76.4 \pm 1.5	78.3 \pm 1.3			
ATI	95.4 \pm 1.3	89.0 \pm 1.4	89.7 \pm 1.3	95.9 \pm 1.3	84.0 \pm 1.7	85.1 \pm 1.5			
ATI- λ	97.1 \pm 1.1	89.5\pm1.4	90.2 \pm 1.3	96.1 \pm 2.0	84.1 \pm 1.8	85.2 \pm 1.5			
ATI- λ -N1	97.6 \pm 1.0	89.5\pm1.3	90.3\pm1.2	96.4\pm1.7	84.4\pm3.6	85.5\pm1.5			
ATI- λ -N2	97.9\pm1.4	89.4 \pm 1.2	90.1 \pm 1.0	92.8 \pm 1.6	84.3 \pm 2.4	85.4 \pm 1.5			
	D \rightarrow A			D \rightarrow W					
	CS (10)	OS* (10)	OS (10)	CS (10)	OS* (10)	OS (10)			
LSVM (s)	85.2 \pm 1.7	40.3 \pm 4.3	45.2 \pm 3.8	97.2 \pm 0.7	81.4 \pm 2.4	83.0 \pm 2.2			
LSVM (t)	88.7 \pm 2.2	52.8 \pm 6.0	57.0 \pm 5.5	91.5 \pm 4.9	59.6 \pm 3.7	63.2 \pm 3.4			
LSVM (st)	91.9 \pm 0.7	68.7 \pm 2.5	71.2 \pm 2.3	98.7 \pm 0.9	87.3 \pm 2.3	88.5 \pm 2.1			
MMD [46]	90.2 \pm 1.8	69.0 \pm 3.4	71.3 \pm 3.0	98.5 \pm 1.0	85.5 \pm 1.6	86.7 \pm 1.4			
ATI	93.5\pm0.2	74.4 \pm 2.7	76.1 \pm 2.5	98.7 \pm 0.7	91.6 \pm 1.7	92.4 \pm 1.5			
ATI- λ	93.5\pm0.2	74.4 \pm 2.5	76.2 \pm 2.3	98.7 \pm 0.8	91.6 \pm 1.7	92.4 \pm 1.5			
ATI- λ -N1	93.4 \pm 0.2	74.6 \pm 2.5	76.4 \pm 2.3	98.9 \pm 0.5	92.0 \pm 1.6	92.7 \pm 1.5			
ATI- λ -N2	93.5\pm0.1	74.9\pm2.3	76.7\pm2.1	99.3\pm0.5	92.2\pm1.9	92.9\pm1.7			
	W \rightarrow A			W \rightarrow D			AVG.		
	CS (10)	OS* (10)	OS (10)	CS (10)	OS* (10)	OS (10)	CS	OS*	OS
LSVM (s)	78.8 \pm 2.9	32.4 \pm 3.8	38.2 \pm 3.4	99.5 \pm 0.3	88.7 \pm 2.2	89.6 \pm 1.9	87.1	58.4	62.0
LSVM (t)	88.7 \pm 2.2	52.8 \pm 6.0	57.0 \pm 5.5	92.3 \pm 3.9	68.2 \pm 5.2	71.1 \pm 4.7	90.9	60.2	63.8
LSVM (st)	90.8 \pm 1.3	66.2 \pm 4.4	69.0 \pm 4.1	99.4 \pm 0.7	93.5 \pm 2.7	94.0 \pm 2.5	94.8	78.4	80.3
MMD [46]	89.1 \pm 3.2	65.1 \pm 3.8	67.8 \pm 3.4	98.2 \pm 1.4	93.9 \pm 2.9	94.4 \pm 2.7	93.8	79.3	80.9
ATI	93.0\pm0.5	71.3 \pm 4.6	74.3 \pm 4.3	99.3 \pm 0.6	96.3 \pm 1.8	96.6 \pm 1.7	96.0	84.4	85.7
ATI- λ	93.0\pm0.5	71.5 \pm 4.8	73.6 \pm 4.4	99.5\pm0.6	96.3 \pm 1.8	96.6 \pm 1.7	96.3	84.6	85.7
ATI- λ -N1	93.0\pm0.6	72.2 \pm 4.5	74.2 \pm 4.1	99.3 \pm 0.6	96.7\pm2.1	97.0\pm1.9	96.4	84.9	86.0
ATI- λ -N2	93.0\pm0.6	72.8\pm4.2	74.8\pm3.9	99.3 \pm 0.6	95.5 \pm 2.2	95.9 \pm 2.0	96.6	84.8	86.0

TABLE 8

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 [1].

4.2.2 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* [17]. 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 the classes 26-40 as unknowns in the target domain. We use the provided DeCAF features (DeCAF7). Following the unsupervised protocol described in [68], 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 9 are consistent with the Office dataset. ATI outperforms the baseline and the other methods by +4.1% for the closed set and by +5.3% for the open set. ATI- λ obtains the best accuracies for the open set.

4.2.3 Sparse Cross-Dataset Analysis

We also introduce an open set evaluation using the *sparse* set-up from [17] with the datasets *Caltech101* (C), *Pascal07* (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

	B→C		B→I		B→S		C→B		C→I		C→S	
	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 [67]	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 [3]	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 [66]	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 [27]	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-λ	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 [67]	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 [3]	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 [66]	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 [27]	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-λ	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 9

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

	C→O	C→P	O→C	O→P	P→C	P→O	AVG.
<i>shared classes</i>	8	7	8	4	7	4	
<i>unknown / all (t)</i>	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 [67]	45.2	33.8	58.1	31.1	63.4	61.1	48.8
gfk [3]	46.4	36.2	61.0	29.7	79.1	72.6	54.2
SA [66]	46.4	36.8	61.1	30.2	79.8	71.1	54.2
CORAL [27]	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-λ	51.5	52.0	63.4	39.1	81.1	71.1	59.7

TABLE 10

Unsupervised open set domain adaptation on the sparse set-up from [17].

	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-λ	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-λ-N ₁	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 11

Semi-supervised open set domain adaptation on the sparse set-up from [17] with 3 labelled target samples per shared class.

class). For each domain shift, we take all samples of the shared classes and consider all other samples as unknowns. Table 10 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 10. ATI outperforms the baseline and the other methods by +5.3% for this highly unbalanced open set protocol. ATI-λ 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 [1]. The average and standard deviation over 5 random splits are reported in Table 11. While ATI improves over the baseline

trained on the source and target samples together (st) by +4.5%, ATI-λ and the locality constraints with one neighbour boost the performance further. ATI-λ-N₁ improves the accuracy of the baseline by +5.1%.

4.2.4 Action recognition

We extend the applicability of our technique to the field of action recognition in video sequences. We introduce an open set domain adaptation protocol between the *Kinetics Human Action Video Dataset* [20] (Kinetics) and the *UCF101 Action Recognition Dataset* [21] (UCF101). The Kinetics dataset is used as source domain and contains a total of 400 human action classes. The UCF101 dataset serves as target domain including 101 action categories, mainly of sports events. Since the labels of the same action differ between the datasets, e.g., *massaging persons head* (Kinetics) and *head massage* (UCF101), we manually map the class labels between the datasets. Additionally, we also merge all action classes in one datasets if they correspond to a single class in the other dataset, e.g., *dribbling basketball*, *playing basketball*, *shooting basketball* (Kinetics) are merged and associated to *basketball* (UCF101). We finally obtain an open set protocol with 66 shared action classes. The list of shared classes, as well as all unrelated categories between both datasets, are provided in the supplemental material.

For action recognition, we use the features extracted from the 5c layer of the spatial and temporal stream of the I3D model [69], which is pretrained on Kinetics [20]. We forward the complete video sequences through the spatial and temporal stream of I3D [69] and the 5c layer of each stream provides a $7 \times 7 \times 1024$ output for a temporal fragment. We then apply spatial average pooling using a 7×7 kernel and average over time to obtain a 1024-dimensional feature vector from both the spatial and temporal stream of the I3D model [69]. Finally, the feature vectors from the spatial and temporal streams are concatenated to get a single 2048-dimensional feature vector per video sequence.

Unsupervised domain adaptation. In the unsupervised setting, we evaluate our method by taking all source samples in a single run. Table 12 shows that the proposed approach out-

Kinetics → UCF101						
LSVM	TCA [67]	gkf [3]	SA [66]	CORAL [27]	ATI	ATI-λ
64.9	71.2	64.9	65.1	69.4	76.6	76.9

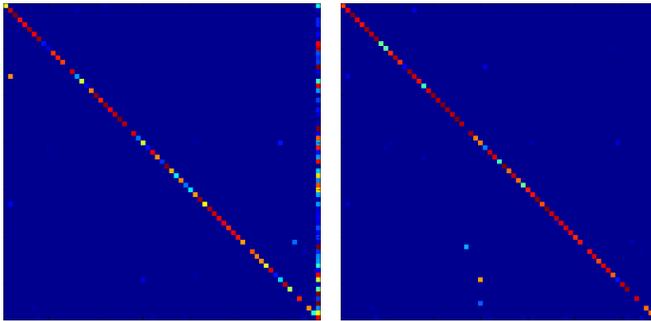
TABLE 12

Unsupervised open set domain adaptation for action recognition.

Kinetics → UCF101			
LSVM (st)	ATI	ATI-λ	ATI-λ-N1
73.5±0.5	84.1±0.7	84.2±0.8	84.5±0.6

TABLE 13

Semi-supervised open set domain adaptation for action recognition.



(a) No adaptation (LSVM): 64.9%

(b) ATI-λ: 76.9%

Fig. 6. Confusion matrices without (a) and with adaptation (b) for the 66 shared classes and unknowns (last row and last column) for the open set protocol for *Kinetics* [20] and *UCF101* [21]. Many instances of the shared classes in the target domain are wrongly classified as unknown instances (last column) if domain adaptation is not applied. The figure is best viewed by zooming in.

performs the baseline and other approaches. ATI-λ achieves the highest accuracy and improves the accuracy by +12.0% compared to LSVM. The resulting confusion matrices of LSVM and ATI-λ are shown in Fig. 6. LSVM misclassifies many instances of shared classes in the target domain as unknown instances (last column of confusion matrix), which is a well-known problem for open set recognition. Although ATI-λ does not resolve this problem completely, it reduces this effect substantially.

Semi-supervised domain adaptation. We extend the unsupervised protocol to evaluate our method on a semi-supervised setting by labelling 3 target samples per shared class. We report the average accuracies of 5 random splits in Table 13. Like in the previous semi-supervised experiments, ATI-λ-N1 obtains the best classification accuracies, outperforming the baseline without adaptation, LSVM (st), by +11.0%.

4.2.5 Synthetic data

We also introduce another open set protocol with a domain shift between synthetic and real data. In this case, we take 152,397 synthetic images of the VISDA'17 challenge [19] as source domain and 5970 instances of real images from the training data of the Pascal3D dataset [70] as target domain. Since both datasets contain several types of vehicles, we obtain 6 shared classes, namely, *aeroplane*, *bicycle*, *bus*, *car*, *motorbike* and *train*, within the 12 categories of each dataset. Following the protocol used in Section 4.2.1, we extract deep features from the fully connected layer-7 (fc7) from the AlexNet model [43] with 4096 dimensions. In addition, we also extract features from the VGG-16 model [71] to evaluate the impact of using deeper features.

The results of the classification task are shown in Table 14. The proposed domain adaptation method achieves the best results for both types of CNN features. When we

VISDA → Pascal3D							
	LSVM	TCA [67]	gkf [3]	SA [66]	CORAL [27]	ATI	ATI-λ
AlexNet	48.0	49.7	50.1	51.2	52.0	61.1	61.4
VGG-16	53.6	55.0	55.2	56.5	60.0	72.0	71.9

TABLE 14

Open set domain adaptation using synthetic images from the VISDA'17 challenge [19] as source and real data from the Pascal3D dataset [70] as target dataset. There are 6 shared classes between both datasets.

compare the performance of the deep features from AlexNet and VGG-16, the accuracy of the baseline (LSVM) increases by +5.6% when using the deeper network VGG-16 instead of AlexNet. ATI and ATI-λ, however, benefit even more from the deeper architecture. For instance, the accuracy of ATI-λ increases by +10.5%. This coincides with the observation that deeper networks have a stronger linearisation effect on manifolds of image domains [14], [15] than shallow networks. Since the proposed approach learns a linear mapping from the feature space of the source domain to the feature space of the target domain, it benefits from a better linearisation. The confusion matrices of the classification task with features extracted from the VGG-16 model are shown in Fig. 7. ATI-λ improves the overall accuracy of LSVM by +18.3% since it resolves confusions between similar classes. For instance, LSVM frequently misclassifies *bicycle* as *motorbike* and *car* as instances of trucks, which are part of the unknown class.

4.3 Closed set domain adaptation

We also report the accuracies of our method for popular domain adaptation datasets using the standard closed set protocols, where all classes are known in both domains.

4.3.1 Office dataset

For the *Office* dataset [1], we run experiments for the 6 domain shifts of the three provided datasets and use deep features extracted from the fc7 feature map from the AlexNet [43] and VGG-16 [71] models.

Unsupervised domain adaptation. For unsupervised domain adaptation, we first report the results for the protocol from [1], where we run 5 experiments for each domain shift using randomised samples of the source dataset. The results are shown in Table 15, where we compare our method with generic domain adaptation methods, i.e., TCA [67], gkf [3], SA [66] and CORAL [27] using AlexNet features. The results are in accordance with the observations from Section 4.2.1. While ATI outperforms all generic domain adaptation methods in average and ATI-λ performs slightly better than ATI, ATI-λ-N1 decreases the accuracy in the unsupervised setting. In addition, we also include the accuracies of using nearest neighbours without domain adaptation, NN, which reports significant lower accuracies than LSVM. LSVM also outperforms NN in other closed set evaluation protocols by a large margin.

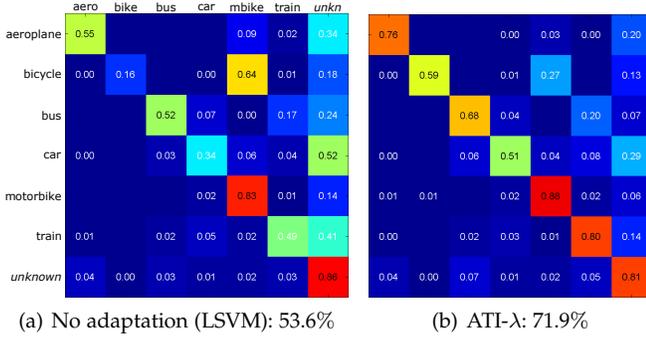


Fig. 7. Confusion matrices without (a) and with adaptation (b) for an open set classification task with 6 shared classes and a domain shift between synthetic [19] (source) and real [70] (target) data. The features are extracted from the fc7 layer of the VGG-16 model [71].

	A→D	A→W	D→A	D→W	W→A	W→D	AVG.
NN	51.3±1.4	45.7±2.1	26.0±0.9	65.5±1.4	28.0±0.5	69.8±1.8	47.7
LSVM	62.3±3.8	55.8±3.1	42.8±1.6	90.1±0.6	41.2±0.4	92.6±1.5	64.1
TCA [67]	60.3±4.0	54.7±3.0	49.4±1.6	90.7±0.4	46.9±2.3	92.0±0.9	65.7
gfk [3]	61.3±3.7	55.7±3.0	45.6±1.6	90.6±0.4	43.1±2.3	93.4±0.9	65.0
SA [66]	60.6±3.5	55.0±3.1	47.3±1.6	90.9±0.6	44.4±1.4	93.3±0.8	65.3
CORAL [27]	64.4±3.9	58.9±3.3	52.1±1.2	92.6±0.3	50.0±1.0	94.0±0.6	68.7
ATI	67.6±3.0	62.3±3.1	54.8±1.3	90.3±0.8	52.4±2.1	92.6±1.7	70.0
ATI- λ	67.3±2.3	62.6±2.5	55.2±2.6	90.1±0.6	53.4±2.5	92.7±2.5	70.2
ATI- λ - N_1	64.6±2.9	60.9±1.3	51.9±1.9	90.2±0.9	48.1±1.6	93.7±2.1	68.2

TABLE 15
Comparison on the unsupervised Office dataset [1] with 31 shared classes and 6 domain shifts using the protocol from [1] and features from the AlexNet model (fc7 layer).

We also compare our method with current state-of-the-art CNN-based domain adaptation methods [6], [9], [47], [48], [50], [51]. In this case, we report the accuracies when all source samples are used in a single run as described by [32]. As shown in Table 16, our method achieves competitive results even for the standard closed set protocol.

Semi-supervised domain adaptation. We also evaluate our approach for semi-supervised domain adaptation on the *Office* dataset. We follow the protocol from [1] and report the accuracies and standard deviations over 5 runs with random samples. In the first experiment with AlexNet features, we also include ATI- λ - N_2 with locality constraints using 2 nearest neighbours and compare our approach with state-of-the-art CNN-based methods [46], [47], [72]. As in Section 4.2.1, we train the SVMs on the transformed source samples and labelled target samples (st). The results are reported in Table 17.

Our method achieves the same average accuracy as MMC [72] and performs slightly worse than [10] for the VGG-16 features. In addition, we report the accuracy for AlexNet features when the mapping W (6) is estimated using only the labelled target samples without solving the individual assignments (1). This variant is denoted by ATI (labels t) and performs worse than ATI.

4.3.2 Office+Caltech dataset

We also evaluate our approach on the extended version of the Office evaluation set [3], which includes the additional *Caltech* (C) dataset. This results in 12 domain shifts, but reduces the amount of shared classes to only 10. As shown

	A→D	A→W	D→A	D→W	W→A	W→D	AVG.
AlexNet features (fc7)							
NN	55.9	49.7	27.4	75.3	31.5	86.2	54.3
LSVM	65.7	60.3	43.2	94.7	44.0	98.9	67.8
DAN [47]	66.8	68.5	50.0	96.0	49.8	99.0	71.7
DAH [51]	66.5	68.3	55.5	96.1	53.0	98.8	73.0
RTN [48]	71.0	73.3	50.5	96.8	51.0	99.6	73.7
BP [6]	-	73.0	-	96.4	-	99.2	-
ADDA [9]	-	75.1	-	97.0	-	99.6	-
ATI	70.3	68.7	55.3	95.0	56.9	98.7	74.2
ATI- λ	69.0	67.0	56.2	95.0	56.9	98.7	73.8
VGG-16 features (fc7)							
NN	61.3	55.4	33.1	78.6	49.4	88.8	61.1
LSVM	76.1	68.6	55.3	95.9	61.5	99.6	76.2
DAN [47]	74.4	76.0	61.5	95.9	60.3	98.6	77.8
AutoDIAL [50]	82.3	84.2	64.6	97.9	64.2	99.9	82.2
ATI	80.6	81.4	67.1	96.1	66.4	99.3	81.8
ATI- λ	80.8	81.3	66.9	96.1	66.5	98.9	81.8

TABLE 16

Comparison on the unsupervised Office dataset [1] with 31 shared classes and 6 domain shifts taking all source samples as in [32].

	A→D	A→W	D→A	D→W	W→A	W→D	AVG.
AlexNet features (fc7)							
LSVM (st)	82.6±5.5	77.0±2.5	63.4±1.6	94.0±0.8	61.8±1.1	96.3±0.8	79.2
DDC [46]	-	84.1±0.6	-	95.4±0.4	-	96.3±0.3	-
DAN [47]	-	85.7±0.3	-	97.2±0.2	-	96.4±0.2	-
MMC [72]	86.1±1.2	82.7±0.8	66.2±0.3	95.7±0.5	65.0±0.5	97.6±0.2	82.2
ATI (labels t)	85.0±2.1	78.3±2.3	63.6±1.5	94.0±0.8	62.3±0.9	96.4±0.8	79.9
ATI	85.5±2.9	82.4±1.1	65.1±1.3	93.4±0.9	65.6±1.5	95.7±1.1	81.3
ATI- λ	85.6±2.6	82.6±0.5	65.3±1.3	93.3±1.0	65.7±1.7	95.7±1.1	81.4
ATI- λ - N_1	88.1±1.7	83.1±2.3	66.0±1.4	93.9±1.2	65.9±1.5	96.2±0.8	82.2
ATI- λ - N_2	87.0±3.5	84.6±3.5	65.3±1.0	93.6±1.2	65.9±1.8	95.8±1.3	82.0
VGG-16 features (fc7)							
LSVM (st)	86.1±1.5	83.4±1.2	67.9±1.0	96.1±0.7	67.1±0.6	96.6±1.0	82.9
SO [52]	84.5±1.7	86.3±0.8	65.7±1.7	97.5±0.7	66.5±1.0	95.5±0.6	82.7
CCSA [10]	88.2±1.0	89.0±1.2	72.1±1.0	97.6±0.4	71.8±0.5	96.4±0.8	85.8
ATI- λ - N_1	90.3±1.9	88.0±1.4	70.8±0.9	95.1±0.7	70.3±2.0	96.3±0.9	85.1

TABLE 17

Comparison on the semi-supervised Office dataset [1] with 31 shared classes and 6 domain shifts, following the protocol from [1].

in Table 18, our method obtains very competitive results with AlexNet features, outperforming in overall the generic domain adaptation method [27] and 3 out of 4 CNN-based methods. If features from a deeper network such as VGG-16 are used, our method obtains the best overall results.

4.3.3 Dense Testbed for Cross-Dataset Analysis

We also present an evaluation on the Dense dataset of the Testbed for Cross-Dataset Analysis [68] using the provided DeCAF features. This protocol comprises 12 domain shifts between the 4 datasets *Bing* (B), *Caltech* (C), *ImageNet* (I) and *Sun* (S), which share 40 classes. Following the protocol described in [68], 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 19 show that ATI- λ outperforms other generic domain adaptation methods.

4.3.4 Sentiment Analysis

To show the behaviour of our method with a different type of feature descriptor, we also present an evaluation on the *Sentiment analysis* dataset [18]. This dataset gathers reviews from Amazon for four products: *books* (B), *DVDs* (D), *electronics* (E) and *kitchen appliances* (K). Each domain

	A→C	A→D	A→W	C→A	C→D	C→W	
AlexNet features (fc7)							
NN	78.4	78.1	71.7	90.7	84.4	80.8	
LSVM	83.3	84.1	77.5	91.8	89.1	82.3	
CORAL [27]	83.2	86.5	79.6	91.4	86.6	82.1	
BP [6]	84.6	92.3	90.2	91.9	92.8	93.2	
DDC [46]	83.5	88.4	83.1	91.9	88.8	85.4	
DAN [47]	84.1	91.1	91.8	92.0	89.3	90.6	
RTN [48]	88.1	95.5	95.2	93.7	94.2	96.9	
ATI	86.5	92.8	88.7	93.8	89.6	93.6	
ATI-λ	87.1	90.6	90.7	93.4	85.4	93.4	
VGG-16 features (fc7)							
NN	86.7	84.4	83.4	91.4	88.2	88.0	
LSVM	87.8	88.7	87.2	93.3	91.8	91.4	
ATI	91.0	92.4	95.9	94.7	93.1	97.4	
ATI-λ	90.4	92.4	91.4	94.5	93.9	96.0	
	D→A	D→C	D→W	W→A	W→C	W→D	AVG
AlexNet features (fc7)							
NN	64.2	58.6	89.0	63.2	58.8	95.4	76.1
LSVM	79.4	70.2	97.9	80.0	72.7	100.0	84.0
CORAL [27]	87.3	77.5	99.3	85.2	76.1	100.0	86.2
BP [6]	84.0	74.9	97.8	86.9	77.3	100.0	88.2
DDC [46]	89.0	79.2	98.1	84.9	73.4	100.0	87.1
DAN [47]	90.0	80.3	98.5	92.1	81.2	100.0	90.1
RTN [48]	93.8	84.6	99.2	95.5	86.6	100.0	93.4
ATI	93.4	85.9	98.9	93.6	86.3	100.0	91.9
ATI-λ	93.6	85.8	99.3	93.6	86.1	100.0	91.8
VGG-16 features (fc7)							
NN	78.9	75.0	95.2	80.9	78.5	100.0	85.6
LSVM	82.5	77.9	98.4	87.8	84.9	100.0	89.3
ATI	93.7	89.8	98.1	95.1	90.3	99.5	94.3
ATI-λ	94.6	89.4	98.4	95.3	89.4	99.6	93.8

TABLE 18

Classification accuracies on the unsupervised Office+Caltech dataset [3] with 10 shared classes and 12 domain shifts using deep features. We take all source samples on a single run [32].

	B→C	B→I	B→S	C→B	C→I	C→S	
LSVM	63.8±2.2	57.4±0.7	20.2±1.0	38.3±0.8	62.9±0.9	21.7±1.6	
TCA [67]	53.8±1.3	49.1±1.1	17.1±1.1	35.6±1.8	59.2±0.8	18.9±1.2	
gfk [3]	63.4±1.8	57.2±1.1	20.6±1.3	38.3±0.9	62.9±1.2	21.7±1.4	
SA [66]	63.0±1.9	57.1±1.4	20.2±1.4	38.3±0.9	62.8±1.0	21.5±1.2	
CORAL [27]	63.9±2.1	57.8±0.8	20.4±2.0	38.3±0.8	63.4±0.9	22.5±1.2	
ATI	69.1±1.3	62.4±1.9	23.4±1.1	39.0±1.4	66.9±1.2	25.2±0.9	
ATI-λ	69.4±1.4	62.9±1.3	23.6±1.0	39.0±1.4	66.9±1.1	25.3±0.9	
	I→B	I→C	I→S	S→B	S→C	S→I	AVG
LSVM	39.3±1.4	70.8±1.5	24.6±1.8	16.6±1.0	26.1±2.0	26.3±0.7	39.0
TCA [67]	36.4±1.2	66.3±2.3	22.2±1.4	13.8±1.4	23.2±1.5	23.2±1.5	34.9
gfk [3]	38.8±1.3	70.9±1.1	24.4±1.4	16.3±0.9	26.7±1.8	26.1±1.0	38.9
SA [66]	39.0±1.3	71.1±1.3	24.2±1.4	16.0±0.9	26.8±1.9	26.4±1.1	38.9
CORAL [27]	39.0±1.2	71.2±1.3	24.9±1.6	16.8±1.0	27.4±2.2	27.7±0.5	39.4
ATI	39.7±1.8	74.4±1.6	25.9±2.1	18.3±1.1	37.1±3.2	35.0±1.0	42.8
ATI-λ	39.8±1.8	74.8±1.5	25.8±2.0	18.7±0.7	37.4±2.9	34.8±0.8	43.2

TABLE 19

Testbed dataset [17] with 40 common classes and 12 domain shifts.

contains 1000 reviews labelled as *positive* and another set of 1000 reviews as *negative*. We use the data provided by [39], which extracts bag-of-words features from the 400 words with the largest mutual information across domains. We report the mean accuracy over 20 splits, where for each run 1600 samples are randomly selected for training and the other 400 for testing. The results in Table 20 show that our approach not only works very well for image and video data, but it can also be applied to other types of data. This demonstrates the versatility of the proposed approach.

	B→E	D→B	E→K	K→D	AVG.
LSVM	75.5±1.6	78.2±2.5	83.1±1.8	73.3±1.8	77.5
TCA [67]	76.6±2.2	78.5±1.6	83.8±1.5	75.0±1.4	78.5
gfk [3]	77.0±2.0	79.2±1.8	83.7±1.7	73.7±1.9	78.4
SA [66]	75.9±1.9	78.4±2.1	83.0±1.7	72.1±1.9	77.4
CORAL [27]	76.2±1.7	78.4±2.0	83.1±2.0	74.2±3.0	78.0
ATI	79.9±2.0	79.2±1.9	83.7±2.1	75.6±1.9	79.6
ATI-λ	79.6±1.4	79.0±1.8	83.6±2.1	74.4±1.7	79.2

TABLE 20

Accuracies of 4 domain shifts on the Sentiment dataset [18] using the bag-of-words features and the protocol from [39].

5 CONCLUSIONS

We have introduced the concept of open set domain adaptation in the context of image classification and action recognition. In contrast to closed set domain adaptation, we do not assume that all instances in the source and target domain belong to the same set of classes, but allow that each domain contains instances of classes that are not present in the other domain. We furthermore proposed an approach for unsupervised and semi-supervised domain adaptation that achieves state-of-the-art results for open sets and competitive results for closed sets. In particular, the flexibility of the approach, which can be used for images, videos and other types of data, makes the approach a versatile tool for real-world applications.

ACKNOWLEDGMENTS

The work has been supported by the ERC Starting Grant ARCA (677650) and the DFG projects GA 1927/2-2 (DFG Research Unit FOR 1505 Mapping on Demand) and GA 1927/4-1 (DFG Research Unit FOR 2535 Anticipating Human Behavior).

REFERENCES

- [1] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision*, 2010, pp. 213–226.
- [2] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 999–1006.
- [3] 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*, 2012, pp. 2066–2073.
- [4] 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.
- [5] 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*, vol. 109, no. 1-2, pp. 28–41, 2014.
- [6] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.
- [7] 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*, 2015, pp. 4121–4129.
- [8] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European Conference on Computer Vision*, 2016, pp. 597–613.
- [9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2962–2971.

- [10] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *IEEE International Conference on Computer Vision*, 2017, pp. 5716–5726.
- [11] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.
- [12] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.
- [13] A. Bendale and T. E. Boult, "Towards open set deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 1563–1572.
- [14] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai, "Better mixing via deep representations," in *International Conference on Machine Learning*, vol. 28, 2013, pp. 552–560.
- [15] P. Upchurch, J. R. Gardner, G. Pleiss, R. Pless, N. Snaveley, K. Bala, and K. Q. Weinberger, "Deep feature interpolation for image content changes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6090–6099.
- [16] P. Busto and J. Gall, "Open set domain adaptation," in *IEEE International Conference on Computer Vision*, 2017, pp. 754–763.
- [17] 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*, 2014, pp. 18–31.
- [18] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 440–447.
- [19] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," *CoRR*, vol. abs/1710.06924, 2017.
- [20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, 2017.
- [21] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.
- [22] 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*, 2013, pp. 769–776.
- [23] 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*, 2013, pp. 692–699.
- [24] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 361–368.
- [25] H. Xu, J. Zheng, and R. Chellappa, "bridging the domain shift by domain adaptive dictionary learning," in *British Machine Vision Conference*, 2015, pp. 96.1–96.12.
- [26] B. Sun and K. Saenko, "From virtual to reality: Fast adaptation of virtual object detectors to real domains," in *British Machine Vision Conference*, 2014.
- [27] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI Conference on Artificial Intelligence*, 2015, pp. 2058–2065.
- [28] S. Herath, M. T. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3956–3965.
- [29] B. Gholami, O. Rudovic, and V. Pavlovic, "Punda: Probabilistic unsupervised domain adaptation for knowledge transfer across visual categories," in *IEEE International Conference on Computer Vision*, 2017, pp. 3601–3610.
- [30] G. Csurka, B. Chidlowkii, S. Clinchant, and S. Michel, "Unsupervised domain adaptation with regularized domain instance denoising," in *European Conference on Computer Vision*, 2016, pp. 458–466.
- [31] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5150–5158.
- [32] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Advances in Neural Information Processing Systems*, 2013, pp. 1286–1294.
- [33] 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*, 2006, pp. 513–520.
- [34] Y. Aytar and A. Zisserman, "Tabula rasa: model transfer for object category detection," in *IEEE International Conference on Computer Vision*, 2011, pp. 2252–2259.
- [35] 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*, 2011, pp. 1785–1792.
- [36] 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.
- [37] P. Busto, J. Liebelt, and J. Gall, "Adaptation of synthetic data for coarse-to-fine viewpoint refinement," in *British Machine Vision Conference*, 2015, pp. 14.1–14.12.
- [38] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Conference on empirical methods in natural language processing*, 2006, pp. 120–128.
- [39] 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*, 2013, pp. 222–230.
- [40] E. Sangineto, "Statistical and spatial consensus collection for detector adaptation," in *European Conference on Computer Vision*, 2014, pp. 456–471.
- [41] 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*, vol. 32, no. 5, pp. 770–787, 2010.
- [42] T. Tommasi and B. Caputo, "Frustratingly easy NBNN domain adaptation," in *IEEE International Conference on Computer Vision*, 2013, pp. 897–904.
- [43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [44] 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*, 2014, pp. 647–655.
- [45] M. Ghifary, W. B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *The Pacific Rim International Conferences on Artificial Intelligence*, 2014, pp. 898–904.
- [46] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014.
- [47] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, 2015, pp. 97–105.
- [48] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.
- [49] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 945–954.
- [50] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Autodial: Automatic domain alignment layers," in *IEEE International Conference on Computer Vision*, 2017, pp. 5077–5085.
- [51] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5385–5394.
- [52] P. Koniusz, Y. Tas, and F. Porikli, "Domain adaptation by mixture of alignments of second-or higher-order scatter tensors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7139–7148.
- [53] H. Lu, L. Zhang, Z. Cao, W. Wei, K. Xian, C. Shen, and A. van den Hengel, "When unsupervised domain adaptation meets tensor representations," in *IEEE International Conference on Computer Vision*, 2017, pp. 599–608.
- [54] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

- [55] F. Li and H. Wechsler, "Open set face recognition using transduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1686–1697, 2005.
- [56] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *CoRR*, vol. abs/1502.00873, 2015.
- [57] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European Conference on Computer Vision*, 2014, pp. 393–409.
- [58] R. Zhang and D. N. Metaxas, "Ro-svm: Support vector machine with reject option for image categorization," in *British Machine Vision Conference*, 2006, pp. 1209–1218.
- [59] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *Journal of Machine Learning Research*, vol. 9, no. 6, pp. 1823–1840, 2008.
- [60] E. Gavves, T. Mensink, T. Tommasi, C. G. M. Snoek, and T. Tuytelaars, "Active transfer learning with zero-shot priors: Reusing past datasets for future tasks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2731–2739.
- [61] T. Achterberg, "SCIP: Solving constraint integer programs," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009.
- [62] L. Kaufman and F. Broeckx, "An algorithm for the quadratic assignment problem using bender's decomposition," *European Journal of Operational Research*, vol. 2, no. 3, pp. 207–211, 1978.
- [63] K. Svanberg, "A class of globally convergent optimization methods based on conservative convex separable approximations," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 555–573, 2002.
- [64] S. G. Johnson, "The NLOpt nonlinear-optimization package," 2007–2010. [Online]. Available: <http://ab-initio.mit.edu/nlopt>
- [65] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.
- [66] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, 2013, pp. 2960–2967.
- [67] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *International Joint Conference on Artificial Intelligence*, 2009, pp. 1187–1192.
- [68] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," *CoRR*, 2015.
- [69] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4724–4733.
- [70] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond Pascal: A benchmark for 3D object detection in the wild," in *IEEE Winter Conference on Applications of Computer Vision*, 2014, pp. 75–82.
- [71] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [72] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

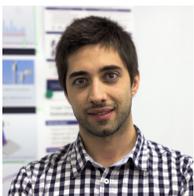


Ahsan Iqbal obtained his master degree in Computer Science from the University of Bonn in 2017. He did an internship in Amazon Berlin from December 2016 to March 2017. Since April 2017, he is a Ph.D. student at the University of Bonn. His research interests are action recognition and action detection.



Juergen Gall obtained his B.Sc. and his Masters degree in mathematics from the University of Wales Swansea (2004) and from the University of Mannheim (2005). In 2009, he obtained a Ph.D. in computer science from the Saarland University and the Max Planck Institut für Informatik. He was a postdoctoral researcher at the Computer Vision Laboratory, ETH Zurich, from 2009 until 2012 and senior research scientist at the Max Planck Institute for Intelligent Systems in Tübingen from 2012 until 2013. Since 2013,

he is professor at the University of Bonn and head of the Computer Vision Group.



Pau Panareda Busto received his B.Sc. and Masters degrees in computer engineering from the Technical University of Catalonia (2010) and his Masters degree in media informatics from the RWTH Aachen University (2013). Since September 2013, he has been a Ph.D. candidate at the University of Bonn in collaboration with Airbus Group. His research interests include computer vision, computer graphics and machine learning.