Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Improved multi-source domain adaptation by preservation of factors

Sebastian Schrom, Stephan Hasler, Jürgen Adamy

2021

Preprint:

Copyright Elsevier 2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

Image and Vision Computing 112 (2021) 104209

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Improved multi-source domain adaptation by preservation of factors

Sebastian Schrom [a,*], Stephan Hasler [b], Jürgen Adamy [a]

[a] Control Methods and Robotics Lab, TU Darmstadt, 64283 Darmstadt, Germany
[b] Honda Research Institute Europe, Carl-Legien-Str. 30, 63065 Offenbach, Germany

ARTICLE INFO

Article history:
Received 17 September 2020
Received in revised form 4 May 2021
Accepted 5 May 2021
Available online 08 May 2021

Keywords:
Domain adaptation
Multi-source domain adaptation
Adversarial domain adaptation
Negative transfer
Visual factors
Domain factors

ABSTRACT

Domain Adaptation (DA) is a highly relevant research topic when it comes to image classification with deep neural networks. Combining multiple source domains in a sophisticated way to optimize a classification model can improve the generalization to a target domain. Here, the difference in data distributions of source and target image datasets plays a major role. In this paper, we describe based on a theory of visual factors how real-world scenes appear in images in general and how recent DA datasets are composed of such. We show that different domains can be described by a set of so called domain factors, whose values are consistent within a domain, but can change across domains. Many DA approaches try to remove all domain factors from the feature representation to be domain invariant. In this paper we show that this can lead to negative transfer since task-informative factors can get lost as well. To address this, we propose Factor-Preserving DA (FP-DA), a method to train a deep adversarial unsupervised DA model, which is able to preserve specific task relevant factors in a multi-domain scenario. We demonstrate on CORe50 how such factors can be identified by standard one-to-one transfer experiments between single domains combined with PCA. By applying FP-DA, we show that the highest average and minimum performance can be achieved. We also report improved performance for an adapted version of the OpenLORIS object dataset.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

The optimization of a well generalizing image classifier based on deep learning requires generally a considerable amount of training images that show high variations in visual class appearances. In general each dataset comes with a dataset bias [1] that is caused by a subset of all possible variations. If training data and application data represent samples from the same distribution, i.e. have a similar dataset bias, this difference mostly has minor influence on the classification performance. If both are composed of samples from different data distributions we usually speak of domains. In this case a significant decrease in classification performance can be expected if the domains are not handled explicitly [2]. The research area of Domain Adaptation (DA) tries to obtain classification models that are able to generalize well to new domains. The new domains are usually called target domains and come with few or no labeled data, while the domains that are used for supervised training of the network are called source domains.

The datasets that are used for verification of DA approaches in computer vision, i.e. the different source and target domains, are often generated under human influence to investigate a specific type of variation that the classification model should deal with. As we will show in this paper, the difference between the datasets is mostly based on values of visual factors, like types of lighting, background or the camera. A domain is consequently defined by a combination of these factors where the value of a factor is consistent in one domain, but can change across domains. We call these factors domain factors. An example is given in Fig. 1, where the domain factors are strongly related to recording locations. Using such a theory of visual factors allows to describe more formal and understandable aspects of datasets, to predict effects of machine learning, in particular of DA approaches, and to interpret and discuss experimental results with more human interpretability than the unspecific notion of different data distributions allows.

In general, there are two main types of DA approaches. For the first type, a layer with a simple mathematical operation is designed to normalize out a given aspect explicitly. The parameters of the layer can be determined for the target domain after training. This generally only works for biases that are known or can easily be modeled, and includes methods like dataset based RGB mean subtraction and batch normalization. The second type of approaches is based on learned normalization that allows to handle general complex bias differences that can hardly be modeled. Deep learning approaches usually enforce domain invariance at a certain feature layer in the optimization process by an additional loss function. Training usually involves unsupervised data of the target domain, but also generalization of multiple source domains to a completely unknown target domain is expected to improve in this way. In this work we want to focus on such learned normalization approaches.

* Corresponding author.
E-mail address: sebastian.schrom@rmr.tu-darmstadt.de (S. Schrom).

https://doi.org/10.1016/j.imavis.2021.104209
0262-8856/© 2021 Elsevier B.V. All rights reserved.

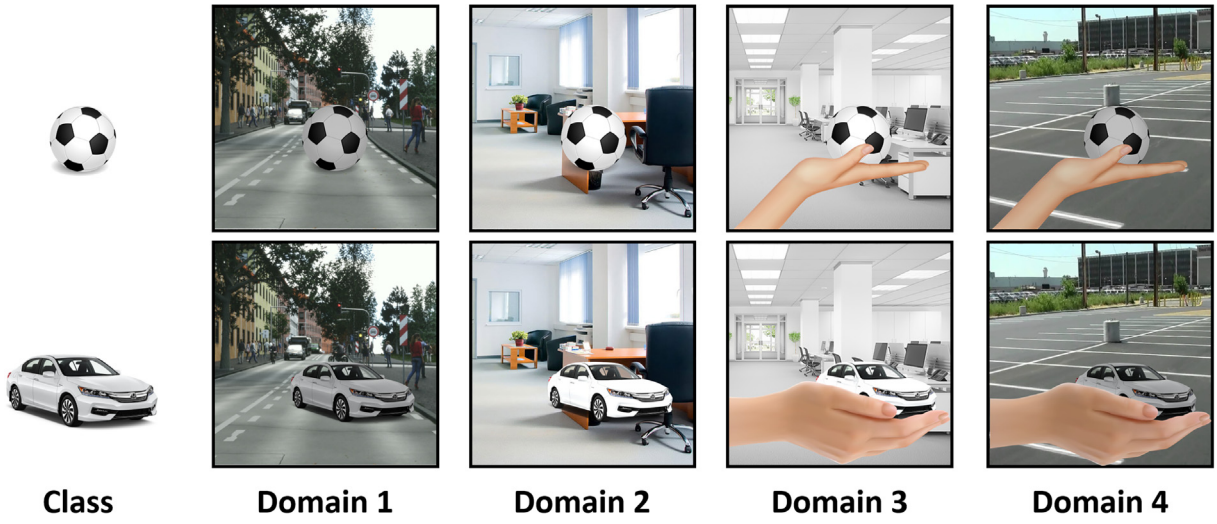

Fig. 1. The variations in the shown domain dataset can be described by different factors, as e.g. *'background'* ∈ {street, office 1, office 2, parking}, *'location'* ∈ {indoor, outdoor}, *'lighting'* ∈ {bright, dark}, *'hand'* ∈ {present, absent}, *'hand posture'* ∈ {flat, hollow}. We call a factor a domain factor, if it has a constant value within each domain, but different values across all domains. All of the mentioned factors are domain factors, except the *'hand posture'*, whose value changes also along classes.

Adversarial DA approaches that are based on additionally attached domain classifiers are often used for learned normalization. Their goal is to classify the domain of a sample, however, they are integrated with a specific architectural or loss function modification. With this modification the obtained feature representation should not be informative about the domains given the training data. In other words, what standard adversarial DA tries to do, is to completely remove all domain factors from the feature representation. As it will be discussed in this paper, this can be harmful, if simultaneously task-informative factors are removed. In a scenario as depicted in Fig. 1 the domain factor *'hand'* would be removed through adversarial DA. This would simultaneously remove the factor *'hand posture'* which is task-informative in domain 3 and 4 and thus complicates the overall classification task. Standard adversarial DA ignores such correlations and can therefore cause this so called negative transfer. Experimental evidence for this negative transfer will be given in this paper. To counteract the negative transfer, we propose Factor-Preserving DA (FP-DA). FP-DA is an adversarial multi-source approach that allows to preserve a chosen factor by switching off competition between domains that differ in this factor. With this method we report increased classification accuracies in a leave-one-out domain setting, both in terms of average performance, but even more pronounced in the minimal target domain performance, which is the more important measure to look at for safety critical systems.

Our main contributions of this work can be summarized as follows:

- We introduce a theory of visual factors that allows to more specifically describe domain data and to interpret the effects of DA approaches.
- We propose a novel training method, FP-DA, that instead of removing all domain factors as in [3], allows preservation of a chosen factor during DA.
- We use PCA to analyze multi-domain scenarios and show how it helps to find factors to be preserved.
- We show the effectiveness of FP-DA on two multi-domain object classification datasets.

The remainder of this paper is structured as follows. We start with a formal definition of domains and its components, followed by the introduction of our theory of visual factors. There we show that scenes captured in images can be described by a superposition of factors and that domains are characterized by domain factors. This is followed by an overview of existing image datasets and DA approaches in the context of visual factors. After this we introduce the frequently used adversarial DA architecture by Ganin et al. [3] on which our adapted training method, FP-DA, is based on. We will prove the advantages of FP-DA using the example of the CORe50 dataset [4] which comes with 11 domains. Our first experiments investigate how DA influences the transfer in a multi-source domain constellation where 10 source domains and a single target domain are given. Here we will show the effects of negative transfer when naively applying DA. This is followed by single source transfer experiments to identify visual factors that are removed by DA and therefore cause significant errors. We show that Principal Component Analysis (PCA) can be a means for the identification of those. Using the obtained knowledge from these experiments we will finally present the best results with FP-DA in such a multi-source DA scenario. The effectiveness of FP-DA is subsequently further substantiated on an adapted version of the OpenLORIS object dataset [5].

2. Theory of visual factors

In this section we will first introduce the formal definition of domains that is mostly used by recent DA approaches. This is followed by the introduction of our theory of visual factors, to which we will refer throughout the remainder of this paper.

2.1. Formal definition of domains

Following the definitions of [2,6,7], we assume to have a set of n data samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where each sample $\mathbf{x}$ represents a multi-dimensional feature vector in the feature space $\mathcal{X}$. Each sample has a corresponding label y from $\mathcal{Y}$ given in $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. In classification tasks, the goal is to find the best approximation of the conditional distribution $P(\mathbf{Y}|\mathbf{X})$ of samples to class labels. However, this approximation strongly depends on the samples included in $\mathbf{X}$. DA approaches try to make this approximation more general and evaluate the transfer between sample sets, called domains, that come with different $\mathbf{X}$. Specifically here the training and test set are called source and target domain. The general formal definition of source and target domain, as described in [8], is given by $D^s = \{\mathcal{X}^s, P(\mathbf{X}^s)\}$ and $D^t = \{\mathcal{X}^t, P(\mathbf{X}^t)\}$. Both come with a related task, given by $\mathcal{T}^s = \{\mathcal{Y}^s, P(\mathbf{Y}^s|\mathbf{X}^s)\}$ and $\mathcal{T}^t = \{\mathcal{Y}^t, P(\mathbf{Y}^t|\mathbf{X}^t)\}$, which we assume to be identical here. In this

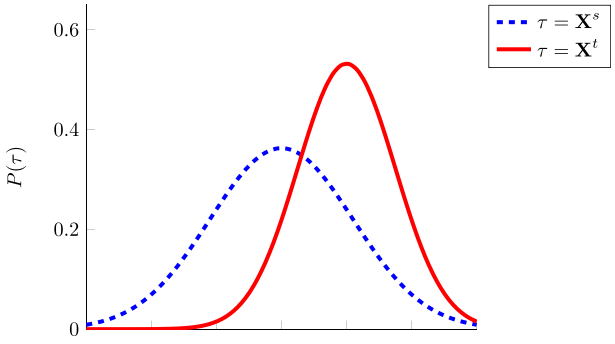

Fig. 2. Exemplary covariate shift between the marginal distributions of source and target domain sample sets $\mathbf{X}^s$ and $\mathbf{X}^t$.

work we consider only homogeneous DA, where the feature spaces $\mathcal{X}^s$ and $\mathcal{X}^t$ are identical, meaning that the difference between domains is related to the marginal distributions, i.e. $P(\mathbf{X}^s) \neq P(\mathbf{X}^t)$. This difference is mostly caused by a different sample selection bias [9] for both sample sets and is called covariate shift. A visualization is given in Fig. 2.

However, the notion of different data distributions does not allow to model the rich constellation of multiple domains comprehensively. Therefore we target to describe domains as combinations of a set of shared underlying factors. Based on relations between factors in the domains the effects of DA approaches can better be predicted and thus be influenced.

2.2. Domains in context of visual factors

We start from the general notion that each high-dimensional image $\mathbf{b}_i$ of a scene i can be generated by a mixing function $M(\mathbf{f}_i) = \mathbf{b}_i$ from an activation of factors $\mathbf{f}_i = (f_{1i}, f_{2i}, \ldots, f_{li})$. Each factor f_r can be a continuous or categorical variable. In general, many different meaningful combinations of the mixer M and the factors $\mathbf{F}$ can be found to model the same image data $\mathbf{I}$. Our goal here is not to analytically solve the factorization given a certain optimality criterium. Instead, we like to discuss more theoretically the properties of factors in relation to image classification tasks under domain transfer. This theory we will use later to describe image datasets and predict and influence the effects of DA approaches.

For the classification of an image $\mathbf{b}_i$, a feature extractor usually transforms the image to a feature vector $\mathbf{x}_i$, ideally of a feature space where classes can easily be separated. Based on $\mathbf{x}_i$, a label classifier predicts the class probabilities $\mathbf{y}_i$. In adversarial DA approaches often an additional domain classifier is attached to the feature extractor via a gradient reversal layer, which influences only backpropagation. The overall process with image generation and classification is depicted in Fig. 3.

In general, the factors can be categorized as task-informative or not task-informative. Task-informative factors strongly correlate with a certain class, but are not necessarily directly related to that class. Such as for example the factor *'hand posture'*, which can give clues about the ground-truth class of a presented object.

For domain classification, some factors can further be categorized as domain-informative. Domain-informative factors have independent of the ground-truth class a constant value for images from a certain domain, which changes across domains. Such factors are often associated with fixed recording locations of each domain, which consequently induce changing lighting and background related factor values across domains. A domain itself therefore can be described by a combination of factors whose values are constant within each domain, but can differ across domains. We call these factors the domain factors. A domain invariant feature extractor must therefore not consider domain factors, since these can lead to misclassification if a classifier uses features based on them and the domain changes. However, removing domain factors from the feature representation can also have adverse effects as we will show.

In general there are infinite possibilities on how to factorize scenes by visual factors. The factorization could for example either be human designed, by choosing certain factors manually, or estimated by a component analysis method based on a given image dataset. The choice ultimately influences the number of factors and the nameability, which is not necessarily given for the latter. Other aspects like the mutual dependence of individual factors, which consequently influences their values, and the controllability of a factor in a scene are also dependent on the chosen factorization. In the end, a well-chosen factorization might provide increased predictability and explainability of classification models handling new factor values or new combinations of known values. However, this also depends on the fineness of the factorization. While for some a discrimination of indoor and outdoor is enough, for others a more fine-grained factorization is required.

Exemplary views on how to factorize scenes can be found in scene rendering or photography. In the rendering view the factorization is based on human interpretable factors:

- Object factors: *'type'*, *'shape'*, *'color'*, *'texture'*, *'viewpoint'*, ...
- Light source factors: *'type'*, *'positioning'*, *'emission power'*, ...
- Camera factors: *'sensor'*, *'filter'*, *'processing'*, *'viewpoint'*, ...

The factorization and with it the number of factors is pre-defined by the rendering program. The factor values and their mutual dependence are fully controllable by a rendering artist. He decides for instance how chosen object colors appear in combination with placed light sources in the rendered scene.

The photographer view differs from the rendering view in reduced controllability and increased mutual dependence of the factors. While in natural images, the object in focus and the viewing angle are usually controlled by the photographer, there is less control over natural lighting conditions and context objects. Certain objects usually induce other context objects, just as a car would most likely induce asphalt.

The two presented views describe extremes regarding controllability of factor values. Sometimes, image datasets, especially those used for DA, describe a mixture of these. This is for example the case when objects are artificially placed in unnatural environments, where the target object related factors are fully controlled but the environment factors are not.

2.3. Effects of machine learning and DA in context of visual factors

In general, a test set defines what changes regarding factor values a classification model should be able to generalize to. First, there may be new, unseen factor values for certain class samples in the test set, and then there could be new combinations of already seen factor values. Both cases are depicted in Fig. 4.

New, unseen values in the test set could be related to factors of target objects, like different object colors, as well as to context objects factors, lighting factors, or others. A classification model requires internal class representations that ignore such factors to perform well on the test set. However, standard machine learning algorithms that are optimized on a training set are not forced to be invariant of such factors and could therefore misclassify samples if these factors are considered for classification. In DA, the information about the new factor values is often implicitly provided by unsupervised samples from the target domain. Using these samples in a sophisticated way during optimization can help to become invariant of the changing factors.

A second case is the occurrence of new combinations of values, while each value was already seen in other combinations during training. With standard machine learning it is likely that combinations are learned as a single representative feature, if factor values are exclusively shown in combination with specific values of other factors. Consequently, the learned model might fail on new combinations. If the

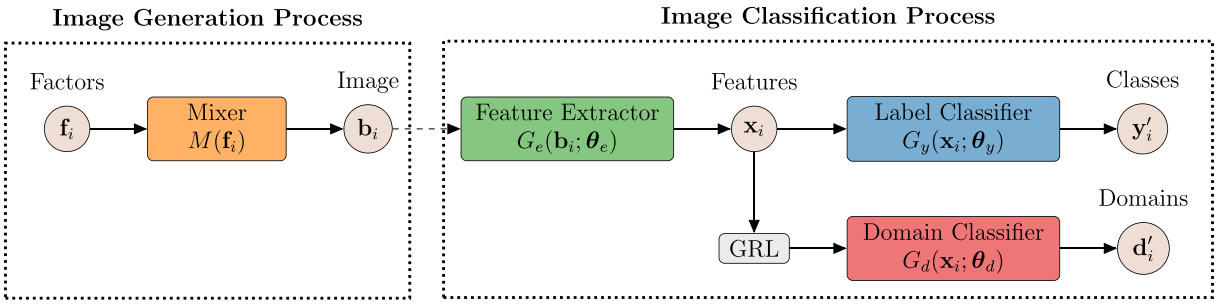

Fig. 3. Each image $\mathbf{b}_i$ can be thought as being generated by a complex mixing function M from an activation of factors $\mathbf{f}_i$. For image classification, in general a feature vector $\mathbf{x}_i$ is computed by a feature extractor with the mapping function $G_e(\mathbf{b}_i;\boldsymbol{\theta}_e)$ from $\mathbf{b}_i$ and its learned parameters $\boldsymbol{\theta}_e$. The class activation $\mathbf{y}'_i$ can be predicted by a label classifier with $G_y(\mathbf{x}_i;\boldsymbol{\theta}_y)$. Adversarial DA approaches often use an additional domain classifier with $G_d(\mathbf{x}_i;\boldsymbol{\theta}_d)$ to predict the domain activation $\mathbf{d}'_i$. It is connected via a gradient reversal layer (GRL) to remove domain discriminative features embedded in $\boldsymbol{\theta}_e$ during optimization.

factors on which the combinations are based are domain factors, then they will be removed from the feature representation by DA already during training. Thus, there will be no influence of the new combinations in the test set.

A general negative side effect of DA removing domain factors occurs when the removed factors are not only domain-informative, but also task-informative or visually close to other task-informative factors. For example, if the hand, which holds the target object, is removed from the feature representation, the factor *'hand posture'* could also be removed, which, as mentioned before, can be task-informative as well. Additionally, in case of visual closeness this might remove parts of the target object due to network architectural reasons related to receptive field sizes.

3. Related work

In the first part of this section we will analyze different image datasets in relation to the concept of the introduced visual factors. The second part will review different related single and multi-source DA approaches.

3.1. Image datasets in context of visual factors

There are many datasets for image categorization, that are based on internet search, like [10,11]. This provides a large variation for the different categories in terms of objects and their context. However, these datasets also come with strong biases, like a western world bias [12,

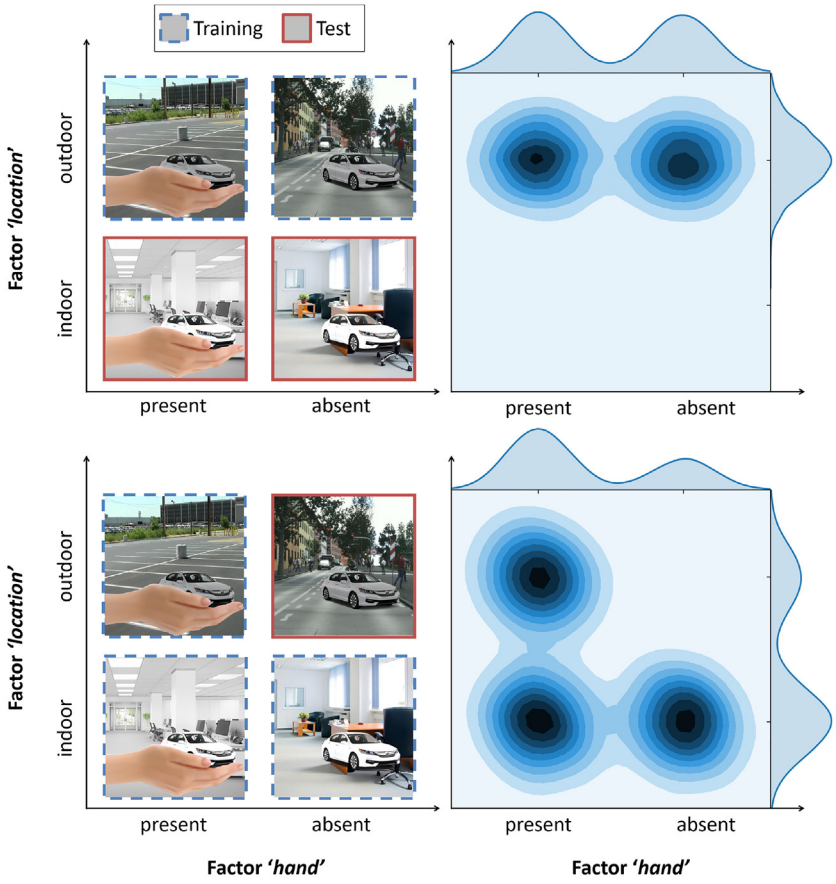

Fig. 4. Constellations of factor values in training and test set. *Top:* New value for factor *'location'* in the test set. *Bottom*: New combination of factor values in the test set. All factor values are already included in the training set.

13] or a capture bias [1], and do not allow control over many factor values, like the object type or its viewing point.

In contrast, datasets used for DA research [14–17] usually come with increased control over specific factor values. There are datasets like [18–21] where some domains are based on restricted internet searches, as for example only images from Amazon or clipart type. However, the control over object instances is still limited. Increased control over objects and background is usually found in the robotics community, where the same objects are placed in front of different backgrounds, like in CORe50 [4], but also in the well-known Office-31 DA dataset [19] where two of the three domains were generated similarly. If the segmentation of the object is known, like in MNIST [22], the background can virtually be changed as in MNIST-M [3] to obtain a new domain. Full control over all factor values is usually found in rendered datasets, like the disentanglement library [23], which shows simple objects in a 3D environment with changing values for factors like size, viewing angle, color and others.

3.2. Recent approaches in context of visual factors

Recent works in DA are mostly based on either kernel-based methods or adversarial methods. Kernel-based methods, like maximum mean discrepancy [24], try to align feature distributions of the involved domains by using a cost function on a shared feature extractor that should minimize the distance between distributions. This is applied for example in [25–30]. Approaches using adversarial methods [31,32] are mostly based on a regressor or a full domain classifier that is additionally attached to the shared feature extractor. The overall target here is that the feature activations of the shared feature extractor should not rely on domain-informative factors. This is often achieved by flipping the gradient from the domain classifier at the connection to the shared feature extractor via a gradient reversal layer [3,29,33,34]. Other approaches, like [35], optimize a first feature extractor supervisedly on the source data and after this train a second feature extractor on the unsupervised target data, in a way that a classifier cannot determine the domain from their outputs. This is similar to the workflow of generative adversarial networks [36].

Newly proposed approaches are usually only tested on data for which a gain in generalization performance can be reported. The effect of negative transfer [6,37] is only rarely addressed, as e.g. in [38–41]. Research showed that it can occur when the label spaces of source and target domain differ [40], or the distributions are imbalanced [41].

Another reason for negative transfer can be latent domains or underlying multimodal structures within pre-defined domains. Latent domains have already been studied in [42–44]. In [45] they use detected latent domains within a single source domain to enhance DA by aligning each latent domain individually in relation to the target domain. Instead of splitting a domain into multiple sub-domains we target to describe multi-domain data by their superposing factors. Approaches as [46–50] criticize that global alignment of domain distributions is generally not sufficient since for instance mismapping of classes within aligned distributions might still persist. In [51] they therefore use individual domain classifiers for each class to reduce negative transfer caused by this underlying mismatch. In [52] they additionally combine global distribution alignment and local alignment in a hierarchical manner. Local alignment on class-level is also applied in [53], where they align class specific clusters of source and target domain in feature space. The alignment is enforced through an additional cluster adaptation loss that is based on pseudo labels generated during training for the unsupervised target domain samples. In [54] they add additionally to a domain classifier a sliced Wasserstein distance measure on the softmax output of the label classifier with the target to increase the discriminability of classes also for the target domain. Unlike adversarial approaches solely based on a domain classifier, here the unsupervised samples are also forwarded to the label classifier. In contrast to the mentioned approaches, ours considers the effect of class independent factors in multi-domain data, which is a complementary direction.

In [55] they consider the case where the difference between a source and target domain can be described by an accumulation of multiple intermediate domains. Their approach is to adapt the model step-wise to the target domain, by generating pseudo labels and fine-tuning the model on the intermediate domains until the target domain is reached. In terms of factors they consider a single continuous domain factor that changes its value smoothly between the adjacent domains. Our approach in contrast considers the more general case of multiple heterogeneous factors in a larger multi-domain setting, while further no intermediate domains are assumed to be given.

We see a general problem in the removal of all domain factors, which tends to remove task-informative factors as well. Having multiple source domains usually leads to the removal of more domain factors and therefore a pronounced negative effect can be expected. The negative effect of irrelevant source domains was reported in [41,56]. Our FP-DA is designed to reduce negative transfer by switching off competition between groups of domains. There are other methods that try to limit or better control the competition between domains. This is mostly done by a weighting strategy that is a based on a score that rates the similarity of domains, either pair-wise, i.e. only competition between the target domain and a single source domain, as in [57–62] or as a weighted competition also between multiple source domains, as in [21,63–65]. However, none of the mentioned approaches consider factor or group based competitions as we do.

4. Our approach

The main idea of our FP-DA is to preserve a chosen factor f_c during DA. It is based on the adversarial DA architecture from [3]. The architecture corresponds to the typical adversarial one shown in Fig. 3, where a domain classifier is used together with a gradient reversal layer to inhibit learning of domain-informative features in the feature extractor. The parameters of the feature extractor $\boldsymbol{\theta}_e$, the label classifier $\boldsymbol{\theta}_y$, and of the domain classifier $\boldsymbol{\theta}_d$ are updated in the following way:

$$\boldsymbol{\theta}_e \leftarrow \boldsymbol{\theta}_e - \mu\left(\frac{\partial L_y}{\partial \boldsymbol{\theta}_e} - \lambda \frac{\partial L_d}{\partial \boldsymbol{\theta}_e}\right) \tag{1}$$

$$\boldsymbol{\theta}_y \leftarrow \boldsymbol{\theta}_y - \mu \frac{\partial L_y}{\partial \boldsymbol{\theta}_y} \tag{2}$$

$$\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d - \mu \frac{\partial L_d}{\partial \boldsymbol{\theta}_d} \tag{3}$$

with L being the corresponding loss for each branch of the architecture and μ the learning-rate. The influence of the domain classifier on the feature extraction path is controlled by the adaptation factor λ which is smoothly increased during optimization. For further details on the general training procedure we refer to [3]. Originally the domain classifier was used to discriminate data from a single source and target domain, however, it can also be used for multiple source domains and in this case also without any data from the target domain. In each case, all domains compete against each other in the domain classifier.

To preserve a given factor, in FP-DA we switch off competition between domains that have different values for a chosen factor f_c. For this we use the decomposition of the gradient

$$\frac{\partial L_d}{\partial \boldsymbol{\theta}_d} = \frac{\partial L_d}{\partial \mathbf{d}'} \cdot \frac{\partial \mathbf{d}'}{\partial \boldsymbol{\theta}_d}, \tag{4}$$

and replace $\frac{\partial L_d}{\partial \mathbf{d}'}$ by

$$\frac{\partial \widetilde{L}_d}{\partial \mathbf{d}'} = \mathbf{z} \odot \left(\frac{\partial L_d}{\partial \mathbf{d}'} \right), \tag{5}$$

where $\mathbf{z}$ is an m-dimensional vector corresponding to the m domains. If an element $z_j = 1$ then the gradient of the domain neuron d_j is kept and propagated to the parameters $\boldsymbol{\theta}_d$. If $z_j = 0$ then no gradient is propagated from the neuron d_j. For a given training sample of domain k, in FP-DA we compute z_j like

$$z_j = \begin{cases} 1 & \text{if } f_c(j) = f_c(k) \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

to only allow competition between domains that have the same value for f_c as the current training sample. Generally, given a domain label d_i for a sample i, the DA approach of [3] generates a negative teaching signal for all other domains in the domain classifier, while FP-DA removes some of this competition in a controlled way by just generating a negative teaching signal for the domains within the corresponding group of the training sample i. Note, for FP-DA we assume f_c to be a categorical variable. The principle of FP-DA is depicted in Fig. 5. A limitation of FP-DA is that within each group of domains that share the same value for f_c, all other domain factors from the dataset must still be represented to learn a general model.

5. Experiments

In this section we will evaluate FP-DA at the example of an object classification task in a multi-source domain scenario. For this we will first naively apply the standard DA method in leave-one-out experiments, where we choose one domain as target domain and the rest as source domains, to investigate the effects of removing all domain factors. This is followed by one-to-one experiments, i.e. single source transfer, where we investigate the transfer performance in context of visual factors. Based on this we will show how factors responsible for high transfer errors can be identified and used for the grouping that is required in FP-DA. Finally we will show how our method is able to improve the average and minimum performance in such a multi-source domain scenario. Note, besides the leave-one-out and one-to-one experiments further constellations in between are possible, however, we decided for the mentioned, since those are the most meaningful cornercases and also tractable to do.

5.1. Experimental setting

Dataset. For our first experiments we chose the CORe50 dataset introduced in [4]. The dataset consists of 11 distinct recording sessions, where 50 objects, divided into 10 categories with 5 objects each, were captured while held and rotated in a human hand. From each session a representative image of three objects is shown in Fig. 6.

Originally this dataset was meant to serve as a benchmark for continuous learning, where class instances from new sessions were added over time and the task was to classify the instance or the category. Nevertheless, we see this dataset with its controlled and clean recording setting also applicable for domain transfer investigations. Therefore we will use each session as a domain and choose category prediction as our main classification task for all experiments. To our knowledge we are the first to use it in the domain transfer context. It can be described mainly by the photographer view since it is based on natural images, however, not fully, since the objects were artificially placed in the environment. The domain factors here are mostly related to the recording location. Since the objects are in each domain the same, they can be described as a constant factor across domains. Clearly each domain was

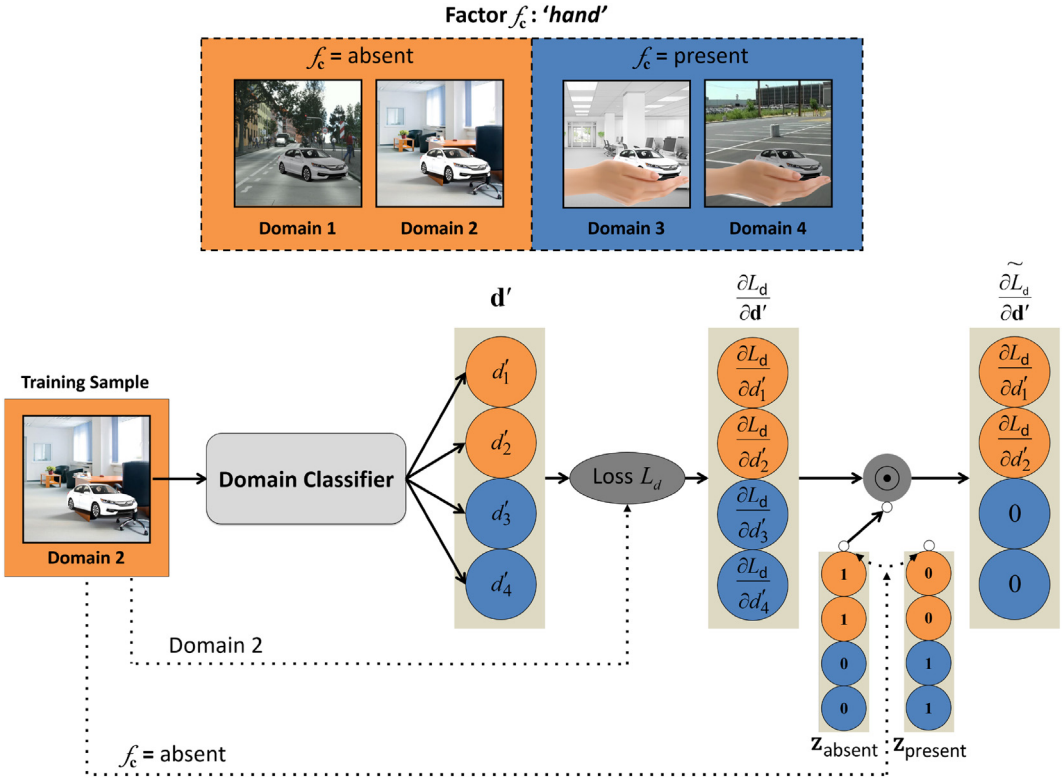

Fig. 5. Our FP-DA training method uses given groups of domains, here generated by the values of the chosen factor f_c, *'hand'*, and allows competition only between domains within each group. A training sample from the group where f_c = absent is forwarded through the domain classifier. The resulting gradient vector $\frac{\partial L_d}{\partial \mathbf{d}'}$ is element-wise multiplied by z_{absent} that zeros the gradients of the domains where f_c = present.

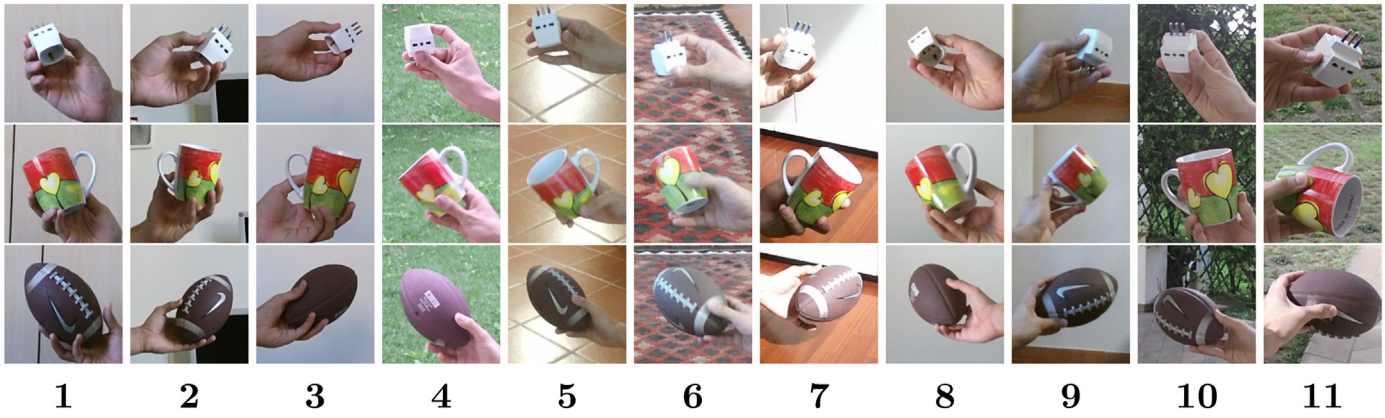

Fig. 6. Exemplary images from the 11 different sessions of the CORe50 dataset. The background of each session is representative for each of the 50 objects. In each session a single hand, i.e. right or left, was used exclusively for presentation. We use each session as an individual domain.

exclusively captured holding the object with either the right or the left hand. However, it remains unclear whether always the same person was presenting the object, if not, this could describe another domain factor that is related to individual styles of presenting and holding the objects.

Since there is only a single recording stream per domain, no clear training/test separation is possible. This would be in general no problem since most of our results focus on generalization to a completely unknown target domain. However, there are some results where unsupervised data of the target domain is used during training. Therefore we chose from each session the half of the 300 images per object as fixed test data, by dividing the recording stream into chunks of 20 images. Note, when training on domain A and testing on domain B, independent of the application of DA, we use all images of A, but only 50% of B, since the other 50% are reserved as unsupervised data for DA.

Implementation details. For the baseline experiments without DA we used the standard VGG-16 architecture from [66]. The weights are initialized by a publicly available set of parameters that was pre-trained on the ImageNet database. For DA experiments we attached additionally the domain classifier after the last convolutional layer of this architecture. It consists of two fully connected layers with 1024 neurons each, similar to [3], and a final output layer with m neurons, corresponding to the number of domains. For both losses, L_y and L_d, we chose the cross-entropy loss function. The adaptation factor λ is smoothly increased during optimization from 0 to 1 with the same update rule as presented in [3]. We chose a batch-size of 64 and an initial learning-rate of $\mu = 0.0001$, which was decreased over the number of steps. On both classifiers we used dropout in the fully-connected layers as the only regularization method and applied softmax on the final output. The input samples are mean normalized, but not modified by any data augmentation method. The architecture was implemented in TensorFlow. For all reported numbers we trained the specific architecture 10 times with randomly initialized parameters $\boldsymbol{\theta}_d$ of the domain classifier and randomly shuffled training data, for two epochs and averaged the classification accuracy.

5.2. Standard multi-source transfer

For the given dataset we first do leave-one-out experiments to find out how well many source domains can generalize to a single target domain. This we do without and with DA. For the latter case we only use the source domain data or we further provide unsupervised data of the target domain. Even without DA a good performance can be expected, as most domains are similar to at least one other domain. DA without unsupervised target data should then improve generalization for unseen combinations of factor values, while using target data should further help in the case of new factor values.

Looking at the results without DA in Fig. 7, we see that the expectations are widely met. The classification accuracy on the target domains reaches mostly more than 90%. The only target domains that show a comparatively poor performance are domains 6 and 10. Looking at their visual characteristics, we can see that they differ significantly from the others by having both strong fine-grained edges in the background. In domain 6 caused by a carpet and in 10 caused by a fence. Interpreting these aspects as factor values, it is unclear whether these are new values or new combinations of known values.

The results for DA without unsupervised data show for most of the target domains a slight drop in performance indicating negative transfer. This is can be on the one hand side due to the additional constraint on the feature extractor by the gradient from the domain classifier, but also due to task-informative factors that are removed now. On domain 6 and 10 the accuracy has also only slightly changed, which suggests that the issue here are mainly completely new factor values and not combinations.

Looking at the results of DA with unsupervised data, the assumption about new factor values in domain 6 and 10 can be approved with a significant increase in accuracy. However, for other experiments, like for example on domain 1, 4, and 8, we mostly see a weaker performance compared to DA without unsupervised data. We believe that because of the increased number of domains during training, even more factors, including task-informative factors, are now removed from the feature representation. Compared to experiments without DA, there is a clear performance drop on seven target domains. We explain this also by the additional constraint and removed task-informative factors.

When comparing the results without DA to DA with unsupervised data in total, the negative transfer for some domains balances out the strong gain for domains 6 and 10 on average, while the latter leads to a strongly improved minimal performance. We think that the negative transfer can be reduced by preserving certain factors. These factors are based on mutual similarity and difference between individual domains and thus can not easily be inferred from the multi-source setting here.

5.3. Single-source transfer without domain adaptation

To determine error causing factors in DA we conducted one-to-one transfer experiments without DA first, i.e. optimized the VGG-16 architecture on each individual domain and evaluated it on all others. The results are given in Table 1. For each target domain we also give the maximal performance among all source domains. These maximal values are all smaller than their multi-source counterpart in Fig. 7. This means

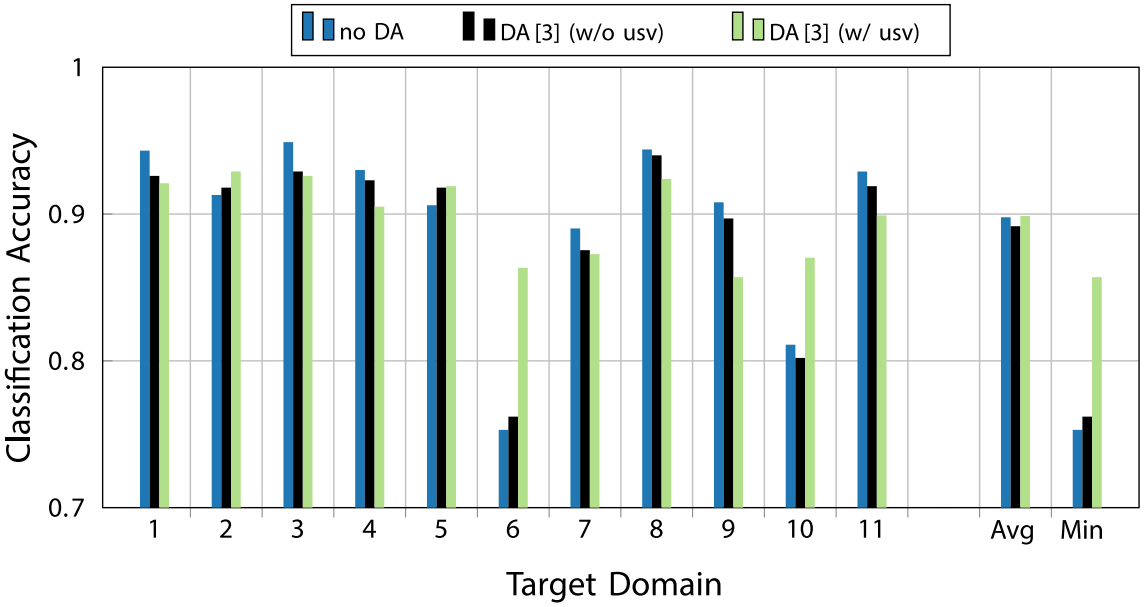

Fig. 7. Multi-source leave-one-out experiments. The left out domain is the target domain, while the others are used as the supervised source domains. The experiments were carried out with and without standard DA, while the latter was applied with and without additional unsupervised (usv) samples from the target domain. Negative transfer can be seen on most easy domains, like 1, 3, and 8, while positive transfer is pronounced among the difficult domains 6 and 10. Both effects cancel out on average but lead to a strong gain in minimum performance for DA w/ usv.

that multi-source training really builds a more general model instead of just memorizing all domains individually.

All investigated transfers are in line with our previous multi-source experiment findings. The weak performance on domain 6 and 10 as target domain are reflected here as well, with lowest average classification accuracy. On the other hand, when training on 10, the performance on the target domains shows the best average and minimum performance, which indicates that the domain covers already a large variety in background related factor values. The highest max performance and also highest average performance is achieved on target domain 1. At first sight this might mostly be related to the simple white background. However, as we will show in the upcoming investigation of error causing factors, this is not the only reason.

The overall matrix shows reoccurring patterns of weak performance. This is especially pronounced at the transfers from domain 4, 5, 6, and 8 to the domains 2, 3, 7, 9, 11. A closer examination of the these domains reveals that the objects presented there are exclusively held in either the right or the left hand. A transfer with a change in hands clearly leads to a performance drop in comparison to a transfer between the same hands. Besides the hand there might be other less prominent error factors. These we like to reveal in the following.

5.4. Identification of error causing factors with PCA

In this section we try to determine error factors in a systematic way. For this we apply PCA on the one-to-one results of Table 1, where each

Table 1
One-to-one transfer experiments with DA. The difference in percentage points to Table 2 is shown. *Min* and *Max* refer to absolute accuracies. Left-hand domains are marked in blue.

Source Domain \ Target Domain	1	2	3	4	5	6	7	8	9	10	11	*Avg*	*Min*
1	-	0.67	0.72	0.74	0.68	0.45	0.64	0.80	0.57	0.48	0.52	0.63	0.45
2	0.69	-	0.80	0.59	0.54	0.55	0.72	0.60	0.76	0.52	0.69	0.65	0.54
3	0.69	0.78	-	0.52	0.51	0.36	0.72	0.60	0.74	0.40	0.66	0.60	0.36
4	0.70	0.46	0.50	-	0.68	0.70	0.51	0.72	0.36	0.68	0.52	0.58	0.36
5	0.67	0.42	0.46	0.68	-	0.62	0.40	0.72	0.37	0.54	0.34	0.52	0.34
6	0.68	0.46	0.44	0.72	0.68	-	0.47	0.67	0.38	0.54	0.49	0.55	0.38
7	0.70	0.75	0.75	0.58	0.52	0.47	-	0.65	0.66	0.47	0.76	0.63	0.47
8	0.83	0.59	0.57	0.76	0.71	0.59	0.57	-	0.48	0.52	0.50	0.61	0.48
9	0.52	0.66	0.74	0.42	0.37	0.30	0.57	0.43	-	0.40	0.66	0.51	0.30
10	0.75	0.69	0.63	0.81	0.75	0.69	0.59	0.74	0.60	-	0.64	0.69	**0.59**
11	0.57	0.66	0.79	0.48	0.42	0.40	0.74	0.53	0.66	0.48	-	0.57	0.40
Avg	0.68	0.61	0.64	0.63	0.59	0.51	0.59	0.64	0.56	0.50	0.58		
Max	**0.83**	0.78	0.80	0.81	0.75	0.70	0.74	0.80	0.76	0.68	0.76		

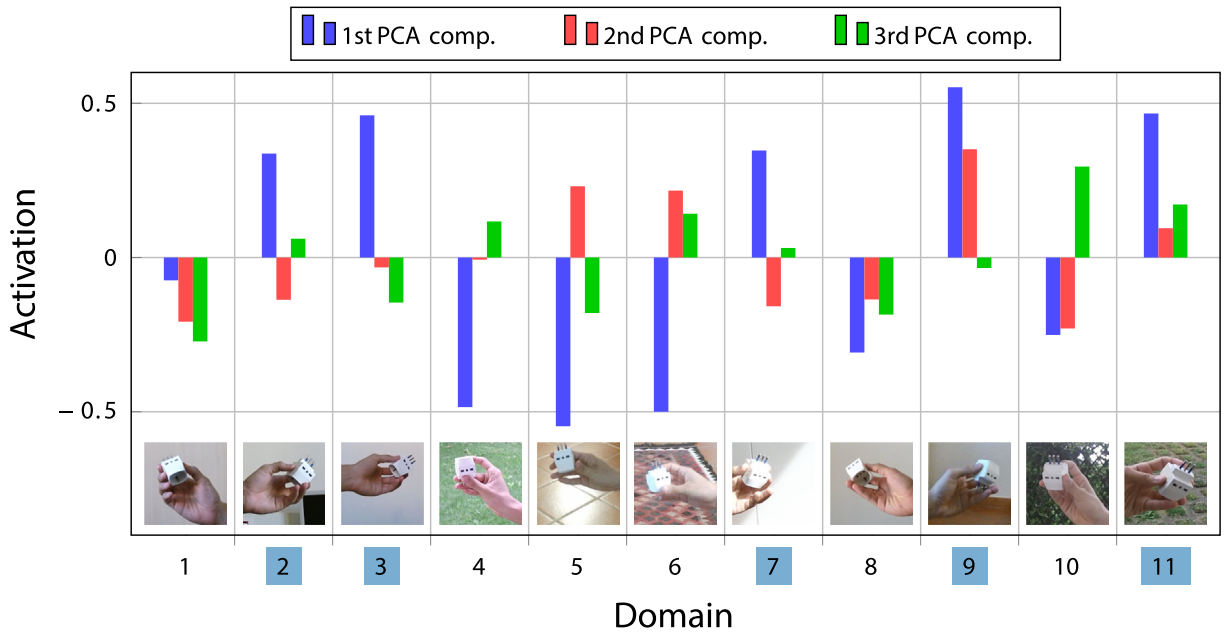

Fig. 8. PCA analysis of Table 1: The sign of the 1st component activation is in line with the hand that was used. Positive values refer to left hand domains (marked blue), negative values to right hand domains. Note the low absolute value for domain 1 which indicates the hand showing the object in a neutral position from the bottom. Positive values in the 2nd component could be related to domains with low contrast, while negative ones to high contrast. The 3rd component might indicate highly textured background by high positive values.

row is treated as a sample with 11 dimensions, and the gaps for transfers between the same domain are filled with 1.0. The total variance in the accuracy is 0.26, where the first component describes already 60% of it. The activation of the first component in Fig. 8 shows a clear grouping in domains with high positive and high negative values. These two groups perfectly represent the used hand. Additionally, the absolute values of the activation seem to carry information how clear on one side of the target object the hand is positioned within the image. Domain 1 describes here a special case with an activation close to zero, indicating that the hand is visible mostly at a neutral position below the target object.

The second component explains 12% of the variance and shows a rather continuous distribution of activation in Fig. 8. We think that is related to light source factors, resulting in different average contrasts for each domain. Positive values indicate here a comparable low contrast of the objects and the holding hand to the background, which is pronounced at domain 5, 6, 9 and partly 11.

The third component represents 10% of the variance. The activation in Fig. 8 is rather continuous again and might correlate with the strength of background texture, having the highest values for domains 6, 10 and 11.

The activation of PCA components gives hints about possible domain factors. Instead of our manual interpretation, it would also be possible to correlate the PCA activation with image metadata like, time of the day, the time of the year or the GPS location of the recording. However, interpretability is not necessarily given, since the PCA suggested factorization could also be a mix of multiple human interpretable factors. Besides PCA also other factorization methods like independent component analysis could have been applied, possibly leading to less clear results due to the missing orthogonality of the components.

Note, since starting from a pre-defined domain separation on which the transfer matrix is based, PCA only allows to detect error causing factors that are consistent throughout the pre-defined domains. In other words, factors that vary within a pre-defined domain, e.g. on category level, can not be identified here.

5.5. Single-source transfer with domain adaptation

For the transfer from one source domain to one target domain with DA, we need to additionally provide unsupervised data of the target domain. Adversarial DA will try to remove domain factors between these two domains. Before, we identified the hand as one such factor and also showed that a changing hand is the main cause of classification error. Consequently, the removal of that factor should lead to an increased performance.

However, when looking at the results in Table 2, usually a further decrease in performance is visible for changing hands. This is especially the case for the transfer from domain 3 to 5 with a decrease of 17 percentage points and 7 to 8 with 16 percentage points. An increase is mainly observed on transfers where the hand does not change between domains. The biggest improvements can be found in the transfers from domain 1 to 10 and 1 to 6 with an increase of 22 and 21 percentage points. We think that this significant increase is also caused by the same effect as in the multi-source experiments, where the unsupervised data of 6 and 10 has provided the biggest benefit by introducing the new factor values related to the background. The average increase in accuracy for transfers where the hand does not change is at 5.7 percentage points, while the average decrease for changing hands is −3.8 percentage points.

The highest minimum performance has dropped significantly from former 59% accuracy on source domain 10 to now 49% on domain 1, indicating the strong negative transfer caused by DA.

Our results clearly indicate strong negative transfer, caused by removing the factor *'hand'* through DA. The reason for this is on one side that the hand is simultaneously a task-informative factor, by its posture giving clues about the object category. On the other side it has visual closeness to the target object, which means removing it can harm classification significantly due to fixed receptive field sizes in the architecture.

5.6. FP-DA for multi-source domain adaptation

In this section we like to investigate how FP-DA can reduce negative transfer by preserving a given factor. The best candidate for

Table 2
One-to-one transfer experiments without DA. The results show the classification accuracy. Good performance is indicated by bright green color, while bad performance by bright red color. Left-hand domains are marked in blue.

Target Domain (w/ usv data for training)

Source Domain	1	2	3	4	5	6	7	8	9	10	11	*Avg*	*Min*
1	-	0.08	-0.06	0.04	0.07	0.21	-0.01	0.04	-0.07	0.22	0.09	0.06	**0.49**
2	0.04	-	0.02	-0.02	-0.02	-0.07	0.04	-0.02	-0.04	0.05	0.03	0.00	0.48
3	-0.03	0.02	-	-0.11	-0.17	-0.02	0.02	-0.10	0.01	0.06	0.07	-0.02	0.33
4	0.05	0.00	-0.05	-	0.12	0.06	-0.05	0.07	-0.08	0.05	-0.03	0.01	0.28
5	0.04	0.06	-0.05	0.02	-	0.09	-0.03	0.05	-0.05	0.12	0.00	0.02	0.32
6	0.08	0.06	-0.03	0.10	0.14	-	-0.01	0.11	-0.08	0.17	-0.02	0.05	0.31
7	-0.09	0.04	0.02	-0.09	-0.09	0.02	-	-0.16	-0.01	0.03	0.02	-0.03	0.43
8	0.00	-0.04	-0.08	0.03	0.09	0.08	-0.07	-	-0.08	0.17	-0.04	0.01	0.41
9	0.05	0.10	0.07	-0.08	-0.05	-0.02	0.11	-0.02	-	-0.02	0.11	0.03	0.28
10	0.03	-0.05	-0.13	0.01	0.00	0.01	-0.07	0.01	-0.15	-	-0.05	-0.04	0.45
11	-0.04	0.01	-0.01	0.00	-0.09	-0.03	0.00	-0.11	0.07	-0.02	-	-0.02	0.34
Avg	0.01	0.03	-0.03	-0.01	0.00	0.03	-0.01	-0.01	-0.05	0.08	0.02		
Max	0.83	0.80	0.82	0.83	0.83	0.76	0.75	**0.84**	0.75	0.74	0.78		

such a factor is the hand, which also corresponds to the first PCA component. Similarly, we will also evaluate the groupings defined by the sign of the activation of the second and third PCA eigenvector. As FP-DA switches off competition between certain pairs of domains, in general it leads to a decreased influence of the domain classification path. To investigate this effect, we also tested several random factors, each having 2 groups with 5 respectively 6 domains. The results, together with the previous multi-source experiments, can be found in Fig. 9.

Looking at the performance when defining the groups based on the 1st PCA component, i.e. based on the hand used for presenting, the overall highest average and highest minimum accuracy can be reached. The increase in comparison to DA [3] with unsupervised data is at 1.9 and 1.1 percentage points, respectively. The detailed results for each domain reveal that FP-DA manages mostly to reduce the negative transfer that was caused by standard DA, in certain cases an even better classification accuracy than without DA could be achieved. This applies for example to domain 4 and 7. Using the 2nd or the 3rd PCA component provides only a reduced gain of average performance of about 1.2 percentage points. We see a similar gain if we completely omit the PCA step and average over random groupings instead. This indicates that in general a reduced competition and thus influence of the domain path is beneficial here, but not the removal of 2nd and 3rd PCA component in particular.

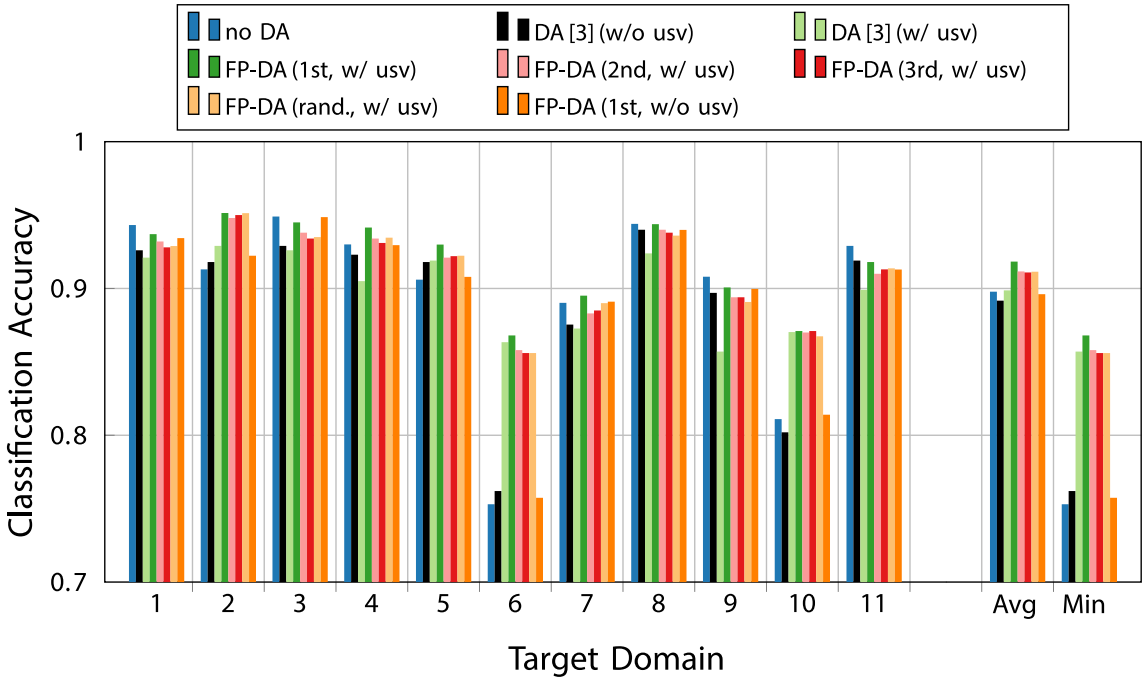

Fig. 9. Multi-source DA with our proposed FP-DA. 1st, 2nd, and 3rd describe the grouping based on the three PCA components, rand. a random permutation. FP-DA with the domain grouping based on the used hand (1st) and usv data shows the highest average and highest minimum performance among all constellations.

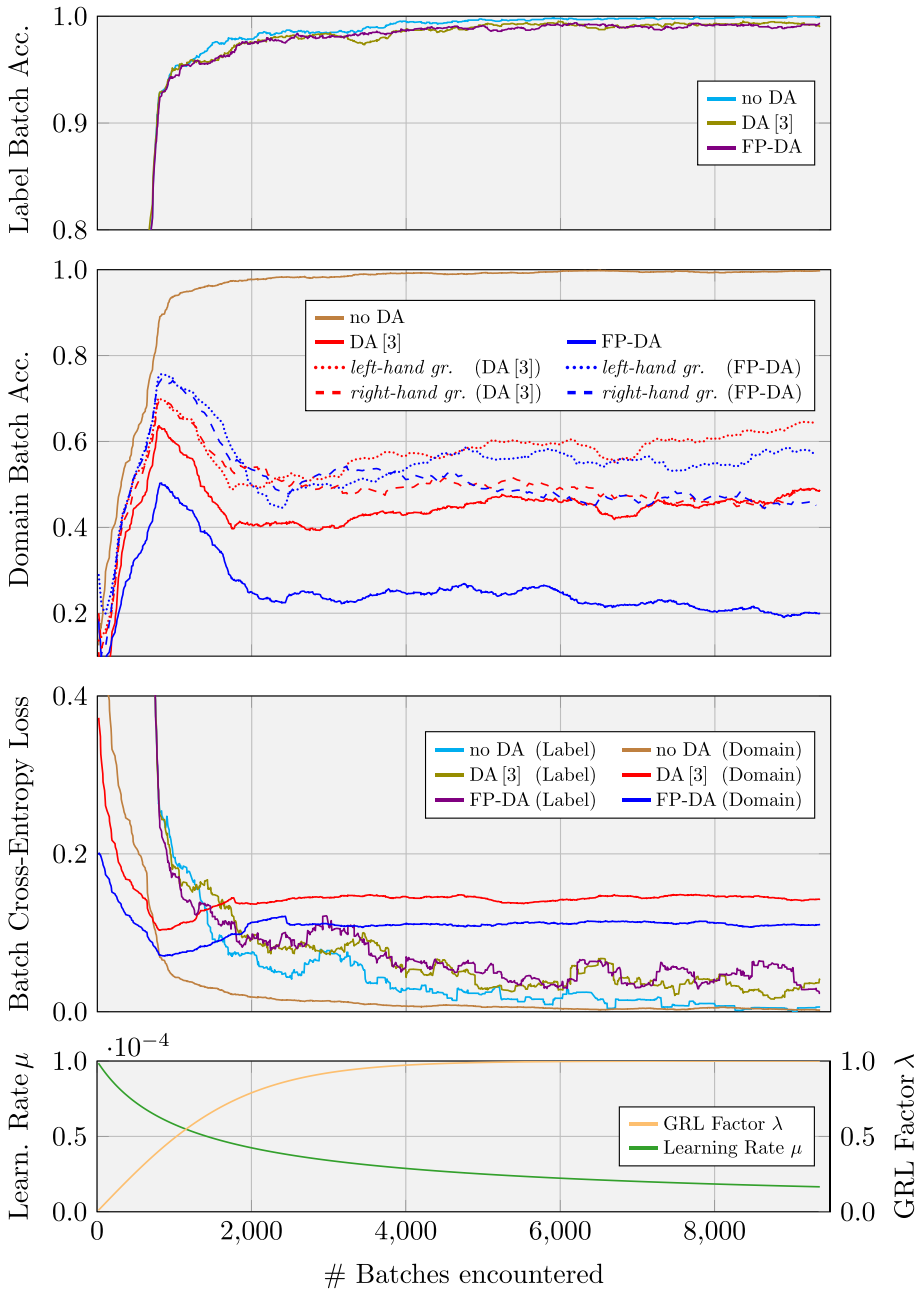

Fig. 10. Exemplary training curves for the leave-one-domain-out experiments where domain 9 was the target domain. The plots show two epochs with a moving average applied to the original curves. The first and the second plot show the accuracy of the label classifier and the domain classifier, respectively, with the latter additionally including a split up into group-wise domain accuracy based on the used hand. The third plot shows the losses of all classifiers, and the fourth plot the hyperparameters λ and μ. Note, without DA, $\lambda = 0$.

Another strategy to deal with the changing hand would be to augment the data by horizontal flipping. However, this is not a method to deal with domain factors in general and was therefore not considered here.

Training convergence. Fig. 10 shows the training curves of a single leave-one-out experiment with domain 9 as the target domain. The plot compares the curves for the three cases, without DA, with DA [3], and with FP-DA (hand grouping), based on the output of the label classifier and the domain classifier. Independent of the case, for the label classifier all accuracy curves converge towards 1. With DA [3] and FP-DA the curves are slightly more noisy due to the concurrent force of the domain classifier. As expected, the accuracy of the domain classifier without DA, converges towards 1. Note, in this case $\lambda = 0$, meaning the domain classifier does not influence the actual learning process. For DA [3] and FP-DA the accuracy of the domain classifier rises strongly in the beginning. However, at a certain point, with increasing λ, the GRL removes more and more domain specific information from the shared features. Thus the overall accuracy goes down again and converges at a much lower level compared to the accuracy of the label classifier. For FP-DA the level of the turning point and the level of convergence are much lower than for DA [3]. For DA [3] the reached accuracy here is at 45%, while for FP-DA the accuracy is at only 19%. We further split the overall

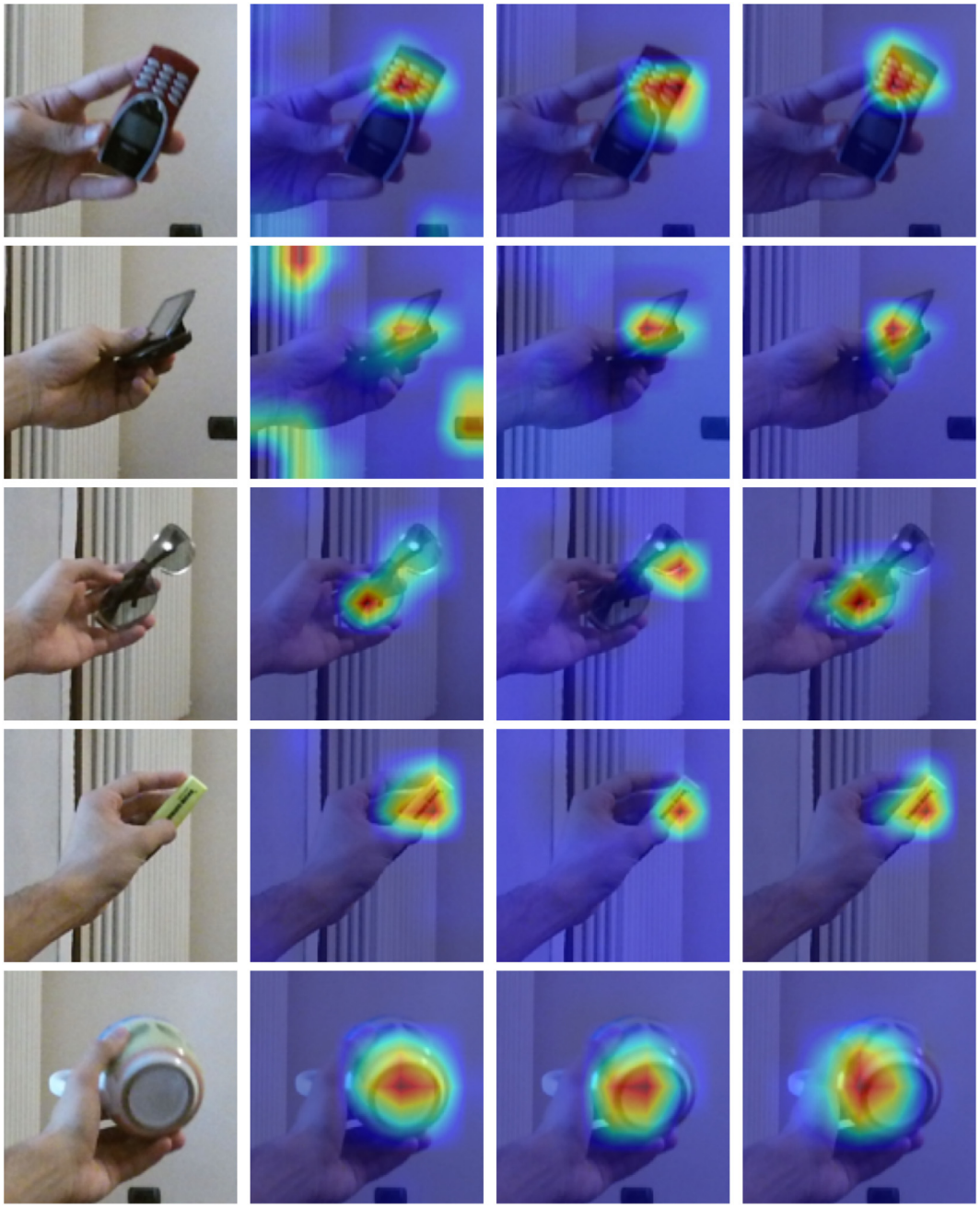

Fig. 11. Grad-CAM visualization for selected images of target domain 9. DA [3] removes background activation (especially row 2), but also shrinks the activation on the object (especially row 4) and shifts it away from the hand (especially row 3). With FP-DA, this area is less shrunk and activations closer to the hand can be observed.

accuracy into the two group-wise accuracies (left-hand, right-hand). In these groups the accuracies of DA [3] and FP-DA are very similar. This shows that FP-DA did not learn to separate domains between the groups initially and consequently the GRL can not destroy this information in the features later. We see a similar temporal interplay for the loss, however, the loss of FP-DA is constantly smaller, because of the group-wise training signal.

Training and inference time of FP-DA. The training of the models was performed on a NVIDIA GeForce RTX 2080 Ti with 11GB of memory. A single training step of our own implementations, composed of forward pass and weight update, takes approximately 141 ms without DA and 144 ms for DA [3] and FP-DA, respectively. Invariant of the used method the inference time for a single test image is at approximately 62 ms.

Grad-CAM visualizations. To get visual evidence for the preservation of the factor *'hand'* by applying FP-DA and grouping the domains by the used hand, we generated class activation maps with Grad-CAM [67]. Grad-CAM highlights an image region that is decisive for a target class by using the gradient of that target class to sum the feature activation maps of the last convolutional layer in a weighted fashion. In Fig. 11 this visualization is shown for selected images for FP-DA, DA [3], and without DA. It shows that without DA, focus on the background can be

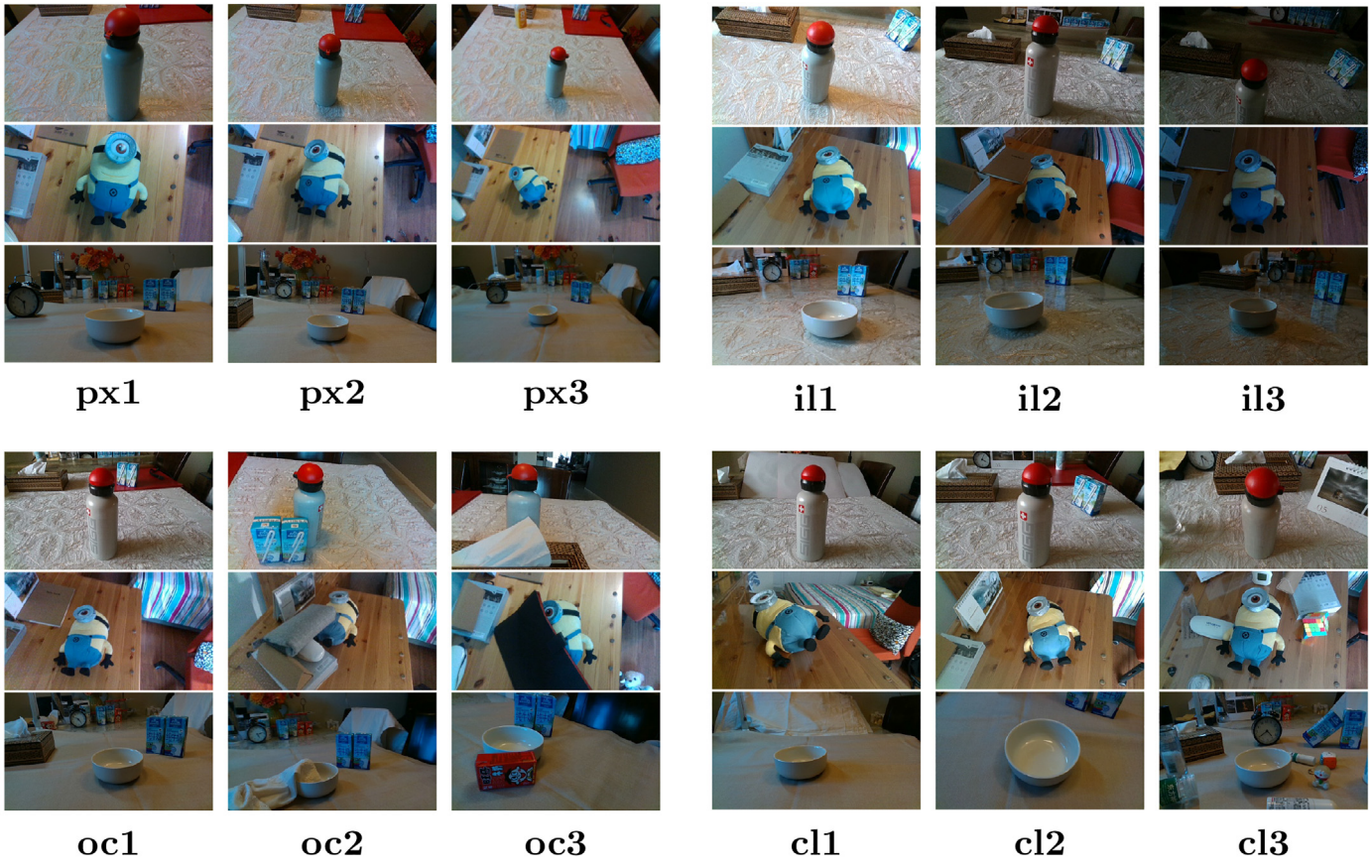

Fig. 12. Example images for the 12 domains of the OpenLORIS object dataset. In each domain one of the factors *'object pixelsize'* (px), *'illumination'* (il), *'occlusion'* (oc), and *'clutter'* (cl) has a fixed value, according to one of three levels of difficulty. Level 1 is the easiest, while level 3 is the most difficult level.

observed, while with DA this is removed and the highlighted area on the object is clearly shrunk. Further, it is moved away from the holding hand. With FP-DA the area is less shrunk and more activation on the hand can be observed, indicating that more information about the *'hand'* and thus also about the object is preserved.

5.7. FP-DA on OpenLORIS dataset

In the OpenLoris object dataset [5] 69 objects were presented in 12 domains (see Fig. 12). In each domain one of the four factors *'object pixelsize'* (px), *'illumination'* (il), *'occlusion'* (oc), and *'clutter'* (cl) was set to one of three difficulty values {level1, level2, level3}, while the other 3 factors are undefined. Thus, the dataset contains some factors by design, but FP-DA might be strongly limited in preserving those factors as their combinations are not present in the data. For the one-to-one experiments, the first PCA component (see Fig. 13), separates the difficult level3 domains plus oc2 from the rest, while the second component slighlty suggests the groups {px, oc} and {il, cl}. Similar to CORe50, in the leave-one-out experiments we see a strong positive effect of DA [3] for the difficult domains, while there is some slight negative transfer among the easier ones. FP-DA only leads to a minimal increase of the average performance and even a decrease in minimal performance. We think that a reason for this are the missing combinations of factors in the data. To reduce the overall complexity of the domain setting, we further repeated the same test with all pairs of factors i.e. we had 6 domains per experiment instead of 12. These tests showed very similar tendencies as the full domain setting. Especially, no significant gain of FP-DA could be observed.

Finally, to check if FP-DA would profit from a combination of factors in the data, we manually overlayed an artificial factor *'brightness'* with the values {normal, dark} over the data. For the value 'normal' we take a domain's original data, while for 'dark' we multiplied the value channel of all images of a domain in the HSV space by 0.3 and added this data as a new domain. Overall now we have two labeled factors per image. We investigated this new factor individually in combination with either px, oc, or cl (see example images in Fig. 14). The combination with il was omitted, since *'brightness'* and *'illumination'* are of similar nature.

For the one-to-one experiments, the first PCA component consistently separates the domains by brightness into 'dark' and 'normal', while the second component groups the level3 domains from the rest (see Fig. 15). In the leave-one-out setting, DA [3] has mostly a slight negative effect on average and minimum performance. For FP-DA, we see a further substantial decrease when using the first PCA component. However, we have a strong gain for the second component, yielding the overall best performance.

Grouping the domains by the first component preserves the factor *'brightness'* and removes the other factor, i.e. depending on the experiment *'object pixelsize'*, *'occlusion'*, or *'clutter'*. When looking at the images it becomes clear that the removed factors can be task-informative. Most target objects come with specific context objects that are unique to the target object. Removing them as a feature consequently complicates the classification task. Note, although in px3 the background variation should not change too much, we observe that more task-informative background noise is visible caused by the greater distance of the camera to the target object. The grouping based on the second PCA component, removes the factor *'brightness'* and preserves most of the variations in the second factor, leading to a clear superiority of FP-DA.

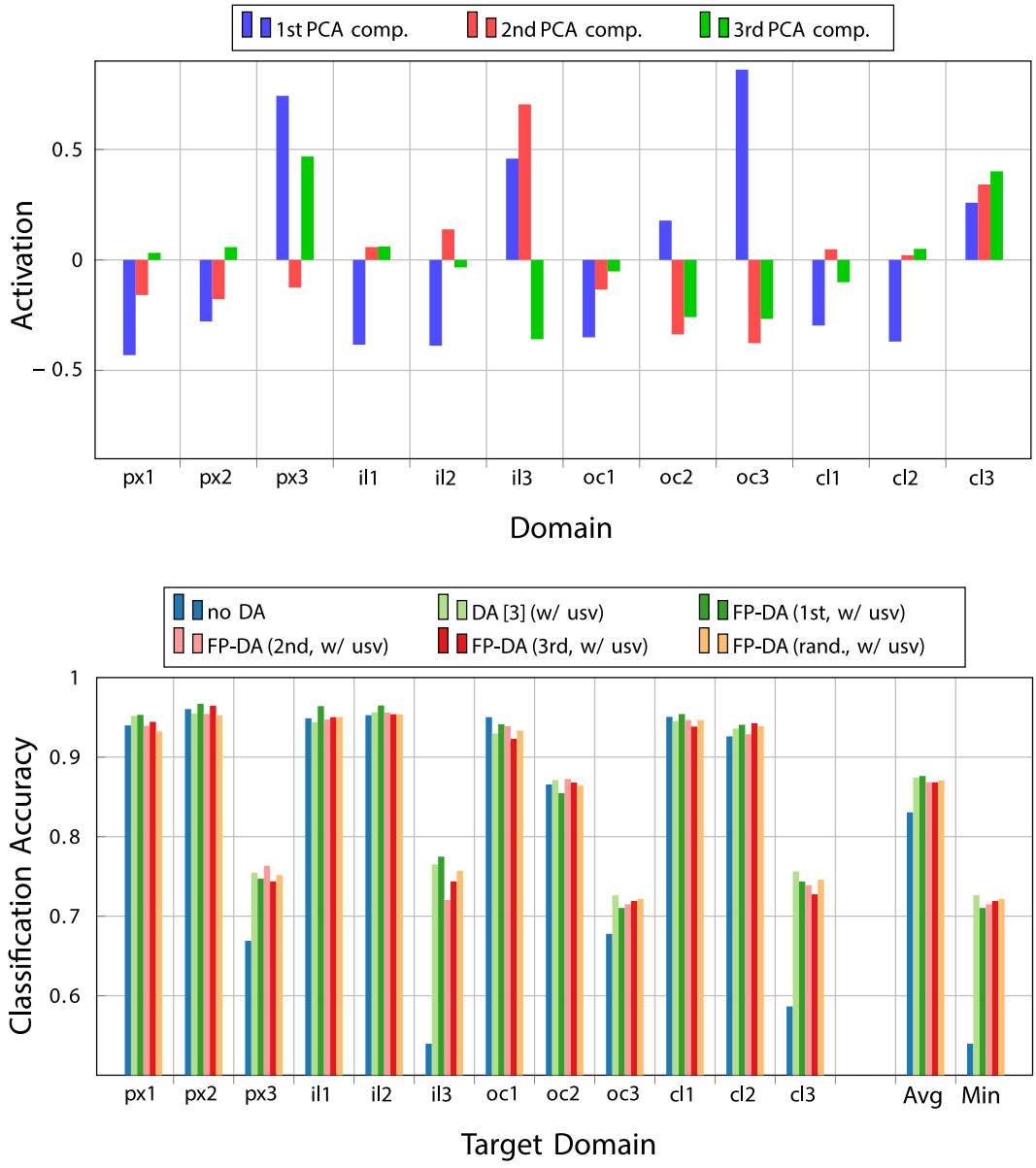

Fig. 13. PCA and FP-DA on the OpenLORIS dataset. *Top*: PCA identifies the most difficult domains plus oc2 in the first component. The second component suggests the groups {px, oc} and {il, cl}, while in the third component only px3 and cl3 are clearly separated from il3, oc2, and oc3. *Bottom*: DA causes negative transfer for most of the easy domains and positive transfer for the difficult ones. FP-DA increases the performance only slightly in *Avg* and decreases it in *Min*.

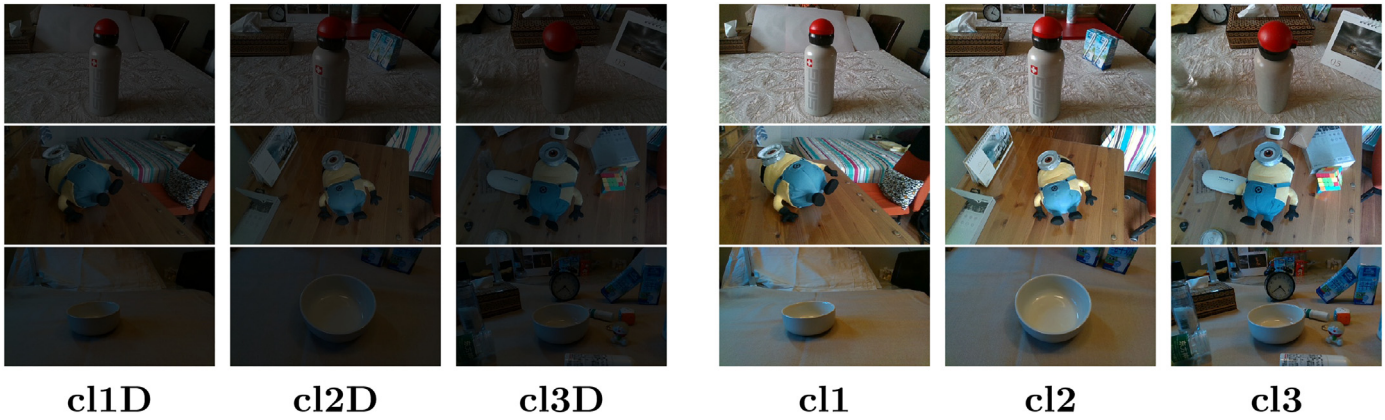

Fig. 14. Example images from the adapted OpenLORIS dataset. Domains indicated with cl*D were generated manually from the corresponding OpenLORIS domains cl* by darkening the images. Thus, each image has two factor labels, one label for *'clutter'* ∈ {level1, level2, level3} and one for *'brightness'* ∈ {dark, normal}.

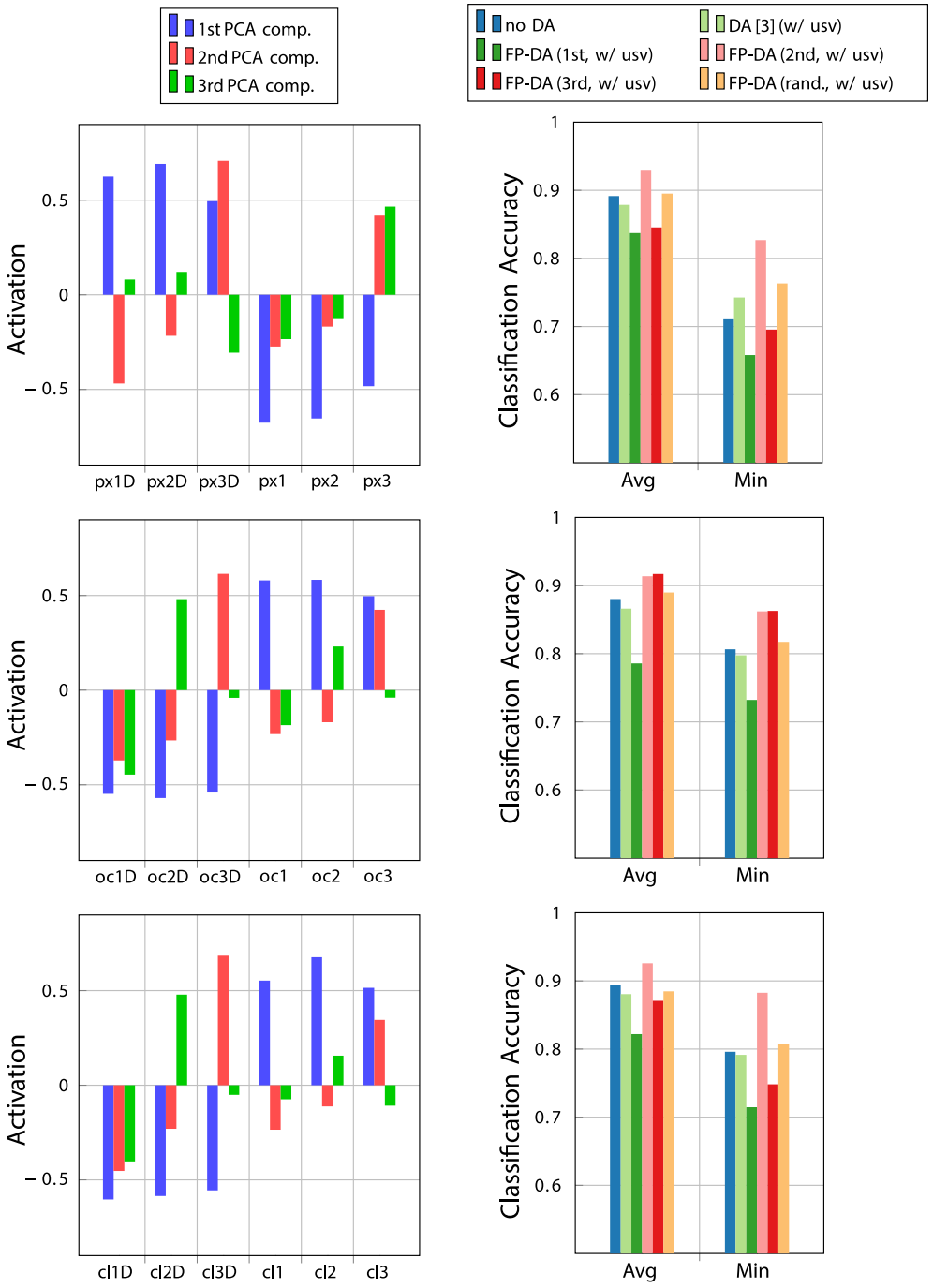

Fig. 15. FP-DA experiments on subsets of OpenLORIS domains with artificially darkened domains. The left column shows the PCA results, the right column the leave-one-out experiments where we only show the relevant average and minimum performance. Darkened domains are marked with *D. DA shows mostly negative transfer. FP-DA based on the first PCA component decreases the performance further, while we see a strong gain in performance in the second component.

6. Conclusion

In this work we discussed that current domain transfer literature lacks of more specific investigations of causes of negative transfer. Using only different data distributions as explanation for effects in DA is in our opinion not sufficient. For in-depth analysis we introduced in this paper our factor theory, which explains the appearance of an image by a mixture of multiple factors with different values that describe a scene. With focus on DA datasets, certain factors can be categorized as domain-informative and domains can be described by a combination of these factors, the domain factors. For a given classification task, certain factors can further be described as task-informative. With these categorizations we explained effects of machine learning and DA, where the challenge lies mainly in new factor values or new combinations of such in the test data.

The standard adversarial DA approach [3], targets to remove all domain factors from the feature representation for domain invariance. We discussed that this can lead to negative transfer if such factors are not only domain-informative, but also task-informative, or visually close to task-informative factors. We showed this negative transfer for the CORe50 and the OpenLORIS object dataset.

To actively preserve such factors and reduce negative transfer, we proposed FP-DA, a training method for multi-source DA that allows only competition within groups of domains that have the same value

for the factor to preserve. Using FP-DA, we showed in our experiments that by preserving such factors the negative transfer could be reduced and the highest average and highest minimum classification accuracy could be achieved on CORe50. On the standard OpenLORIS dataset FP-DA did not show an improvement due to the missing combinations of factors. However, by overlaying an additional factor we could show that FP-DA works best in all measures here as well. Furthermore, we applied PCA on one-to-one transfer result data to analyze multi-domain settings. We showed that one of the first PCA components can be a beneficial factor to be preserved. In the related work we mentioned several approaches that focus on complementary aspects of domain transfer, e.g. insufficient alignment of individual classes. In large multi-domain data all of these aspects can be expected. Therefore we target to combine our method with other methods in future.

CRediT authorship contribution statement

Sebastian Schrom: Conceptualization, Software, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Stephan Hasler**: Conceptualization, Methodology, Validation, Formal analysis, Writing - review & editing, Project administration. **Jürgen Adamy**: Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Torralba, A.A. Efros, Unbiased look at dataset bias, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2011, pp. 1521–1528, https://doi.org/10.1109/CVPR.2011.5995347.

[2] G. Csurka, A comprehensive survey on domain adaptation for visual applications, Domain Adaptation in Computer Vision Applications, Springer, Cham 2017, pp. 1–35.

[3] Y. Ganin, V.S. Lempitsky, Unsupervised domain adaptation by backpropagation, International Conference on Machine Learning (ICML) 2015, pp. 1180–1189.

[4] V. Lomonaco, D. Maltoni, Core50: a new dataset and benchmark for continuous object recognition, Conference on Robot Learning (CoRL), volume 78 of *Proceedings of Machine Learning Research*, PMLR 2017, pp. 17–26.

[5] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang, F. Qiao, R.H.M. Chan, OpenLORIS-Object: A robotic vision dataset and benchmark for lifelong deep learning, International Conference on Robotics and Automation (ICRA) 2020, pp. 4767–4773.

[6] K.R. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (2016) 9.

[7] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359.

[8] T. Tommasi, Learning to Learn by Exploiting Prior Knowledge, Ph.D. thesis École Polytechnique Fédérale de Lausanne, 2012.

[9] W.M. Kouw, An introduction to domain adaptation and transfer learning, CoRR abs/1812.11806, 2018.

[10] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: A large-scale hierarchical image database, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2009, pp. 248–255.

[11] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, 2007.

[12] T. DeVries, I. Misra, C. Wang, L. van der Maaten, Does object recognition work for everyone? Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, IEEE Computer Society 2019, pp. 52–59.

[13] S. Shankar, Y. Halpern, E. Breck, J. Atwood, J. Wilson, D. Sculley, No classification without representation: Assessing geodiversity issues in open data sets for the developing world, CoRR abs/1711.08536, 2017.

[14] X. Peng, B. Sun, K. Ali, K. Saenko, Learning deep object detectors from 3d models, in: International Conference on Computer Vision (ICCV), IEEE Computer Society, 2015 1278–1286, https://doi.org/10.1109/ICCV.2015.151.

[15] D. Li, Y. Yang, Y. Song, T.M. Hospedales, Deeper, broader and artier domain generalization, International Conference on Computer Vision (ICCV), IEEE Computer Society 2017, pp. 5543–5551, https://doi.org/10.1109/ICCV.2017.591.

[16] P. Koniusz, Y. Tas, H. Zhang, M.T. Harandi, F. Porikli, R. Zhang, Museum exhibit identification challenge for the supervised domain adaptation and beyond, European Conference on Computer Vision (ECCV), volume 11220 of *Lecture Notes in Computer Science*, Springer 2018, pp. 815–833, https://doi.org/10.1007/978-3-030-01270-0_48.

[17] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, K. Saenko, Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation, CoRR abs/1806.09755, 2018.

[18] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2017, pp. 5385–5394, https://doi.org/10.1109/CVPR.2017.572.

[19] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), European Conference on Computer Vision (ECCV), volume 6314 of *Lecture Notes in Computer Science*, Springer 2010, pp. 213–226, https://doi.org/10.1007/978-3-642-15561-1_16.

[20] A. Bergamo, L. Torresani, Exploiting Weakly-Labeled Web Images to Improve Object Classification: A Domain Adaptation Approach, in: Conference on Neural Information Processing Systems (NIPS), Curran Associates, Inc., 2010 (pp. 181–189).

[21] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, International Conference on Computer Vision (ICCV), IEEE Computer Society 2019, pp. 1406–1415, https://doi.org/10.1109/ICCV.2019.00149.

[22] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.

[23] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, Challenging common assumptions in the unsupervised learning of disentangled representations, International Conference on Machine Learning (ICML), volume 97 of *Proceedings of Machine Learning Research* 2019, pp. 4114–4124.

[24] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A.J. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (2012) 723–773.

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

[26] M. Long, Y. Cao, J. Wang, M.I. Jordan, Learning transferable features with deep adaptation networks, International Conference on Machine Learning (ICML), volume 37 of *JMLR Workshop and Conference Proceedings* 2015, pp. 97–105 , JMLR.org.

[27] M. Long, H. Zhu, J. Wang, M.I. Jordan, Unsupervised domain adaptation with residual transfer networks, in: D.D. Lee, M. Sugiyama, U. Von Luxburg, I. Guyon, R. Garnett (Eds.),Conference on Neural Information Processing Systems (NIPS) 2016, pp. 136–144.

[28] M. Long, Y. Cao, Z. Cao, J. Wang, M.I. Jordan, Transferable representation learning with deep adaptation networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2019) 3071–3085.

[29] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, D. Erhan, Domain separation networks, Conference on Neural Information Processing Systems (NIPS) 2016, pp. 343–351.

[30] M. Karimpour, S. Noori Saray, J. Tahmoresnezhad, M. Pourmahmood Aghababa, Multi-source domain adaptation for image classification, Mach. Vis. Appl. 31 (2020) 44.

[31] E. Tzeng, J. Hoffman, T. Darrell, K. Saenko, Simultaneous deep transfer across domains and tasks, International Conference on Computer Vision (ICCV), IEEE Computer Society 2015, pp. 4068–4076, https://doi.org/10.1109/ICCV.2015.463.

[32] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, Domain-adversarial neural networks, CoRR abs/1412.4446, 2014.

[33] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V.S. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (2016) 59:1–59: 35.

[34] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional Adversarial Domain Adaptation, in: Conference on Neural Information Processing Systems (NeurIPS), 2018 1647–1657.

[35] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2017, pp. 2962–2971, https://doi.org/10.1109/CVPR.2017.316.

[36] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, Y. Bengio, Generative adversarial nets, Conference on Neural Information Processing Systems (NIPS) 2014, pp. 2672–2680.

[37] M.T. Rosenstein, Z. Marx, L.P. Kaelbling, T.G. Dietterich, To transfer or not to transfer, Conference on Neural Information Processing Systems (NIPS) Workshop on Transfer Learning, 898, 2005, pp. 1–4.

[38] S. Schrom, S. Hasler, Effects of domain awareness in generalizing over cameras in road detection, Machine Learning Reports 03/2017, 2017.

[39] S. Schrom, S. Hasler, Domain mixture: An overlooked scenario in domain adaptation, International Conference On Machine Learning And Applications (ICMLA), IEEE Computer Society 2019, pp. 22–27, https://doi.org/10.1109/ICMLA.2019.00013.

[40] Z. Cao, K. You, M. Long, J. Wang, Q. Yang, Learning to transfer examples for partial domain adaptation, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2019, pp. 2985–2994, https://doi.org/10.1109/CVPR.2019.00310.

[41] L. Ge, J. Gao, H.Q. Ngo, K. Li, A. Zhang, On handling negative transfer and imbalanced distributions in multiple source transfer learning, Statist. Analys. Data Mining 7 (2014) 254–271.

[42] J. Hoffman, B. Kulis, T. Darrell, K. Saenko, Discovering latent domains for multisource domain adaptation, in: A.W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part II, volume 7573 of *Lecture Notes in Computer Science*, Springer 2012, pp. 702–715, https://doi.org/10.1007/978-3-642-33709-3_50.

[43] B. Gong, K. Grauman, F. Sha, Reshaping visual datasets for domain adaptation, in: C.J.C. Burges, L. Bottou, Z. Ghahramani, K.Q. Weinberger (Eds.),Advances in Neural Information Processing Systems NIPS 2013, pp. 1286–1294.

[44] C. Xiong, S. McCloskey, S. Hsieh, J.J. Corso, Latent domains modeling for visual domain adaptation, in: C.E. Brodley, P. Stone (Eds.), Conference on Artificial Intelligence, AAAI Press 2014, pp. 2860–2866.
[45] M. Mancini, L. Porzi, S.R. Bulò, B. Caputo, E. Ricci, Boosting domain adaptation by discovering latent domains, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2018, pp. 3771–3780, https://doi.org/10.1109/CVPR.2018.00397.
[46] X. Jiang, Q. Lao, S. Matwin, M. Havaei, Implicit class-conditioned domain alignment for unsupervised domain adaptation, International Conference on Machine Learning (ICML), volume 119 of *Proceedings of Machine Learning Research*, PMLR 2020, pp. 4816–4827.
[47] S. Xie, Z. Zheng, L. Chen, C. Chen, Learning semantic representations for unsupervised domain adaptation, in: J.G. Dy, A. Krause (Eds.), International Conference on Machine Learning (ICML), volume 80 of *Proceedings of Machine Learning Research*, PMLR 2018, pp. 5419–5428.
[48] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, T. Mei, Transferrable prototypical networks for unsupervised domain adaptation, Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision Foundation / IEEE 2019, pp. 2239–2247, https://doi.org/10.1109/CVPR.2019.00234.
[49] J. Liang, R. He, Z. Sun, T. Tan, Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation, Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision Foundation / IEEE 2019, pp. 2975–2984, https://doi.org/10.1109/CVPR.2019.00309.
[50] Z. Deng, Y. Luo, J. Zhu, Cluster alignment with a teacher for unsupervised domain adaptation, International Conference on Computer Vision (ICCV), IEEE 2019, pp. 9943–9952, https://doi.org/10.1109/ICCV.2019.01004.
[51] Z. Pei, Z. Cao, M. Long, J. Wang, Multi-adversarial domain adaptation, in: S.A. McIlraith, K.Q. Weinberger (Eds.), AAAI Conference on Artificial Intelligence, AAAI Press 2018, pp. 3934–3941.
[52] L. Hu, M. Kan, S. Shan, X. Chen, Unsupervised domain adaptation with hierarchical gradient synchronization, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2020, pp. 4042–4051, https://doi.org/10.1109/CVPR42600.2020.00410.
[53] Q. Zhou, S. Wang, et al., Cluster adaptation networks for unsupervised domain adaptation, Image Vis. Comput. 108 (2021) 104137.
[54] Y. Zhang, N. Wang, S. Cai, Adversarial sliced wasserstein domain adaptation networks, Image Vis. Comput. 102 (2020) 103974.
[55] M. Schutera, F.M. Hafner, J. Abhau, V. Hagenmeyer, R. Mikut, M. Reischl, Cuepervision: self-supervised learning for continuous domain adaptation without catastrophic forgetting, Image Vis. Comput. 106 (2021) 104079.
[56] L. Duan, D. Xu, S. Chang, Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2012, pp. 1338–1345, https://doi.org/10.1109/CVPR.2012.6247819.
[57] J. Guo, D.J. Shah, R. Barzilay, Multi-source domain adaptation with mixture of experts, Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics 2018, pp. 4694–4703.
[58] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain Adaptation with Multiple Sources, in: Conference on Neural Information Processing Systems (NIPS), Curran Associates, Inc, 2008 1041–1048.
[59] I. Redko, N. Courty, R. Flamary, D. Tuia, Optimal transport for multi-source domain adaptation under target shift, Conference on Artificial Intelligence and Statistics (AISTATS), volume 89 of *Proceedings of Machine Learning Research*, PMLR 2019, pp. 849–858.
[60] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, K. Keutzer, Multi-source distilling domain adaptation, in: AAAI Conference on Artificial Intelligence, AAAI Press, 2020 12975–12983.
[61] R. Xu, Z. Chen, W. Zuo, J. Yan, L. Lin, Deep cocktail network: Multi-source unsupervised domain adaptation with category shift, Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society 2018, pp. 3964–3973, https://doi.org/10.1109/CVPR.2018.00417.
[62] J. Wen, R. Greiner, D. Schuurmans, Domain aggregation networks for multi-source domain adaptation, International Conference on Machine Learning (ICML), volume 119 of *Proceedings of Machine Learning Research*, PMLR 2020, pp. 10214–10224.
[63] Y. Li, M. Murias, G. Dawson, D.E. Carlson, Extracting relationships by multi-domain matching, Conference on Neural Information Processing Systems (NeurIPS) 2018, pp. 6799–6810.
[64] S. Rakshit, B. Banerjee, G. Roig, S. Chaudhuri, Unsupervised multi-source domain adaptation driven by deep adversarial ensemble learning, German Conference on Pattern Recognition (GCPR), volume 11824 of *Lecture Notes in Computer Science*, Springer 2019, pp. 485–498, https://doi.org/10.1007/978-3-030-33676-9_34.
[65] H. Wang, W. Yang, Z. Lin, Y. Yu, Tmda: Task-specific multi-source domain adaptation via clustering embedded adversarial training, International Conference on Data Mining (ICDM), IEEE Computer Society 2019, pp. 1372–1377.
[66] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations (ICLR), 2015.
[67] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, International Conference on Computer Vision (ICCV), IEEE Computer Society 2017, pp. 618–626, https://doi.org/10.1109/ICCV.2017.74.