Domain Adaptation by Mixture of Alignments of Second- or Higher-Order Scatter Tensors

Piotr Koniusz*[,1,2] Yusuf Tas*[,1,2] Fatih Porikli[2]
[1]Data61/CSIRO, [2]Australian National University
firstname.lastname@{data61.csiro.au, anu.edu.au}

Abstract

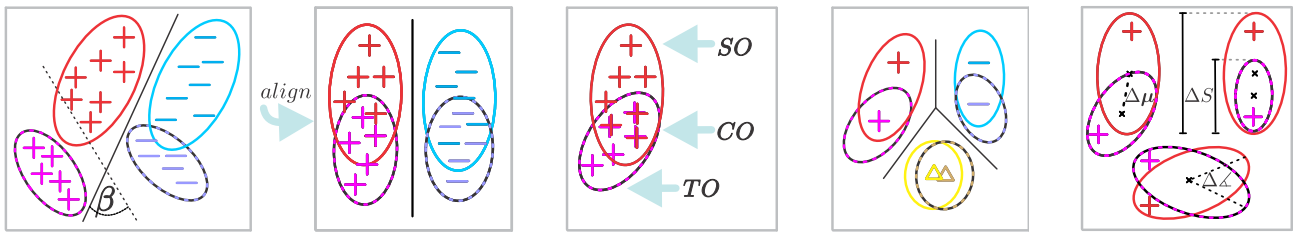
In this paper, we propose an approach to the domain adaptation, dubbed Second- or Higher-order Transfer of Knowledge (So-HoT), based on the mixture of alignments of second- or higher-order scatter statistics between the source and target domains. The human ability to learn from few labeled samples is a recurring motivation in the literature for domain adaptation. Towards this end, we investigate the supervised target scenario for which few labeled target training samples per category exist. Specifically, we utilize two CNN streams: the source and target networks fused at the classifier level. Features from the fully connected layers fc7 of each network are used to compute second- or even higher-order scatter tensors; one per network stream per class. As the source and target distributions are somewhat different despite being related, we align the scatters of the two network streams of the same class (within-class scatters) to a desired degree with our bespoke loss while maintaining good separation of the between-class scatters. We train the entire network in end-to-end fashion. We provide evaluations on the standard Office benchmark (visual domains) and RGB-D combined with Caltech256 (depth-to-rgb transfer). We attain state-of-the-art results.

1. Introduction

Domain adaptation and transfer learning are the problems widely studied in computer vision and machine learning communities [1, 27]. They are directly inspired by the human cognitive abilities of generalizing to new concepts from very few data samples (cf. training from scratch on over a million of labeled images of the ImageNet dataset [31]). From psychological point of view, transfer of learning is *"the dependency of human conduct, learning or performance on prior experience"*. This problem was introduced in 1901 under a notion of "*transfer of particle*" [46]. In machine learning, transfer learning (or inductive learning) concerns *"storing knowledge gained while solving one problem and applying it to a different but related problem"* [45]. In practical computer vision and machine learning systems, transfer learning refers to *"an ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality"*. In general, given a new (target) task, the arising question is how to identify the commonality between this task and previous (source) tasks, and transfer knowledge from the previous tasks to the target one. Therefore, one has to address three questions: what to transfer, how to transfer, and when to transfer [39].

In what follows, we propose an approach to the domain adaptation, dubbed Second- or Higher-order Transfer of Knowledge (*So-HoT*), based on the mixture of alignments of second- and/or higher-order scatter statistics between the source and target domains. Specifically, we utilize second- and/or higher-order scatter tensors, one per each network stream per class, such that the first stream corresponds to the source domain while the second to the target. The scatters are built from the feature vectors produced by the *fc7* layer of AlexNet [22]. We propose that, as the source and target distributions are only partially related by their commonality, the scatters of the same class from both streams (*within-class scatters*) should be aligned to a desired degree to capture this commonality as an overlap between parts of the two distributions. At the same time, to achieve high classification accuracy, we maintain good separation between the scatters representing different classes (*between-class scatters*). We devise a simple loss that brings each pair of within-class scatters closer in terms of their covariances as well as their corresponding means. Therefore, the CNN parameters stored by convolutional filters and weights of the target network regularized by the source data in this end-to-end fashion must produce statistics consistent with the source network. We view such a regularization paradigm as being motivated by the theory of privileged learning [41]. In our case, the statistics of the source network regularize

*Both authors contributed equally.



(a) (b) (c) (d) (e)

Figure 1: Our alignment problem. In Figure 1a, a two-class toy problem with positive and negative samples (+) and (−) is given. The solid and dashed ellipses indicate the source and target domain distributions. The two hyperplane lines that separate (+) from (−) on the target data indicate large uncertainty (denoted as β) in the optimal orientation for the target problem. Figure 1b shows that the source and target distributions can be aligned enough to separate well two classes for both the source and target problems. Figure 1c shows that partially aligned distributions have the commonality (*CO*) as well as the source and target specific parts (*SO*) and (*TO*) that represent dissimilarity between the source and target. Figure 1d depicts a multi-class problem. Beside of partially aligned means, the orientations of the source and target distributions are allowed to partially differ – as a result, they *i.e.* fit better into the piece-wise linear decision boundary. Figure 1e shows that differences in means $\Delta\boldsymbol{\mu}$, scale/shear ΔS and orientation $\Delta\measuredangle$ of within-class scatters are all part of the alignment process.

the target (and vice-versa) whilst in the privileged learning, the side information regularizes the solution dictated by the empirical loss evaluated on the main data samples. See Figures 1 and 2 for illustrative examples.

Furthermore, as distributions of the source and target domains may require different level of alignment per class (the commonality depends on the class label), we investigate not only an unweighted alignment loss (class-independent level of alignment) but also its weighted counterpart which learns one weight per class (class-specific levels of alignment).

Additionally, as we work with second- and/or higher-order tensors, we propose a kernelized variant of our alignment loss which provides computational speed-ups for typical domain adaptation datasets.

To summarize, our main contributions are: i) a novel loss that we call *So-HoT*, which defines the commonality between the source and target domains as the mixture of alignments of second- and/or higher-order scatter tensors, ii) unweighted and weighted variants of alignments, and iii) a fast kernelized alternative of our alignment loss.

Next, we detail the notion of domain adaptation and transfer learning, review the related literature and explain how our work differs from the state-of-the-art approaches.

2. Related Work

Domain adaptation assumes that the transfer of knowledge takes place among two or more distinct domains *e.g.*, e-commerce reviews and biomedical data. In contrast, transfer learning utilizes the same domain *e.g.*, images of natural scenes with related but different distributions where the goal may be to learn objects of a new class while leveraging other already learned classes [37, 39]. Not surprisingly, these both notions are often interchangeable *e.g.*, natural images and sketches have related distributions but they come from distinct domains at the same time. Another example is a so-called domain shift *e.g.*, bicycle in natural images vs. on-line retailer galleries. Transfer of knowledge may vary from simply carrying over discriminative information from a source to target domain under the same set of classes to inferring a solution to a new distinct task from a set of former ones [37, 17]. Domain adaptation comes in many flavors. Single- or multiple-source [4] setups are possible *e.g.*, single stream of natural images vs. multiple streams supplied with photos of objects: on cluttered backgrounds, on a clear background, in a daytime or night setting, or even in multi-spectral setting. Moreover, the problem in hand may be homogeneous or heterogeneous [39, 47] in nature *e.g.*, identical source and target representations using RGB images vs. a source represented by a CNN trained on images [7, 22] and a target using an LSTM [16, 15] which is trained on text corpora or video data [13]. The architecture in use may be shallow [5, 36] or deep [8] so that the commonality is established only at the classifier level or across entire source and target networks, respectively. Noteworthy is also recent trend in the CNN fine-tuning which by itself is a powerful domain adaptation and transfer learning tool [10, 33] which requires large training datasets. Moreover, domain adaptation and transfer learning address problems such as: learning new categories from few annotated samples (supervised domain adaptation [3, 40]), utilizing available unlabeled data (unsupervised [36, 8, 14] or semi-supervised domain adaptation [5, 40]), recognizing new categories in embedded spaces *e.g.*, attribute-based, without any training samples (zero-shot learning [24]).

In this paper, we investigate the case of a deep supervised single-source domain adaptation which can be easily extended to the multi-source and heterogeneous cases.

The Commonality. Deep learning [22, 35, 12] has been used in the context of domain adaptation in recent works *e.g.*, [40, 8, 3, 44, 23, 38, 28]. These works differ in how they establish the so-called commonality between domains. In [40], the authors propose to align both domains via the

cross entropy which "maximally confuses" both domains for supervised and semi-supervised settings. In [8], an unsupervised approach utilizes the assumption that predictions must be made based on features which cannot discriminate between the source and target domains. Specifically, they minimize a trade-off between the so-called source risk and the empirical divergence to find examples in the source domain indistinguishable from the target samples.

Our work differs from these methods in that we define the commonality as the desired degree of overlap between the second- and/or higher-order scatters of the source and target. After such an alignment, we let the non-overlapping tails of distributions also guide learning which results in a more general classifier (*i.e.* avoid the domain-specific bias).

Moreover, in [3], the authors capture the "interpolating path" between the source and target domains using linear projections into a low-dimensional subspace which lies on the Grassman manifold. In [44], the authors propose to learn the transformation between the source and target by the deep model regression network. These two approaches assume that the source representation can be interpolated or regressed into the target as, given the nature of CNNs, they can approximate highly non-linear functions.

Our model differs in that our source and target network streams co-regularize each other to produce the commonality between the source and target distributions and accommodate the domain-specific parts that should not be aligned.

For visual domains, the commonality can be captured in the spatially-local sense. In [38], the authors utilize so-called "domainness maps" which capture locally the degree of domain specificity. Similarly, in [23], the authors extract local patches of varying sizes at process each of these patches via CNNs. Our work is orthogonal to these techniques. We represent the commonality globally, however, our ideas could also be applied in a spatially-local setting.

Correlation Methods. Some recent works use correlation between the source and target distributions. Inspired by the Canonical Correlation Analysis (CCA), the authors of [47] utilize a correlation subspace as a joint representation for associating the data across different domains. They also use kernelized CCA. In [36], the authors propose an unsupervised domain adaptation by the correlation alignment.

Our work is somewhat related in that we utilize second-order statistics. However, we align partially the class-specific source and target distributions to define the commonality (partial intersection of scatters) in the supervised setting. We also align partially the distribution means while the above unsupervised approaches use zero-centered feature vectors and the full alignment of the generic (c.f. class-specific) source and target distributions. We detail how to learn the degree of alignment in an end-to-end fashion and introduce the kernelized loss between the second- and/or higher-order scatter tensors; all being novel propositions.

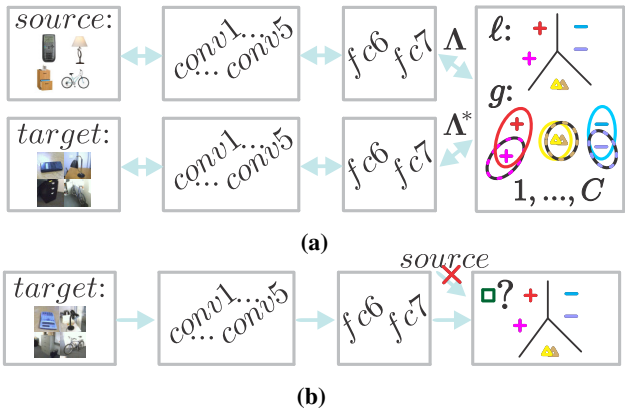


Figure 2: The pipeline. Figure 2a shows the source and target network streams which merge at the classifier level. The classification and alignment loss ℓ and g take the data $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$ from both streams and participate in end-to-end learning. At the test time, we use the target stream and the trained classifier as in Figure 2b.

Tensor Methods. Correlation approaches outlined above use second-order scatter matrices which are tensors of order $r=2$. In this work, we also investigate the applicability of higher-order scatters $r \geq 3$ for alignment. Third-order tensors have been found useful for various vision tasks. For example, spatio-temporal third-order tensor on video data is proposed for action analysis in [18], non-negative tensor factorization is used for image denoising in [34], tensor textures are proposed for texture rendering in [43], and higher order tensors are used for face recognition in [42]. A survey of multi-linear algebraic methods for tensor subspace learning is available in [29]. The above applications use a single tensor, while our goal is to use tensors as the domain- and class-specific representations, similar to the sum-kernel approaches [21, 19, 20], and apply them to alignment tasks.

3. Background

In this section, we review notations and the necessary background on scatter tensors, polynomial kernels and their linearizations, which are useful in deriving our mixture of alignments of second- and/or higher-order scatter tensors.

3.1. Notations

Let $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional feature vector. Then, we use $\boldsymbol{\mathcal{X}} = \uparrow \otimes_r \mathbf{x}$ to denote the r-mode super-symmetric rank-one tensor $\boldsymbol{\mathcal{X}}$ generated by the r-th order outer-product of $\mathbf{x}$, where the element of $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d_{\times^r}$ at the $(i_1, i_2, ..., i_r)$-th index is given by $\Pi_{j=1}^r x_{i_j}$. $\mathcal{I}_N$ stands for the index set $\{1, 2, ..., N\}$. We denote the space of super-symmetric tensors of dimension $d \times ... \times d$ with r modes as $\mathfrak{S}^d_{\times^r} \subset \mathbb{R}^{\times^r d}$, where $\mathbb{R}^{\times^r d}$ is the space of tensors $\mathbb{R}^{d \times ... \times d}$ with r modes. The Frobenius norm of tensor is given by $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum_{i_1, i_2, ..., i_r} \mathcal{X}^2_{i_1, i_2, ..., i_r}}$, where $\mathcal{X}_{i_1, i_2, ..., i_r}$ represents

the $(i_1, i_2, ..., i_r)$-th element of $\boldsymbol{\mathcal{X}}$. Similarly, the inner-product between two tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ is given by $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1,i_2,...,i_r} \mathcal{X}_{i_1,i_2,...,i_r} \cdot \mathcal{Y}_{i_1,i_2,...,i_r}$. Using Matlab style notation, the $(i_3, ..., i_r)$-th slice of $\boldsymbol{\mathcal{X}}$ is given by $\boldsymbol{\mathcal{X}}_{:,:,i_3,...,i_r}$. The space of positive semi-definite matrices is $\mathcal{S}_+^d$. Lastly, $\mathbb{1}$ denotes a vector with all coefficients equal one.

3.2. Second- or Higher-order Scatter Tensors

We define a scatter tensor of order r as a mean-centered TOSST representation [19]:

Definition 1. *Suppose $\boldsymbol{\phi}_n \in \mathbb{R}^d, \forall n \in \mathcal{I}_N$ represent some data vectors, then a scatter tensor $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d_{\times^r}$ of order r on these data vectors is given by:*

$$\boldsymbol{\mathcal{X}} = \frac{1}{N} \sum_{n=1}^{N} \uparrow\otimes_r (\boldsymbol{\phi}_n - \boldsymbol{\mu}) \quad \textit{and} \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\phi}_n. \tag{1}$$

In our supervised domain adaptation setting, the scatter tensors are obtained via applying (1) on the class-specific data vectors such as outputs of the *fc7* layer of AlexNet. When we need to highlight order r of $\boldsymbol{\mathcal{X}}$, we write $\boldsymbol{\mathcal{X}}^{(r)}$. The following properties of the scatter tensors are worth noting (see [19] for proofs):

Proposition 1. *For a scatter tensor $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d_{\times^r}$, we have:*

1. *Super-Symmetry: $\boldsymbol{\mathcal{X}}_{i_1,i_2,...,i_r} = \boldsymbol{\mathcal{X}}_{\Pi(i_1,i_2,...,i_r)}$ for indexes $(i_1, i_2, ..., i_r)$ and their any permutation Π. The number of unique coefficients of $\boldsymbol{\mathcal{X}}$ is $\binom{d+r-1}{r}$.*
2. *Every slice is at least positive semi-definite for any even order $r \geq 2$ and $\boldsymbol{\mathcal{X}}_{:,:,i_3,...,i_r} \in \mathcal{S}_+^d, \forall (i_3, ..., i_r) \in \mathcal{I}_d$. For $r = 2$, tensor $\boldsymbol{\mathcal{X}}$ also is a covariance matrix.*
3. *Indefiniteness for any odd order $r \geq 1$,* i.e.*, under a CP decomposition [26], it can have positive, negative, or zero entries in its core-tensor.*

Due to the indefiniteness of tensors of odd orders and potential rank deficiency, we restrict ourselves to work with the Euclidean distance between such scatter representations. Also, as the number of unique coefficients of $\boldsymbol{\mathcal{X}}$ is of order $\sim d^r$, which is prohibitive for $r \geq 3$, we propose a light-weight kernelized variant of the Euclidean distance which avoids explicit use of tensors. The following easily verifiable two results will come handy in the sequel:

Proposition 2. *Suppose we want to evaluate the Frobenius norm between tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^* \in \mathfrak{S}^d_{\times^r}$, then it holds that:*

$$||\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^*||_F^2 = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle - 2 \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^* \rangle + \langle \boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{X}}^* \rangle. \tag{2}$$

Proof. $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}^*$ can be vectorized and the Frobenius norm replaced by the ℓ_2-norm for which the above expansion is known to hold. □

Proposition 3. *Suppose $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ are two arbitrary vectors, then for an ordinal $r > 0$, we have:*

$$\langle \mathbf{x}, \mathbf{y} \rangle^r = \langle \uparrow\otimes_r \mathbf{x}, \uparrow\otimes_r \mathbf{y} \rangle. \tag{3}$$

Moreover, for sets of vectors $\mathbf{x}_n, \mathbf{y}_{n'} \in \mathbb{R}^d$, we have:

$$\sum_n \sum_{n'} \langle \mathbf{x}_n, \mathbf{y}_{n'} \rangle^r = \Big\langle \sum_n \uparrow\otimes_r \mathbf{x}_n, \sum_{n'} \uparrow\otimes_r \mathbf{y}_{n'} \Big\rangle. \tag{4}$$

Proof. The expansion in (3) is derived in [21] while (4) can be verified due to bilinear properties of the dot-product. □

4. Proposed Approach

In this section, we first formulate the problem of mixture of alignments of second- and/or higher-order scatter tensors, which precedes an exposition to our next two contributions: a weighted mixture of alignments and a kernelized approach which avoids explicit evaluations of scatters.

4.1. Problem Formulation

Suppose $\mathcal{I}_N$ and $\mathcal{I}_{N^*}$ are the indexes of N source and N^* target training data points. $\mathcal{I}_{N_c}$ and $\mathcal{I}_{N_c^*}$ are the class-specific indexes for $c \in \mathcal{I}_C$, where C is the number of classes. Suppose we have feature vectors from *fc7* in the source network stream, one per image, and associated with them labels. Such pairs are given by $\boldsymbol{\Lambda} \equiv \{(\boldsymbol{\phi}_n, y_n)\}_{n \in \mathcal{I}_N}$, where $\boldsymbol{\phi}_n \in \mathbb{R}^d$ and $y_n \in \mathcal{I}_C$, $\forall n \in \mathcal{I}_N$, as shown in Figure 2a. For the target data, by analogy, we define pairs $\boldsymbol{\Lambda}^* \equiv \{(\boldsymbol{\phi}_n^*, y_n^*)\}_{n \in \mathcal{I}_N^*}$, where $\boldsymbol{\phi}^* \in \mathbb{R}^d$ and $y_n^* \in \mathcal{I}_C$, $\forall n \in \mathcal{I}_N^*$. Class-specific sets of feature vectors are given as $\boldsymbol{\Phi}_c \equiv \{\boldsymbol{\phi}_n^c\}_{n \in \mathcal{I}_{N_c}}$ and $\boldsymbol{\Phi}_c^* \equiv \{\boldsymbol{\phi}_n^{c*}\}_{n \in \mathcal{I}_{N_c^*}}$, $\forall c \in \mathcal{I}_C$. Then, $\boldsymbol{\Phi} \equiv (\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_C)$ and $\boldsymbol{\Phi}^* \equiv (\boldsymbol{\Phi}_1^*, ..., \boldsymbol{\Phi}_C^*)$. Note that we use the asterisk symbol written in superscript (*e.g.* $\boldsymbol{\phi}^*$) to denote variables associated with the target network whilst the source-related and generic variables have no such indicator. Below, we formulate our problem as a trade-off between the classifier loss ℓ and the alignment loss g which acts on the scatter tensors and related to them means:

$$\begin{aligned} \underset{\substack{\mathbf{W},\mathbf{b},\boldsymbol{\Theta},\boldsymbol{\Theta}^* \\ \text{s. t. } ||\boldsymbol{\phi}_n||_2^2 \leq \tau, \\ ||\boldsymbol{\phi}_{n'}^*||_2^2 \leq \tau, \\ \forall n \in \mathcal{I}_N, n' \in \mathcal{I}_N^*}}{\arg\min} \quad & \ell(\mathbf{W}, \mathbf{b}, \boldsymbol{\Lambda} \cup \boldsymbol{\Lambda}^*) + \lambda ||\mathbf{W}||_F^2 \\ & \underbrace{+ \frac{\sigma_1}{C} \sum_{c \in \mathcal{I}_C} ||\boldsymbol{\mathcal{X}}_c - \boldsymbol{\mathcal{X}}_c^*||_F^2 + \frac{\sigma_2}{C} \sum_{c \in \mathcal{I}_C} ||\boldsymbol{\mu}_c - \boldsymbol{\mu}_c^*||_2^2}_{g(\boldsymbol{\Phi}, \boldsymbol{\Phi}^*)}. \end{aligned} \tag{5}$$

For ℓ, we use a generic loss used by CNNs, say Softmax. The matrix $\mathbf{W} \in \mathbb{R}^{d \times C}$ contains unnormalized probabilities (c.f. hyperplane of SVM), $\mathbf{b} \in \mathbb{R}^C$ is the bias term, and λ is the regularization constant. Moreover, the union $\boldsymbol{\Lambda} \cup \boldsymbol{\Lambda}^*$ of the source and target training data reveals that we train one universal classifier for both domains[1]. In Equation (5),

[1] For VGG streams, we use a couple of domain-specific classifiers *e.g.*, $\ell(\mathbf{W}, \mathbf{b}, \boldsymbol{\Lambda}) + \ell(\mathbf{W}^*, \mathbf{b}^*, \boldsymbol{\Lambda}^*) + \lambda ||\mathbf{W}||_F^2 + \lambda^* ||\mathbf{W}^*||_F^2 + \beta' ||\mathbf{W} - \mathbf{W}^*||_F^2$.

separating the class-specific distributions is addressed by ℓ while bringing closer the within-class scatters of both network streams is handled by g (as Figure 2 shows).

Specifically, our loss g depends on two sets of variables $(\boldsymbol{\mathcal{X}}_1(\boldsymbol{\Phi}_1),...,\boldsymbol{\mathcal{X}}_C(\boldsymbol{\Phi}_c)),(\boldsymbol{\mu}_1(\boldsymbol{\Phi}_1),...,\boldsymbol{\mu}_C(\boldsymbol{\Phi}_C))$ and $(\boldsymbol{\mathcal{X}}^*_1(\boldsymbol{\Phi}^*_1),...,\boldsymbol{\mathcal{X}}^*_C(\boldsymbol{\Phi}^*_C)),(\boldsymbol{\mu}^*_1(\boldsymbol{\Phi}^*_1),...,\boldsymbol{\mu}^*_C(\boldsymbol{\Phi}^*_C))$ – one set per network stream. Feature vectors $\boldsymbol{\Phi}(\boldsymbol{\Theta})$ and $\boldsymbol{\Phi}^*(\boldsymbol{\Theta}^*)$ depend on the parameters of the source and target network streams $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}^*$ that we optimize over *e.g.*, they represent coefficients of convolutional filters and weights of *fc* layers. $\boldsymbol{\mathcal{X}}_c$, $\boldsymbol{\mathcal{X}}^*_c$, $\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}^*_c$ denote the scatter tensors and means, respectively, one tensor/mean pair per network stream per class, evaluated as in (1). Lastly, σ_1 and σ_2 control the overall degree of the scatter and mean alignment, τ constraints the ℓ_2-norm of feature vectors (needed if λ is low). Derivatives of loss g are given in Appendix A.

In this work, we assume that highly non-linear CNN streams are able to rotate the within-class scatters sufficiently as dictated by our loss to yield a desired overlap of two scatters. Such an assumption is common in *i.e.* [3, 44].

4.2. Weighted Alignment Loss

Below we propose a weighted variant of alignment loss g that incorporates class-specific weights $\boldsymbol{\zeta},\bar{\boldsymbol{\zeta}}\in\mathbb{R}^C$ that adjust the degree of alignment per class between the within-class scatters as well as related to them means. As the statistical literature states that combination of moments $m=1,...,\infty$ can capture any distribution, we combine $r'=2,...,r$ orders:

$$g^{(r)}(\boldsymbol{\Phi},\boldsymbol{\Phi}^*,\{\boldsymbol{\zeta}_{r'}\}_{r'\in\mathcal{I}_r},\bar{\boldsymbol{\zeta}})=\frac{\sigma_1}{rC}\sum_{r'\in\mathcal{I}_r\setminus\{1\}}\sum_{c\in\mathcal{I}_C}\boldsymbol{\zeta}_{cr'}||\boldsymbol{\mathcal{X}}^{(r')}_c-\boldsymbol{\mathcal{X}}^{*(r')}_c||^2_F$$
$$+\frac{\sigma_2}{C}\sum_{c\in\mathcal{I}_C}\bar{\boldsymbol{\zeta}}_c||\boldsymbol{\mu}_c-\boldsymbol{\mu}^*_c||^2_2+\frac{\alpha_1}{r}\sum_{r'\in\mathcal{I}_r\setminus\{1\}}||\boldsymbol{\zeta}_{r'}-\mathbb{1}||^2_2+\alpha_2||\bar{\boldsymbol{\zeta}}-\mathbb{1}||^2_2, \quad (6)$$

where α_1 and α_2 control the degree of weight deviation. To use the weighted alignment, we replace the corresponding loss in Eq. (5) by the alignment loss g defined in (6). Then, we additionally minimize (5) over $\bar{\boldsymbol{\zeta}}$ and a set $\{\boldsymbol{\zeta}_{r'}\}_{r'\in\mathcal{I}_r\setminus\{1\}}$ that determines contributions of tensors of order $r'=2,...,r$.

4.3. Kernelized Alignment Loss

Evaluating scatter tensors during the gradient descent is costly on $d=4096$ dim. feature vectors of *fc7*. Below we propose an efficient kernelization of the Frobenius norm.

Proposition 4. *The inner-product of scatter tensors $\boldsymbol{\mathcal{X}}^{(r)},\boldsymbol{\mathcal{Y}}^{(r)}\in\mathfrak{S}^d_{\times r}$ of order r from Eq. (1), can be written implicitly as a sum of entries of a polynomial kernel $\bar{\bar{\mathbf{K}}}^r\in\mathbb{R}^{N\times N^*}$, where $\bar{\bar{K}}^r_{nn'}=\langle\mathbf{x}_n-\boldsymbol{\mu},\mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r$, and $\mathbf{x}_n\in\mathbb{R}^d,\forall n\in\mathcal{I}_N$ and $\mathbf{y}_{n'}\in\mathbb{R}^d,\forall n'\in\mathcal{I}_{N^*}$ are some N and N^* feature vectors (that form $\boldsymbol{\mathcal{X}}^{(r)}$ and $\boldsymbol{\mathcal{Y}}^{(r)}$), $\boldsymbol{\mu}$ and $\boldsymbol{\mu}^*$ are their means. Then:*

$$\langle\boldsymbol{\mathcal{X}}^{(r)},\boldsymbol{\mathcal{Y}}^{(r)}\rangle=\frac{1}{NN^*}\sum_n\sum_{n'}\langle\mathbf{x}_n-\boldsymbol{\mu},\mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r=\frac{1}{NN^*}\mathbb{1}^T\bar{\bar{\mathbf{K}}}^r\mathbb{1}. \quad (7)$$

Proof. Substituting $\mathbf{x}_n-\boldsymbol{\mu}$ and $\mathbf{y}_{n'}-\boldsymbol{\mu}^*$ into Proposition 3, the proof follows. □

Proposition 5. *Suppose we have polynomial kernels $\mathbf{K}^r\in\mathbb{R}^{N\times N}$, $\bar{\mathbf{K}}^r\in\mathbb{R}^{N^*\times N^*}$ and $\bar{\bar{\mathbf{K}}}^r\in\mathbb{R}^{N\times N^*}$ defined as $K^r_{nn'}=\langle\mathbf{x}_n-\boldsymbol{\mu},\mathbf{x}_{n'}-\boldsymbol{\mu}\rangle^r$, $\bar{K}^r_{nn'}=\langle\mathbf{y}_n-\boldsymbol{\mu}^*,\mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r$ and $\bar{\bar{K}}^r_{nn'}=\langle\mathbf{x}_n-\boldsymbol{\mu},\mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r$, where $\mathbf{x}_n,\mathbf{y}_{n'},\boldsymbol{\mu},\boldsymbol{\mu}^*,N,N^*$ are defined as in Proposition 4. The Frobenius norm between two scatter tensors $\boldsymbol{\mathcal{X}}^{(r)},\boldsymbol{\mathcal{Y}}^{(r)}\in\mathfrak{S}^d_{\times r}$ of order r, which are defined in Eq. (1), can be expressed implicitly as:*

$$||\boldsymbol{\mathcal{X}}^{(r)}-\boldsymbol{\mathcal{X}}^{*(r)}||^2_F=\frac{1}{N^2}\mathbb{1}^T\mathbf{K}^r\mathbb{1}+\frac{1}{{N^*}^2}\mathbb{1}^T\bar{\mathbf{K}}^r\mathbb{1}-\frac{2}{NN^*}\mathbb{1}^T\bar{\bar{\mathbf{K}}}^r\mathbb{1}. \quad (8)$$

Proof. Combining Proposition 2 with 4, the proof follows. □

Derivatives of (8) are in Appendix B. Equation (8) can be evaluated on class-specific feature vectors and substituted directly into the loss functions in (5) and (6). This way, we obtain two different regimes for evaluating the Frobenius norm on the scatter tensors: one explicit and one kernelized; both exhibiting different strengths as detailed below.

Complexity. The Frobenius norm on the scatter tensors has complexity $\mathcal{O}((N+N^*+1)D)$, where $D=\binom{d+r-1}{r}$ as detailed in Proposition 1. The kernelized variant proposed above has complexity $\mathcal{O}((N^2+NN^*+N^*N^*)(d+\rho)$, where $\rho\leq\log r$ estimates the complexity of "rising x to the power of r". As $\rho\ll d$, its cost is negligible and can be safely left out from the above analysis.

It is easy to verify that, for the standard domain adaptation problems with $N=20$ source and $N^*=3$ target training points per class, $d=4096$ and $r=2$, explicit evaluations of the Frobenius norm are $\sim52\times$ slower than the proposed by us kernelized substitute. For the same scenario but with the scatter tensor of order $r=3$, explicit evaluations of the Frobenius norm are not tractable, as they take $\sim143000\times$ more time than the kernelized substitute, which demonstrates the clear benefit of our approach. The kernelization makes Eq. (6) tractable for $r>2$.

5. Experiments

In this section, we present experiments demonstrating the usefulness of our framework. We start by describing datasets we use in evaluations.

5.1. Datasets

Office dataset. A popular dataset for evaluating algorithms against the effect of domain shift is the Office dataset [32] which contains 31 object categories in three domains: Amazon, DSLR and Webcam. The 31 categories in the dataset consist of objects commonly encountered in office settings,

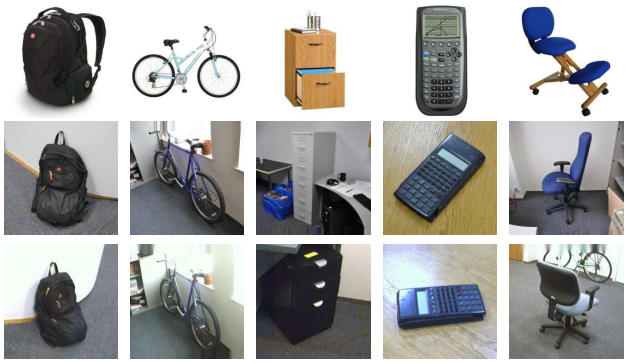


Figure 3: The Office dataset. Top, middle and bottom rows show examples from the Amazon, DSLR, and Webcam domains.

such as keyboards, file cabinets, and laptops. The Amazon domain contains on average 90 images per class and 2817 images in total. As these images were captured from a website of online merchants, they are captured against clean background and at a unified scale. The DSLR domain contains 498 low-noise high resolution images (4288×2848). There are 5 objects per category. Each object was captured from different viewpoints on average 3 times. For Webcam, the 795 images of low resolution (640×480) exhibit significant noise and color as well as white balance artifacts. Otherwise, 5 objects per category were also used in the capturing process. Figure 3 illustrates the three domains. We distinguish the following six domain shifts: Amazon-Webcam ($\mathcal{A} \rightarrow \mathcal{W}$), Amazon-DSLR ($\mathcal{A} \rightarrow \mathcal{D}$), Webcam-Amazon ($\mathcal{W} \rightarrow \mathcal{A}$), Webcam-DSLR ($\mathcal{W} \rightarrow \mathcal{D}$), DSLR-Amazon ($\mathcal{D} \rightarrow \mathcal{A}$) and DSLR-Webcam ($\mathcal{D} \rightarrow \mathcal{W}$).

We evaluate across 10 randomly chosen data splits per domain shift. We follow the standard protocol for this dataset and, for each training source split, we sample 20 images per category for the Amazon domain and 8 examples per category for the DSLR and Webcam domains. From the training target splits, we sample 3 images per class per split per domain. We present results for the supervised setting and report accuracies on the remaining target images, as the the standard protocol for this dataset suggests.

RGB-D-Caltech256 dataset. The RGB-D [25] and Caltech256 [11] datasets have been used as the source and target for evaluations of unsupervised domain adaptation problems [2, 30]. We use the 10 classes that are common between the two datasets *e.g.*, calculator, cereal box, coffee mug, ball, tomato. We use 50 and 5/10 images per class in the source and target domains for the supervised setting. We test on the remaining target samples. We report the mean average accuracy over 5 data splits, that is, we select randomly the source and target data samples for each split.

5.2. Experimental Setup

In each stream, we employ the AlexNet architecture [22] which was pre-trained on the ImageNet dataset [31] for the best results. At the training and testing time, we use the pipelines shown in Figures 2a and 2b, respectively. Where stated, we use the 16-layer VGG model [35] per stream to quantify the impact of different CNN models on our algorithm. We set non-zero learning rates on the fully-connected and the last two convolutional layers of the two streams.

On the RGB-D-Caltech256 dataset, we use the RGB images from Caltech256 as the target domain. In contrast to [2, 30] which use both the RGB data and depth maps as a source, we adapt our source stream based on AlexNet to use only the depth data from the RGB-D dataset – this helps us isolate performance of our algorithm in case of distinct heterogeneous domains. As these both domains are very different from each other, we apply two classifiers – one per network stream (see the footnote[1] in Section 4).

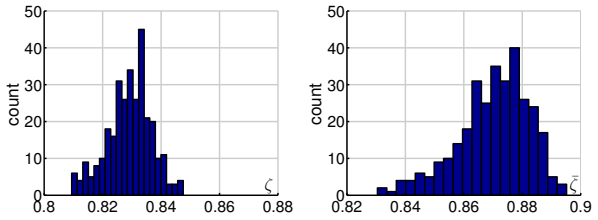
We evaluate Second- and/or Higher-order Transfer of Knowledge (*So-HoT*) approaches such as: unweighted and weighted second-order alignment losses (*So*) and (*So*+ζ), the third-order loss (*To*) and its weighted variant (*To*+ζ), and combined second- and third- (*So*+*To*+ζ) as well as fourth-order (*So*+*To*+*Fo*+ζ) weighted alignment losses. The model parameters were selected by cross-validation.

5.3. Comparison to the State of the Art

We apply our algorithm on the Office dataset. Table 1 presents results for the six domain shifts. Our second-order alignment loss (*So*) is compared against the baseline (*S*+*T*)

		$\mathcal{A}\rightarrow\mathcal{W}$	$\mathcal{A}\rightarrow\mathcal{D}$	$\mathcal{W}\rightarrow\mathcal{A}$	$\mathcal{W}\rightarrow\mathcal{D}$	$\mathcal{D}\rightarrow\mathcal{A}$	$\mathcal{D}\rightarrow\mathcal{W}$	acc.
DLID	[3]	51.9	-	-	89.9	-	78.2	73.33
DeCAF$_6$ S+T	[6]	80.7±2.3	-	-	-	-	94.8±1.2	87.75
DaNN	[9]	53.6±0.2	-	-	83.5±0.0	-	71.2±0.0	69.43
Source CNN	[40]	56.5±0.3	64.6±0.4	42.7±0.1	93.6±0.2	47.6±0.1	92.4±0.3	66.23
Target CNN	[40]	80.5±0.5	81.8±1.0	59.9±0.3	81.8±1.0	59.9±0.3	80.5±0.5	74.06
Source+Target CNN	[40]	82.5±0.9	85.2±1.1	65.2±0.7	96.3±0.5	65.8±0.5	93.9±0.5	81.48
Dom. Conf.+Soft Labs.	[40]	82.7±0.8	86.1±1.2	65.0±0.5	**97.6±0.2**	66.2±0.3	**95.7±0.5**	82.22
Ours: Source+Target CNN (*S*+*T*)		82.4±2.0	85.5±0.9	65.1±1.4	95.8±0.8	66.0±1.2	94.3±0.6	81.53
Ours: Second-order (*So*)		**84.5±1.7**	**86.3±0.8**	**65.7±1.7**	97.5±0.7	**66.5±1.0**	95.5±0.6	**82.68**

Table 1: Comparison of our second-order alignment loss (*So*) to the state of the art on the Office dataset.



(a) (b)

Figure 4: Histograms of ζ and $\bar{\zeta}$ weights in plots 4a and 4b, learned by (*So*+ζ) on $\mathcal{A}\rightarrow\mathcal{W}$, show levels of alignment of the scatter tensors and the means according to the loss in Eq. (6).

for which the source and target training samples were used together to fine-tune a standard CNN network. As can be seen, our method outperforms such a baseline as well as recent approaches such as *Domain Confusion with Soft Labels* and fine-tuning on the source or target data, respectively.

Performance on the VGG architecture. To evaluate effectiveness of our algorithm on other powerful networks, we follow the same pipeline as in Figure 2, except that we employ the pre-trained VGG [35] in place of AlexNet [22]. As VGG utilizes more parameters than AlexNet, we demonstrate in Table 2 that applying our second-order alignment loss (*So*) on $\mathcal{A}\rightarrow\mathcal{W}$ and $\mathcal{A}\rightarrow\mathcal{D}$ improves performance compared to the baselines (*S*+*T*) by 0.74% and 2.43%. Without resorting to data augmentations, we outperform *e.g.* a multi-scale multi-patch CNN approach [23] by 0.6% on $\mathcal{A}\rightarrow\mathcal{W}$.

Weighted vs. Unweighted Alignment. In this experiment, we demonstrate the benefit of using the weighted alignment of the scatter matrices and their means on the $\mathcal{A}\rightarrow\mathcal{W}$ and $\mathcal{A}\rightarrow\mathcal{D}$ domain shits. Table 2 shows that our weighted second- (*So*+ζ) and third-order (*To*+ζ) alignment losses, introduced in Eq. (6), improve over our unweighted second-order alignment loss (*So*) from Eq. (5) by 0.74% and 1.05% on $\mathcal{A}\rightarrow\mathcal{W}$, respectively. Learning ζ and $\bar{\zeta}$ can be implemented at no visible increase in computations.

In Figure 4, we show histograms of the ζ and $\bar{\zeta}$ weights from (*So*+ζ) over the 31 classes and the 10 splits. The histograms reveal that the levels of alignment of the scatter matrices and their means vary according to the Beta distributions. The means of these distributions are slightly below the desired mean value of one which indicates that, in this experiment, σ_1 and σ_2 from Eq. (6) were initialized with values larger than needed. Also, their optimal values might vary over time – learning weights compensates for this.

Alignment of combined Second-, Third- and Fourth-order Scatter Tensors. Our kernelized loss in Eq. (8) admits alignment between second- and/or higher-order scatter tensors which, beyond the scale/shear and orientation, capture higher-order statistical moments. In Table 2, we evaluate third-order weighted alignment loss (*To*+ζ), as well as combined second-, third- (*So*+*To*+ζ) and fourth-order (*So*+*To*+*Fo*+ζ) weighted alignments. As the order increases, the performance improves. For (*So*+*To*+*Fo*+ζ), we outperform (*So*) by 1.42% and 2.9% on $\mathcal{A}\rightarrow\mathcal{W}$ and $\mathcal{A}\rightarrow\mathcal{D}$.

Heterogeneous setting on RGB-D-Caltech256. In this experiment, we verify the behavior of our second-order alignment loss (*So*) w.r.t. the varying number of target training samples N^*. Figure 5 shows that the largest improvement of 1.24% and 1.04% over the baseline (*S*+*T*) is obtained for a small number $N^*=3$ and $N^*=5$, respectively.

6. Conclusions

We have presented an approach to domain adaptation by partial alignment of the within-class scatters to discover the commonality. The state-of-the-art results we obtain suggest that our simple strategy is effective despite challenges of domain adaptation. Moreover, the presented weighted approach and kernelized alignment loss improve the results and computational efficiency. Our method can be easily extended to multiple domains and other network architectures. Our supplementary material presents additional results.

Acknowledgements. We thank Dr. Hongping Cai and Dr. Peter Hall for our early discussions on the alignment loss. We thank NVIDIA for the donation of GPUs.

		sp1	sp2	sp3	sp4	sp5	sp6	sp7	sp8	sp9	sp10	aver. acc.
$\mathcal{A}\rightarrow\mathcal{W}$	*S*+*T*	91.5	87.9	91.5	89.6	89.0	87.2	86.2	87.2	91.2	86.5	88.76±1.9
	So	91.9	89.3	91.7	90.6	89.0	88.2	87.6	87.6	91.9	87.2	89.50±1.8
	So+ζ	92.4	89.9	90.5	92.2	88.9	88.2	90.0	89.5	91.3	89.6	**90.24±1.3**
	To+ζ	92.5	90.3	91.8	91.9	89.0	89.9	89.3	89.6	91.6	89.6	**90.55±1.1**
	So+*To*+ζ	92.6	90.5	92.0	92.0	89.2	90.0	89.6	89.9	91.8	89.9	**90.75±1.2**
	So+*To*+*Fo*+ζ	93.0	90.7	92.1	92.9	89.2	89.9	89.6	89.9	92.0	89.7	**90.92±1.3**
$\mathcal{A}\rightarrow\mathcal{D}$	*S*+*T*	90.6	88.9	89.4	92.4	90.1	87.2	91.1	88.2	90.9	89.4	89.83±1.4
	So	92.4	92.4	91.1	92.4	92.9	89.6	93.4	91.9	94.1	92.6	**92.26±1.1**
	So+*To*+ζ	92.7	92.9	91.6	92.5	93.3	89.7	93.7	91.9	94.0	93.0	**92.52±1.2**
	So+*To*+*Fo*+ζ	93.1	93.1	92.0	92.7	93.3	89.9	94.1	91.9	94.0	93.4	**92.73±1.1**

Table 2: The Office dataset on VGG streams. (Top) $\mathcal{A}\rightarrow\mathcal{W}$ and (Bottom) $\mathcal{A}\rightarrow\mathcal{D}$ domain shifts are evaluated on second-order (*So*), second- (*So*+ζ) and third-order+weights (*To*+ζ), second- and third- (*So*+*To*+ζ) and fourth-order (*So*+*To*+*Fo*+ζ) alignment with weight learning. Our baseline fine-tuning on the combined source and target domains (*S*+*T*) is also evaluated for comparison.

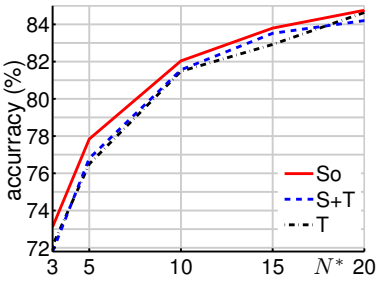


Figure 5: RGB-D-Caltech256. Second-order algorithm (*So*) vs. fine-tuning on (*S*+*T*) and target only (*T*). N^* is the number of target tr. samples per class.

Appendices

A. Derivatives of the alignment loss g w.r.t. the feature vectors

Suppose $\boldsymbol{\Phi}=[\boldsymbol{\phi}_1,...,\boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}^*=[\boldsymbol{\phi}_!^*,...,\boldsymbol{\phi}_N^*]$ are some feature vectors of quantity N and N^*, respectively, which are used to evaluate $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$. For $r=2$, we have to first compute the derivative of the covariance matrix $\boldsymbol{\Sigma}$ w.r.t. $\boldsymbol{\phi}_{m'n'}$. To do so, we proceed by computing derivatives of: i) the autocorrelation matrix in (9) and ii) the outer product of means $\boldsymbol{\mu}$ in (10) and (11):

$$\frac{\partial \sum_n \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T}{\partial \phi_{m'n'}} = \boldsymbol{j}_{m'} \boldsymbol{\phi}_{n'}^T + \boldsymbol{\phi}_{n'} \boldsymbol{j}_{m'}^T, \tag{9}$$

$$\frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \mu_{m'}} = \boldsymbol{j}_{m'} \boldsymbol{\mu}^T + \boldsymbol{\mu} \boldsymbol{j}_{m'}^T, \tag{10}$$

$$\frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \phi_{m'n'}} = \sum_m \frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \mu_m} \frac{\partial \mu_m}{\partial \phi_{m'n'}} = \frac{1}{N} \left(\boldsymbol{j}_{m'} \boldsymbol{\mu}^T + \boldsymbol{\mu} \boldsymbol{j}_{m'}^T \right), \tag{11}$$

where $\boldsymbol{j}_{m'}$ is a vector of zero entries except for position m' which is equal one. Putting together (9), (10) and (11) yields the derivative of $\boldsymbol{\Sigma}$ w.r.t. $\boldsymbol{\phi}_{m'n'}$:

$$\frac{\partial \left(\frac{1}{N} \sum_n \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \right) - \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \phi_{m'n'}} = \frac{1}{N} \left(\boldsymbol{j}_{m'} \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right)^T + \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right) \boldsymbol{j}_{m'}^T \right). \tag{12}$$

The derivatives of $||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2$ w.r.t. covariance $\boldsymbol{\Sigma}$ as well as $\boldsymbol{\phi}_{m'n'}$ and $\boldsymbol{\phi}_{m'n'}^*$ are provided below:

$$\frac{\partial ||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial \boldsymbol{\Sigma}} = 2 \left(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^* \right) \tag{13}$$

$$\begin{aligned} \frac{\partial ||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial \phi_{m'n'}} &= \sum_{m,n} \frac{\partial ||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial \Sigma_{mn}} \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_{m'n'}} \right)_{mn} \\ &= \frac{2}{N} \sum_{m,n} (\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^*)_{mn} \left(\boldsymbol{j}_{m'} \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right)^T + \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right) \boldsymbol{j}_{m'}^T \right)_{mn} \\ &= \frac{4}{N} \left(\boldsymbol{\Sigma}_{m',:} - \boldsymbol{\Sigma}_{m',:}^* \right) \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right). \end{aligned} \tag{14}$$

The derivatives of $||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2$ w.r.t. $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ are:

$$\frac{\partial ||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial \boldsymbol{\Phi}} = \frac{4}{N} (\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^*) \left(\boldsymbol{\Phi} - \boldsymbol{\mu} \mathbb{1}^T \right), \tag{15}$$

$$\frac{\partial ||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial \boldsymbol{\Phi}^*} = -\frac{4}{N^*} (\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^*) \left(\boldsymbol{\Phi}^* - \boldsymbol{\mu}^* \mathbb{1}^T \right). \tag{16}$$

The derivatives of $||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2$ w.r.t. $\boldsymbol{\mu}$, $\boldsymbol{\phi}_n$ and $\boldsymbol{\phi}_{n'}^*$ are:

$$\frac{\partial ||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial \boldsymbol{\mu}} = 2 \left(\boldsymbol{\mu} - \boldsymbol{\mu}^* \right), \tag{17}$$

$$\frac{\partial ||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial \boldsymbol{\phi}_{n'}} = \frac{2 \left(\boldsymbol{\mu} - \boldsymbol{\mu}^* \right)}{N}, \quad \frac{\partial ||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial \boldsymbol{\phi}_{n'}^*} = \frac{2 \left(\boldsymbol{\mu} - \boldsymbol{\mu}^* \right)}{N^*}. \tag{18}$$

B. Kernelized derivative of the Frobenius norm between tensors w.r.t. the feature vectors

Suppose that some feature vectors $\boldsymbol{\Phi}=[\boldsymbol{\phi}_1,...,\boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}^*=[\boldsymbol{\phi}_!^*,...,\boldsymbol{\phi}_N^*]$ are given in quantities N and N^* and that the Frobenius norm between tensors $\boldsymbol{\mathcal{X}}^{(r)}$ and $\boldsymbol{\mathcal{Y}}^{(r)}$ of order $r \geq 1$ build from $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ is being evaluated. Then, the derivative of Equation (8) w.r.t. feature vector $\boldsymbol{\phi}_{n^\ddagger}$ becomes:

$$\begin{aligned} \frac{\partial ||\boldsymbol{\mathcal{X}}^{(r)} - \boldsymbol{\mathcal{X}}^{*(r)}||_F^2}{\partial \boldsymbol{\phi}_{n^\ddagger}} &= \frac{1}{N^2} r \sum_{n=1}^{N} \sum_{n'=1}^{N} K_{nn'}^{r-1} \frac{\partial K_{nn'}}{\partial \boldsymbol{\phi}_{n^\ddagger}} \\ &\quad - \frac{2}{NN^*} r \sum_{n=1}^{N} \sum_{n'=1}^{N^*} \bar{\bar{K}}_{nn'}^{r-1} \frac{\partial \bar{\bar{K}}_{nn'}}{\partial \boldsymbol{\phi}_{n^\ddagger}}, \end{aligned} \tag{19}$$

where

$$\begin{aligned} \frac{\partial K_{nn'}}{\partial \boldsymbol{\phi}_{n^\ddagger}} &= \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'} \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\phi}_{n'} \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\mu} \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} + \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\mu} \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} \\ &= \begin{cases} \boldsymbol{\phi}_{n'} - \frac{\boldsymbol{\phi}_{n'}}{N} - (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_n}{N}) & , \text{if } n = n^\ddagger, n' \neq n^\ddagger \\ \boldsymbol{\phi}_n - (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_{n'}}{N}) - \frac{\boldsymbol{\phi}_n}{N} + \frac{2}{N}\boldsymbol{\mu} & , \text{if } n \neq n^\ddagger, n' = n^\ddagger \\ 2\boldsymbol{\phi}_n - (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_{n'}}{N}) - (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_n}{N}) & , \text{if } n = n^\ddagger, n' = n^\ddagger \\ 0 - \frac{\boldsymbol{\phi}_{n'}}{N} - \frac{\boldsymbol{\phi}_n}{N} & , \text{if } n \neq n^\ddagger, n' \neq n^\ddagger, \end{cases} \end{aligned} \tag{20}$$

$$\begin{aligned} \frac{\partial \bar{\bar{K}}_{nn'}}{\partial \boldsymbol{\phi}_{n^\ddagger}} &= \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'}^* \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\phi}_{n'}^* \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\mu}^* \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} + \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\mu}^* \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} \\ &= \begin{cases} \boldsymbol{\phi}_{n'}^* - \boldsymbol{\mu}^* & , \text{if } n = n^\ddagger, n' \neq n^\ddagger \\ 0 - 0 & , \text{if } n \neq n^\ddagger, n' = n^\ddagger \\ \boldsymbol{\phi}_n^* - \boldsymbol{\mu}^* & , \text{if } n = n^\ddagger, n' = n^\ddagger \\ 0 - \frac{1}{N}\boldsymbol{\phi}_{n'}^* - \frac{1}{N}\boldsymbol{\mu}^* & , \text{if } n \neq n^\ddagger, n' \neq n^\ddagger. \end{cases} \end{aligned} \tag{21}$$

Putting together Equations (19), (20) and (21) and setting $q=r-1$ yields the derivatives w.r.t. matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$:

$$\begin{aligned} \frac{\partial ||\boldsymbol{\mathcal{X}}^{(r)} - \boldsymbol{\mathcal{X}}^{*(r)}||_F^2}{\partial \boldsymbol{\Phi}} &= \frac{2}{N^2} r \boldsymbol{\Phi} \left(\mathbf{K}^{qT} - \frac{1}{N} (\mathbb{1}^T \mathbf{K}^q)^T \mathbb{1}^T \right) \\ &+ \frac{2r\boldsymbol{\mu}}{N^2} \left(\frac{1}{N} \mathbb{1}^T \mathbf{K}^q \mathbb{1} - \mathbb{1}^T \mathbf{K}^q \right) - \frac{2r\boldsymbol{\Phi}^*}{NN^*} \left(\bar{\bar{\mathbf{K}}}^{qT} - \frac{1}{N} (\bar{\bar{\mathbf{K}}}^{qT} \mathbb{1}) \mathbb{1}^T \right) \\ &+ \frac{2}{NN^*} r \boldsymbol{\mu}^* \left(\mathbb{1}^T \bar{\bar{\mathbf{K}}}^{qT} - \frac{1}{N} \mathbb{1}^T \bar{\bar{\mathbf{K}}}^{qT} \mathbb{1} \right) \end{aligned} \tag{22}$$

and

$$\begin{aligned} \frac{\partial ||\boldsymbol{\mathcal{X}}^{(r)} - \boldsymbol{\mathcal{X}}^{*(r)}||_F^2}{\partial \boldsymbol{\Phi}^*} &= \frac{2}{N^{*2}} r \boldsymbol{\Phi}^* \left(\bar{\bar{\mathbf{K}}}^{qT} - \frac{1}{N^*} (\mathbb{1}^T \bar{\bar{\mathbf{K}}}^q)^T \mathbb{1}^T \right) \\ &+ \frac{2r\boldsymbol{\mu}^*}{N^{*2}} \left(\frac{1}{N^*} \mathbb{1}^T \bar{\bar{\mathbf{K}}}^q \mathbb{1} - \mathbb{1}^T \bar{\bar{\mathbf{K}}}^q \right) - \frac{2r\boldsymbol{\Phi}}{NN^*} \left(\bar{\bar{\mathbf{K}}}^q - \frac{1}{N^*} (\bar{\bar{\mathbf{K}}}^q \mathbb{1}) \mathbb{1}^T \right) \\ &+ \frac{2}{NN^*} r \boldsymbol{\mu} \left(\mathbb{1}^T \bar{\bar{\mathbf{K}}}^q - \frac{1}{N^*} \mathbb{1}^T \bar{\bar{\mathbf{K}}}^q \mathbb{1} \right). \end{aligned} \tag{23}$$

References

[1] J. Baxter, R. Caruana, T. Mitchell, L. Y. Pratt, D. L. Silver, and S. Thrun. Learning to learn: Knowledge consolidation and transfer in inductive systems. NIPS Workshop, http://plato.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop.1995.html, 1995. Accessed: 30-10-2016. 1

[2] L. Chen, W. Li, and D. Xu. Recognizing rgb images by learning from rgb-d data. *CVPR*, 2014. 6

[3] S. Chopra, S. Balakrishnan, and R. Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. *ICML Workshop*, 2013. 2, 3, 5, 6

[4] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *JMLR*, 9:1757–1774, 2008. 2

[5] H. Daumé, III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59, 2010. 2

[6] 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. *ICML*, 2014. 6

[7] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. 2

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016. 2, 3

[9] M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. *CoRR*, abs/1409.6041, 2014. 6

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, pages 580–587, 2014. 2

[11] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 6

[12] M. Harandi and B. Fernando. Generalized backpropagation, étude de cas: Orthogonality. *CoRR*, abs/1611.05927, 2016. 2

[13] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4 – 21, 2017. Regularization Techniques for High-Dimensional Data Analysis. 2

[14] S. Herath, M. T. Harandi, and F. Porikli. Learning an invariant hilbert space for domain adaptation. *CVPR*, 2017. 2

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[16] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982. 2

[17] N. Intrator and S. Edelman. Making a low-dimensional representation suitable for diverse tasks. *Connection Sci.*, 8(2):205–223, 1996. 2

[18] T.-K. Kim, K.-Y. K. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. *CVPR*, 2007. 3

[19] P. Koniusz and A. Cherian. Sparse coding for third-order super-symmetric tensor descriptors with application to texture recognition. *CVPR*, 2016. 3, 4

[20] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3D skeletons. *ECCV*, 2016. 3

[21] P. Koniusz, F. Yan, P. Gosselin, and K. Mikolajczyk. Higher-order occurrence pooling for bags-of-words: Visual concept detection. *PAMI*, 2016. 3, 4

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *NIPS*, pages 1106–1114, 2012. 1, 2, 6, 7

[23] I. Kuzborskij, F. M. Carlucci, and B. Caputo. When naïve bayes nearest neighbors meet convolutional neural networks. *CVPR*, 2016. 2, 3, 7

[24] R. P. L. Fei-Fei; Fergus. One-shot learning of object categories. *TPAMI*, 28:594–611, April 2006. 2

[25] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. *ICRA*, pages 1817–1824, 2011. 6

[26] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Analysis and Applications*, 21:1253–1278, 2000. 4

[27] W. Li, T. Tommasi, F. Orabona, D. Vázquez, M. López, J. Xu, and H. Larochelle. Task-cv: Transferring and adapting source knowledge in computer vision. ECCV Workshop, http://adas.cvc.uab.es/task-cv2016, 2016. Accessed: 22-11-2016. 1

[28] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. *ICML Workshop*, pages 97–105, 2015. 2

[29] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011. 3

[30] S. Motiian and G. Doretto. Information bottleneck domain adaptation with privileged information for visual recognition. *ECCV*, 2016. 6

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 6

[32] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *ECCV*, pages 213–226, 2010. 5

[33] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014. 2

[34] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. *ICML*, 2005. 3

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, abs/1409.1556, 2015. 2, 6, 7

[36] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. *CoRR*, abs/1511.05547, 2015. 2, 3

[37] S. Thrun. Is learning the n-th thing any easier than learning the first? *NIPS*, pages 640–646, 1996. 2

[38] T. Tommasi, M. Lanzi, P. Russo, and B. Caputo. Learning the roots of visual domain shift. *ECCV Workshop*, 2016. 2, 3

[39] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers:

Learning categories from few examples with multi model knowledge transfer. *CVPR*, pages 3081–3088, 2010. 1, 2

[40] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. *ICCV*, pages 4068–4076, 2015. 2, 6

[41] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(56):544–557, 2009. 1

[42] M. A. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. *ECCV*, 2002. 3

[43] M. A. Vasilescu and D. Terzopoulos. Tensortextures: multilinear image-based rendering. *ACM Transactions on Graphics*, 23(3):336–342, 2004. 3

[44] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. *ECCV*, 2016. 2, 3, 5

[45] J. West, D. Venture, and S. Warnick. Spring research presentation: A theoretical foundation for inductive transfer. *Brigham Young University, College of Physical and Mathematical Sciences*, 2007. 1

[46] R. S. Woodworth and E. L. Thorndike. The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review (I)*, 8(3):247–261, 1901. 1

[47] Y.-R. Yeh, C.-H. Huang, and Y.-C. F. Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *Transactions on Image Processing*, 23(5), 2014. 2, 3