This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

When Unsupervised Domain Adaptation Meets Tensor Representations*

Hao Lu[†], Lei Zhang[‡], Zhiguo Cao[†], Wei Wei[‡], Ke Xian[†], Chunhua Shen[§], Anton van den Hengel[§]
[†]School of Automation, Huazhong Unviersity of Science and Technology, China
[‡]School of Computer Science and Engineering, Northwestern Polytechnical University, China
[§]School of Computer Science, The University of Adelaide, Australia
{poppinace,zgcao}@hust.edu.cn

Abstract

Domain adaption (DA) allows machine learning methods trained on data sampled from one distribution to be applied to data sampled from another. It is thus of great practical importance to the application of such methods. Despite the fact that tensor representations are widely used in Computer Vision to capture multi-linear relationships that affect the data, most existing DA methods are applicable to vectors only. This renders them incapable of reflecting and preserving important structure in many problems. We thus propose here a learning-based method to adapt the source and target tensor representations directly, without vectorization. In particular, a set of alignment matrices is introduced to align the tensor representations from both domains into the invariant tensor subspace. These alignment matrices and the tensor subspace are modeled as a joint optimization problem and can be learned adaptively from the data using the proposed alternative minimization scheme. Extensive experiments show that our approach is capable of preserving the discriminative power of the source domain, of resisting the effects of label noise, and works effectively for small sample sizes, and even one-shot DA. We show that our method outperforms the state-of-the-art on the task of cross-domain visual recognition in both efficacy and efficiency, and particularly that it outperforms all comparators when applied to DA of the convolutional activations of deep convolutional networks.

1. Introduction

The difficulty of securing an appropriate and exhaustive set of training data, and the tendency for the domain of application to drift over time, often lead to variations between the distributions of the training (source) and test (target) data. In Machine Learning this problem is labeled *domain mismatch*. Failing to model such a distribution shift may cause significant performance degradation. Domain adaptation (DA) techniques capable of addressing this problem of distribution shift have thus received significant attention recently [23].

The assumption underpinning DA is that, although the domains differ, there is sufficient commonality to support adaptation. Many approaches have modeled this commonality by learning an invariant subspace, or set of subspaces [1, 10, 12, 13]. These methods are applicable to vector data only, however. Applying these methods to structured high-dimensional representations (e.g., convolutional activations), thus requires that the data be vectorized first. Although this solves the algebraic issue, it does not solve the underlying problem.

Tensor arithmetic is a generalization of matrix and vector arithmetic, and is particularly well suited to representing multi-linear relationships that neither vector nor matrix algebra can capture naturally [33]. The higher-order statistics of a vector-valued random variables are most naturally expressed as tensors, for instance. The power of tensor representations has also been demonstrated for a range of computer vision tasks (see Section 2 for examples). Deep convolutional neural networks (CNNs) [18] represent the state-of-the-art method for a substantial number of visual tasks [15, 20, 24], which makes DA a critical issue for their practical application. The activations of such CNNs, and the interactions between them, are naturally represented as tensors, meaning that DA should also be applied using this representation. We show in Section 5 that the proposed method outperforms all comparators in DA of the convolutional activations of CNNs.

Vectorization also often results in the so-called *curse of dimensionality* [27], as the matrices representing the relationships between vectorized tensors have n^2 elements, where n is the number of elements in the tensor. This leads to errors in the estimation of this large number of parameters and high computational complexity. Furthermore, after

*H. Lu and L. Zhang contributed equally. This work was done when H. Lu, L. Zhang, and K. Xian were visiting The University of Adelaide. Z. Cao is the corresponding author.

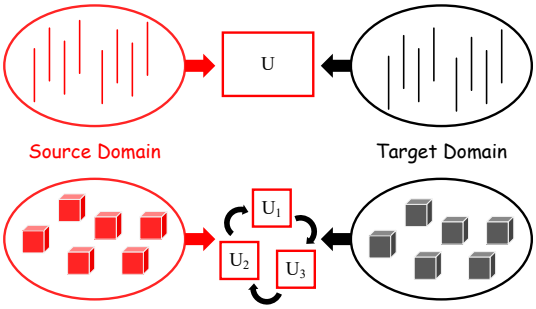


Figure 1: Vector subspace (top) vs. tensor subspace (bottom). Third-order (3-mode) tensors are used as an example. Compared to the vector subspace, the tensor subspace consists of a set of subspaces characterizing each mode respectively. Higher-order tensor modeling offers us an opportunity to investigate multiple interactions and couplings that capture the commonality and differences between domains.

vectorization, many existing approaches become sensitive to the scarcity of source data (compared to the number of dimensions) and noise in the labels. The proposed direct tensor method uses much lower dimensional entities, thus avoiding these estimation problems.

To address these issues we propose to learn an invariant tensor subspace that is able to adapt the tensor representations directly. The key question is thus whether we can *find an invariant tensor subspace such that the domain discrepancy is reduced when the source data are adapted into the target domain*. Following this idea, a novel approach termed Tensor-Aligned Invariant Subspace Learning (TAISL) is proposed for unsupervised DA. By introducing a set of alignment matrices, the tensor representations from the source domain are aligned to an underlying tensor subspace shared by the target domain. As illustrated in Fig. 1, the tensor subspace is able to preserve the intrinsic structure of representations by modeling the correlation between different modes. Instead of executing a holistic adaptation (where all feature dimensions would be taken into account), our approach performs mode-wise partial adaptation (where each mode is adapted separately) to avoid the curse of dimensionality. Seeking such a tensor subspace and learning the alignment matrices are consequently formulated into a joint optimization problem. We also propose an alternating minimization scheme, which allows the problem to be effectively optimized by off-the-shelf solvers.

Extensive experiments on cross-domain visual recognition demonstrate the following merits of our approach: i) it effectively reduces the domain discrepancy and preserves the discriminative power of the original representations; ii) it is applicable to small-sample-size adaptation, even when there is only one source sample per category; iii) it is robust to noisy labels; iv) it is computationally efficient, because the tensor subspace is constructed in a much smaller space than the vector-form paradigm; and v) it shows superior performance over state-of-the-art vector representation-based approaches in both the classification accuracy and computation time. Source code is made available online at: `https://github.com/poppinace/TAISL`.

2. Related work

Our work is closely related to subspace-based unsupervised DA and tensor representations.

Subspace-based domain adaptation. Gopalan *et al*. [13] present one of the first visual DA approaches, which samples a finite set of subspaces along geodesic flows to bridge the source and target domains. Later in [12], Gong *et al*. kernelize this idea by integrating an infinite number of subspaces that encapsulate the domain commonness and difference in a smooth and compact manner. Recently, [10] argues that it is sufficient to directly align the subspaces of two domains using a linear projection. Intuitively, such a linear mapping defines a shift of viewing angle that snapshots the source data from the target perspective. Subsequently, [1] extends [10] in a landmark-based kernelized paradigm. The performance improvement is due to the nonlinearity of the Gaussian kernel and sample reweighting. Alternatively, [28] imposes a low-rank constraint during the subspace learning to reconstruct target samples with relevant source samples. More recently, [30] proposes to use the covariance matrix, a variant of the subspace, to characterize the domain, the adaptation is then cast as two simple but effective procedures of whitening the source data and recoloring the target covariance.

Tensor representations. Tensor representations play a vital role in many computer vision applications [16, 18, 19, 32]. At the early stage of face representations, [32] introduced the idea of "tensorfaces" to jointly model multiple variations (viewpoint, expression, illumination, etc.). [19] achieves robust visual tracking by modeling frame-wise appearance using tensors. [16] proposes tensor-based canonical correlation analysis as a representation for action recognition and detection. In other low-level tasks, such as image inpainting and image synthesis [40], modeling images as a tensor is also a popular choice.

More recently, the most notable example is the deep CNNs [18], as convolutional activations are intrinsically represented as tensors. The state-of-the-art performance of generic visual recognition and semantic image segmentation benefits from fully-convolutional models [15, 20]. Aside from this, by reusing convolutional feature maps, proposal generation and object detection can be performed simultaneously in a faster R-CNN fashion [24]. Yet, convolutional activations still suffer from the domain shift [21, 37]. How to adapt convolutional activations effectively remains an open question.

Tensor representations are important, while solutions to adapt them are limited. To fill this gap, we present one of the first DA approaches for tensor representations adaptation.

3. Learning an invariant tensor subspace

Before we present our technical details, some mathematical background related to tensor decomposition is provided. In the following mathematical expressions, we denote matrices and tensors by uppercase boldface letters and calligraphic letters, respectively, such as $\boldsymbol{U}$ and $\mathcal{U}$.

3.1. Tensor decomposition revisited

A tensor of order (mode) K is denoted by $\mathcal{X} \in \mathbb{R}^{n_1 \times \ldots \times n_K}$. Its `mode-`$k$ `product` is defined as $\mathcal{X} \times_k \boldsymbol{V}$. The operator $\times_k$ indicates matrix multiplication performed along the k-th mode. Equivalently, $(\mathcal{X} \times_k \boldsymbol{V})_{(k)} = \boldsymbol{V}\boldsymbol{X}_{(k)}$, where $\boldsymbol{X}_{(k)}$ is called the `mode-`k `matrix unfolding`, a procedure of reshaping a tensor $\mathcal{X}$ into a matrix $\boldsymbol{X}_{(k)} \in \mathbb{R}^{n_k \times n_1 \ldots n_{k-1} n_{k+1} \ldots n_K}$.

In this paper we draw upon *Tucker Decomposition* [17] to generate tensor subspaces. Tucker decomposition decomposes a K-mode tensor $\mathcal{X}$ into a core tensor multiplied by a set of factor matrices along each mode as follows:

$$\mathcal{X} = \mathcal{G} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \cdots \times_K \boldsymbol{U}^{(K)} = [\![\mathcal{G}; \mathcal{U}]\!] \ , \quad (1)$$

where $\mathcal{G} \in \mathbb{R}^{d_1 \times \ldots \times d_K}$ is the core tensor, and $\boldsymbol{U}^{(k)} \in \mathbb{R}^{n_k \times d_k}$ denotes the factor matrix of the k-th mode. The column space of $\boldsymbol{U}^{(k)}$ expands the corresponding signal subspace. To simply the notation, with $\mathcal{U} = \{\boldsymbol{U}^{(k)}\}_{k=1,\ldots,K}$, Tucker decomposition can be concisely represented as the right part of Eq. 1. Here, $\mathcal{U}$ is the tensor subspace, and $\mathcal{G}$ is the tensor subspace representation of $\mathcal{X}$. Alternatively, via the Kronecker product, Tucker decomposition can be expressed in matrix form as $\boldsymbol{X}_{(k)} = \boldsymbol{U}^{(k)} \boldsymbol{G}_{(k)} \boldsymbol{U}_{\setminus k}^T$, where

$$\boldsymbol{U}_{\setminus k} = \boldsymbol{U}^{(K)} \otimes \cdots \otimes \boldsymbol{U}^{(k+1)} \otimes \boldsymbol{U}^{(k-1)} \otimes \cdots \otimes \boldsymbol{U}^{(1)}, \quad (2)$$

and $\otimes$ denotes the Kronecker product.

3.2. Naive tensor subspace learning

Perhaps the most straight-forward way to adapt domains is to assume an invariant subspace between the source domain S and the target domain T. This assumption is reasonable when the domain discrepancy is not very large. With this idea, we first introduce the Naive Tensor Subspace Learning (NTSL) below, which can be viewed as a baseline of our approach.

Given N_s samples $\{\mathcal{X}_s^n\}_{n=1,\ldots,N_s}$ from source domain, each sample is denoted as a K-mode tensor $\mathcal{X}_s^n \in \mathbb{R}^{n_1 \times \ldots \times n_K}$. For simplicity, N_s samples are stacked into a $(K+1)$-mode tensor $\mathcal{X}_s \in \mathbb{R}^{n_1 \times \ldots \times n_K \times N_s}$. Similarly, let $\mathcal{X}_t \in \mathbb{R}^{m_1 \times \ldots \times m_K \times N_t}$ be a set of N_t samples from the target domain T. In general, we consider $n_k = m_k, k = 1, 2, ..., K$, because the case with heterogeneous data is out the scope of this paper. Provided that S and T share a underlying tensor subspace $\mathcal{U} = \{\boldsymbol{U}^{(k)}\}_{k=1,\ldots,K}, \boldsymbol{U}^{(k)} \in \mathbb{R}^{n_k \times d_k}$, on the basis of Tucker decomposition, seeking $\mathcal{U}$ is equivalent to solve the following optimization problem as

$$\begin{aligned} \min_{\mathcal{U}, \mathcal{G}_s, \mathcal{G}_t} & \|\mathcal{X}_s - [\![\mathcal{G}_s; \mathcal{U}]\!]\|_F^2 + \|\mathcal{X}_t - [\![\mathcal{G}_t; \mathcal{U}]\!]\|_F^2 \\ \text{s.t. } & \forall k, \ \boldsymbol{U}^{(k)T} \boldsymbol{U}^{(k)} = \boldsymbol{I} \end{aligned} \ , \quad (3)$$

where $\mathcal{G}_s$ and $\mathcal{G}_t$ denote the tensor subspace representation of $\mathcal{X}_s$ and $\mathcal{X}_t$, respectively. $\boldsymbol{I}$ is an identity matrix with appropriate size. Here $\mathcal{U}$ is the invariant tensor subspace in which the idea of DA lies. One can employ the off-the-shelf Tucker decomposition algorithm to solve Eq. (3) effectively. Once the optimum $\mathcal{U}^*$ is identified, $\mathcal{G}_s$ can be obtained by the following straight-forward multilinear product as

$$\mathcal{G}_s = \mathcal{X}_s \times_1 \boldsymbol{U}^{*(1)T} \times_2 \boldsymbol{U}^{*(2)T} \times_3 \cdots \times_K \boldsymbol{U}^{*(K)T}. \quad (4)$$

A similar procedure can be applied to derive $\mathcal{G}_t$. Next, if DA is evaluated in the context of classification, one can learn a linear classifier with $\mathcal{G}_s$ and source label L_s, and then verifies the classification performance on $\mathcal{G}_t$.

3.3. Tensor-aligned invariant subspace learning

Eq. (3) assumes a shared subspace between two domains. However, when the domain discrepancy becomes larger, enforcing only a shared subspace is typically not sufficient. To address this, we present Tensor-Aligned Invariant Subspace Learning (TAISL) which aims to reduce the domain discrepancy more explicitly. Motivated by the idea that a simple linear transformation can effectively reduce the domain discrepancy [2, 10], we introduce a set of alignment matrices into Eq. (3). This yields the following optimization problem as

$$\begin{aligned} \min_{\mathcal{U}, \mathcal{G}_s, \mathcal{G}_t, \mathcal{M}} & \|[\![\mathcal{X}_s; \mathcal{M}]\!] - [\![\mathcal{G}_s; \mathcal{U}]\!]\|_F^2 + \|\mathcal{X}_t - [\![\mathcal{G}_t; \mathcal{U}]\!]\|_F^2 \\ \text{s.t. } & \forall k, \ \boldsymbol{U}^{(k)T} \boldsymbol{U}^{(k)} = \boldsymbol{I} \end{aligned} \ , \quad (5)$$

where $\mathcal{M} = \{\boldsymbol{M}^{(k)}\}_{k=1,\ldots,K}, \boldsymbol{M}^{(k)} \in \mathbb{R}^{m_k \times n_k}$. With $\mathcal{M}$, samples from S can be linearly aligned to T. Here, $\boldsymbol{M}^{(k)}$ is unconstrained, which is undesirable in a well-defined optimization problem. To narrow down the search space, a natural choice to regularize $\boldsymbol{M}^{(k)}$ is the Frobenius norm $\|\boldsymbol{M}^{(k)}\|_F^2$. However, [22] suggests that the original data variance should be preserved after the alignment. Otherwise, there is a high probability the projected data will cluster into a single point. As a consequence, we employ a PCA-like constraint on $\mathcal{M}$ to maximally preserve the data variance. This gives our overall optimization problem

$$\begin{aligned} \min_{\mathcal{U}, \mathcal{G}_s, \mathcal{G}_t, \mathcal{M}} & \|[\![\mathcal{X}_s; \mathcal{M}]\!] - [\![\mathcal{G}_s; \mathcal{U}]\!]\|_F^2 + \|\mathcal{X}_t - [\![\mathcal{G}_t; \mathcal{U}]\!]\|_F^2 \\ & + \lambda \|[\![[\![\mathcal{X}_s; \mathcal{M}]\!]; \mathcal{M}^T]\!] - \mathcal{X}_s\|_F^2 \\ \text{s.t. } & \forall k, \ \boldsymbol{U}^{(k)T} \boldsymbol{U}^{(k)} = \boldsymbol{I}, \boldsymbol{M}^{(k)} \boldsymbol{M}^{(k)T} = \boldsymbol{I} \end{aligned} \ , \quad (6)$$

where λ is a weight on the regularization term. Intuitively, the regularization term measures how well $\mathcal{M}$ reconstructs the source data. Note that, in contrast $\boldsymbol{U}^{(k)}$, which is column-wise orthogonal, $\boldsymbol{M}^{(k)}$ is row-wise orthogonal. Moreover, both $\boldsymbol{U}^{(k)}$ and $\boldsymbol{M}^{(k)}$ have no effect

on the $(K+1)$-th mode, because the adaptation of data dimension makes no sense.

Relation to subspace alignment. As mentioned in Section 2, the seminal subspace alignment (SA) framework is introduced in [10]. Given two vector subspaces $\boldsymbol{U}_s$ and $\boldsymbol{U}_t$ of two domains, the domain discrepancy is measured by the Bregman divergence as $\|\boldsymbol{U}_s\boldsymbol{M} - \boldsymbol{U}_t\|_F^2$. Here $\boldsymbol{M}$ aligns the subspaces. In our formulation, $\mathcal{M}$ seems to align the data directly at the first glance. However, if one takes the properties of the mode-k product into account, one can see that this is not the case. According to the definition of the Tucker decomposition, for $\mathcal{X}_s$, we have $\mathcal{X}_s = \mathcal{G}_s \times_1 \boldsymbol{U}_s^{(1)} \times_2 \cdots \times_K \boldsymbol{U}_s^{(K)}$, so $[\![\mathcal{X}_s; \mathcal{M}]\!]$ can be expanded as

$$\begin{aligned} &\mathcal{X}_s \times_1 \boldsymbol{M}^{(1)} \times_2 \cdots \times_K \boldsymbol{M}^{(K)} \\ =\ &\mathcal{G}_s \times_1 (\boldsymbol{M}^{(1)}\boldsymbol{U}_s^{(1)}) \times_2 \cdots \times_K (\boldsymbol{M}^{(K)}\boldsymbol{U}_s^{(K)}) \end{aligned}. \tag{7}$$

That is, the alignment of the tensor is equivalent to the alignment of the tensor subspace. As a consequence, our approach can be viewed as a natural generalization of [10] to the multidimensional case. However, unlike SA, in which the DA and subspaces are learned separately, the alignment matrices $\mathcal{M}$ and the tensor subspace $\mathcal{U}$ in our approach are learned jointly in an unified paradigm.

4. Optimization

Here we discuss how to solve the problem in Eq. (6). Since $\mathcal{M}$ and $\mathcal{U}$ are coupled in Eq. (6), it is hard for a joint optimization. A general strategy is to use alternative minimization to decompose the problem into subproblems and to iteratively optimize these subproblems until convergence, acquiring an approximate solution [28, 38, 39].

Optimize $\mathcal{U}$, $\mathcal{G}_s$, and $\mathcal{G}_t$ given $\mathcal{M}$: By introducing an auxiliary variable $\mathcal{Z} = [\![\mathcal{X}_s; \mathcal{M}]\!]$, the subproblem over $\mathcal{U}$, $\mathcal{G}_s$ and $\mathcal{G}_t$ can be given as

$$\begin{aligned} &\min_{\mathcal{U},\mathcal{G}_s,\mathcal{G}_t} \|\mathcal{Z} - [\![\mathcal{G}_s; \mathcal{U}]\!]\|_F^2 + \|\mathcal{X}_t - [\![\mathcal{G}_t; \mathcal{U}]\!]\|_F^2 \\ &\text{s.t. } \forall k,\ \boldsymbol{U}^{(k)T}\boldsymbol{U}^{(k)} = \boldsymbol{I} \end{aligned}, \tag{8}$$

which is exactly the same problem in Eq. (3) and can be easily solved in the same paradigm.

Optimize $\mathcal{M}$ given $\mathcal{U}$, $\mathcal{G}_s$, and $\mathcal{G}_t$: By introducing another auxiliary variable $\mathcal{Y} = [\![\mathcal{G}_s; \mathcal{U}]\!] \in \mathbb{R}^{n_1 \times \cdots \times n_K \times N_s}$, we arrive at the subproblem over $\mathcal{M}$ as

$$\begin{aligned} &\min_{\mathcal{M}} \|[\![\mathcal{X}_s; \mathcal{M}]\!] - \mathcal{Y}\|_F^2 + \lambda \|[\![[\![\mathcal{X}_s; \mathcal{M}]\!]; \mathcal{M}^T]\!] - \mathcal{X}_s\|_F^2 \\ &\text{s.t. } \forall k,\ \boldsymbol{M}^{(k)}\boldsymbol{M}^{(k)T} = \boldsymbol{I} \end{aligned}. \tag{9}$$

Directly solving $\mathcal{M}$ is intractable, but we can optimize each $\boldsymbol{M}^{(k)}$ individually. To this end, Eq. (9) needs to be reformulated further. Let $\boldsymbol{Y}_{(k)}$ be the k-mode unfolding matrix of tensor $\mathcal{Y}$, and $\boldsymbol{M}_{\backslash k} = \boldsymbol{I} \otimes \boldsymbol{M}^{(K)} \otimes \cdots \otimes \boldsymbol{M}^{(k+1)} \otimes \boldsymbol{M}^{(k-1)} \otimes \cdots \otimes \boldsymbol{M}^{(1)}$. Unfolding the k-th mode of the first term in Eq. (9) can be given as

$$\| [\![\mathcal{X}_s; \mathcal{M}]\!] - \mathcal{Y}]_{(k)} \|_F^2 = \|\boldsymbol{M}^{(k)}\boldsymbol{X}_{s(k)}\boldsymbol{M}_{\backslash k}^T - \boldsymbol{Y}_{(k)}\|_F^2. \tag{10}$$

For the regularizer, since $\mathcal{M}$ cannot be directly decomposed into individual $\boldsymbol{M}^{(k)}$, we raise an assumption here to make the optimization tractable in practice. Considering that

$$\begin{aligned} [\![[\![\mathcal{X}_s; \mathcal{M}]\!]; \mathcal{M}^T]\!] = \mathcal{X}_s \times_1 (\boldsymbol{M}^{(1)T}\boldsymbol{M}^{(1)}) \times_2 \ldots \\ \times_K (\boldsymbol{M}^{(K)T}\boldsymbol{M}^{(K)}) \end{aligned}, \tag{11}$$

for the k-th mode, we have

$$\left[\mathcal{X}_s \times_k (\boldsymbol{M}^{(k)T}\boldsymbol{M}^{(k)})\right]_{(k)} = \boldsymbol{M}^{(k)T}\boldsymbol{M}^{(k)}\boldsymbol{X}_{s(k)}\ . \tag{12}$$

Provided that $\boldsymbol{M}_{\backslash k}^T$ is given and all $\boldsymbol{M}^{(i)}$s for $i \neq k$ well preserve the energy of $\mathcal{X}_s$, i.e., we assume $\boldsymbol{M}^{(i)T}\boldsymbol{M}^{(i)} \approx \boldsymbol{I}$, $i \neq k$. Though this assumption seems somewhat heuristic, we show later in experiments the loss decreases normally, which suggests it is at least a good approximation. Hence, optimizing Eq. (9) over $\mathcal{M}$ can be decomposed to K subproblems. The k-th subproblem over $\boldsymbol{M}^{(k)}$ gives

$$\begin{aligned} &\min_{\boldsymbol{M}^{(k)}} \|\boldsymbol{M}^{(k)}\boldsymbol{Q}_{(k)} - \boldsymbol{Y}_{(k)}\|_F^2 + \lambda\|\boldsymbol{M}^{(k)T}\boldsymbol{M}^{(k)}\boldsymbol{X}_{s(k)} - \boldsymbol{X}_{s(k)}\|_F^2 \\ &\text{s.t. } \forall k,\ \boldsymbol{M}^{(k)}\boldsymbol{M}^{(k)T} = \boldsymbol{I} \end{aligned}, \tag{13}$$

where $\boldsymbol{Q}_{(k)} = \boldsymbol{X}_{s(k)}\boldsymbol{M}_{\backslash k}^T$. Notice that

$$\|\boldsymbol{M}^{(k)T}\boldsymbol{M}^{(k)}\boldsymbol{X}_{s(k)} - \boldsymbol{X}_{s(k)}\|_F^2 = \|\boldsymbol{X}_{s(k)}\|_F^2 - \|\boldsymbol{M}^{(k)}\boldsymbol{X}_{s(k)}\|_F^2. \tag{14}$$

Since $\|\boldsymbol{X}_{s(k)}\|_F^2$ remains unchanged during the optimization of $\boldsymbol{M}^{(k)}$, this term can be ignored. Therefore, Eq. (13) can be further simplified as

$$\begin{aligned} &\min_{\boldsymbol{M}^{(k)}} \|\boldsymbol{M}^{(k)}\boldsymbol{Q}_{(k)} - \boldsymbol{Y}_{(k)}\|_F^2 - \lambda\|\boldsymbol{M}^{(k)}\boldsymbol{X}_{s(k)}\|_F^2 \\ &\text{s.t. } \forall k,\ \boldsymbol{M}^{(k)}\boldsymbol{M}^{(k)T} = \boldsymbol{I} \end{aligned}. \tag{15}$$

Finally, by replacing $\boldsymbol{P} = \boldsymbol{M}^{(k)T}$, we can transform Eq. (15) into a standard orthogonality constraint based optimization problem as

$$\begin{aligned} &\min_{\boldsymbol{P}} \|\boldsymbol{Q}_{(k)}^T\boldsymbol{P} - \boldsymbol{Y}_{(k)}^T\|_F^2 - \lambda\|\boldsymbol{X}_{s(k)}^T\boldsymbol{P}\|_F^2 \\ &\text{s.t. } \forall k,\ \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I} \end{aligned}, \tag{16}$$

which can be effectively solved by a standard solver, like the solver presented in [36]. This alternating minimization approach is summarized in Algorithm 1. We observe that the optimization converges only after several iterations.

5. Results and discussion

In this section, we first illustrate the merits of our approach on a standard DA dataset, and then focus on comparisons with related and state-of-the-art methods.

5.1. Datasets, protocol, and baselines

Office–Caltech10 (OC10) dataset. OC10 dataset [12] is the extension of Office [25] dataset by adding another *Caltech* domain, resulting in four domains of *Amazon* (A), *DSLR* (D), *web-cam* (W), and *Caltech* (C). 10 common categories are chosen, leading to around 2500 images and 12 DA problem settings. This dataset reflects the domain shift caused by appearance, viewpoint, background and image resolution. For short, a DA task is denoted by S→T.

Algorithm 1: Alternating minimization for TAISL

Input: Source data: $\mathcal{X}_s$; Target data: $\mathcal{X}_t$
Output: Tensor subspace: $\mathcal{U}$; Alignment matrices: $\mathcal{M}$
Initialize: $\boldsymbol{M}^{(k)} = I, k = 1, ..., K$;
Tensor subspace dimensionality: $d_k, k = 1, ..., K$;
Weight coefficient: λ;
Maximum iteration: T;
for $t \leftarrow 1$ ***to*** T **do**
  Subspace learning over $\{\mathcal{U}, \mathcal{G}_s, \mathcal{G}_t\}$ as per Eq. (8);
  for $k \leftarrow 1$ ***to*** K **do**
    Optimization over $\boldsymbol{M}^{(k)}$ as per Eq. (16);
  Check for convergence;

ImageNet–VOC2007 (IV) dataset. We also evaluate our method on the widely-used ImageNet [7] and PASCAL VOC2007 [8] datasets. The same 20 categories of the VOC2007 are chosen from the ImageNet 2012 dataset to form the source domain, and the `test` set of VOC2007 is used as the target domain. Notice that VOC 2007 is a multi-label dataset. IV dataset reflects the shift when transferring from salient objects to objects in complex scenes. We use this to verify the effectiveness of DA approaches when multiple labels occur.

Experimental protocol. In this paper, we focus on the small-sample-size adaptation, because if enough source and target data are made available, we have better choices with deep adaptation techniques [11, 26] to co-adapt the feature representation, domain distributions and the classifier. In particular, the sampling protocol in [12] is used. Concretely, for both datasets, 20 images are randomly sampled from each category of the source domain (8 images if the domain is *web-cam* or *DSLR*) in each trials. The mean and standard deviation of average multi-class accuracy over 20 trials are reported on OC10 dataset. For the IV dataset, we follow the standard evaluation criterion [8] to use the average precision (AP) as the measure. Similarly, the mean and standard deviation of AP over 10 trials are reported for each category.

Baseline approaches. Several approaches are employed for comparisons:

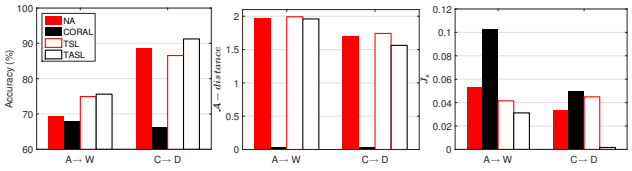
- *No Adaptation (NA)*: NA indicates to train a classifier directly using the labeled source data and applies to the target domain. This is a basic baseline.
- *Principal Component Analysis (PCA)*: PCA is a direct baseline compared to our NTSL approach. It assumes an invariant vector subspace between domains.
- *Daumé III* [6]: Daumé III is a classical DA approach through augmenting the feature representations. Each source data point $\mathbf{x_s}$ is augmented to $\mathbf{x_s}' = (\mathbf{x_s}, \mathbf{x_s}, \mathbf{0})$, and each target data point $\mathbf{x_t}$ to $\mathbf{x_t}' = (\mathbf{x_t}, \mathbf{0}, \mathbf{x_t})$.
- *Transfer Component Analysis (TCA)* [22]: TCA formulates DA in a reproducing kernel Hilbert space by minimizing the maximum mean discrepancy measure.
- *Geodesic Flow Kernel (GFK)* [12]: GFK proposes a closed-form solution to bridge the subspaces of two domains using a geodesic flow in a Grassmann manifold.
- *Domain Invariant Projection (DIP)* [2]: DIP seeks domain-invariant representations by matching the source and target distributions in a low-dimensional reproducing kernel Hilbert space.
- *Subspace Alignment (SA)* [10]: SA directly adopts a linear projection to match the differences between the source and target subspaces. Our approach is closely related to this method.
- *Low-rank Transfer Subspace Learning (LTSL)* [28]: LTSL imposes a low-rank constraint during the subspace learning to enforce only relevant source data are used to reconstruct the target data.
- *Landmarks Selection Subspace Alignment (LSSA)* [1]: LSSA extends SA via selecting landmarks and using further nonlinearity with Gaussian kernel.
- *Correlation Alignment (CORAL)* [30]: CORAL characterizes domains using their covariance matrices. DA is performed through simple whitening and recoloring.

Notice that, for a fair comparison, some methods, e.g., STM [5], that take source labels into account during the optimization are not chosen for comparison, because TAISL does not utilize the information of source labels during DA.

Parameters setting. We extract the convolutional activations from the CONV5_3 layer of VGG–16 model [29] as the tensor representation. We allow the input image to be of arbitrary size, so a simple spatial pooling [14] procedure is applied as the normalization. Specifically, each image will be mapped into a $6 \times 6 \times 512$ third-order tensor. For those conventional approaches, convolutional activations are vectorized into a long vector as the representation. For NTSL and TAISL, we empirically set the tensor subspace dimensionality as $d_1 = d_2 = 6$, and $d_3 = 128$. The first and second modes refer to the spatial location, and the third mode corresponds to the feature. We set such parameters with a motivation to preserve the spatial information and to seek the underlying commonness in the low-dimensional subspace. The weight parameter is set to $\lambda = 1e^{-5}$, and the maximum iteration $T = 10$. Note that we adopt these hyper parameters for all DA tasks when reporting the results. For the comparator approaches, parameters are set according to the suggestions of corresponding papers. One-vs-rest linear SVMs are used as the classifiers, and the penalty parameter C_{svm} is fixed to 1. Please refer to the Supplementary Materials for further details and results.

5.2. Evaluation on the Office–Caltech10 dataset

Before we present the full DA results, we first highlight the merits of tensor subspaces for DA from three aspects: 1) quantifying the domain discrepancy to show how well



(a) (b) (c)

Figure 2: Classification accuracy (a) and domain discrepancy measured by domain-level $\mathcal{A}$-distance (b) and class-level J_s divergence (c) over two DA tasks.

TAISL preserves the discriminative power of the source domain, 2) evaluating the classification performance with a limited number of source/target data to see what scenarios TAISL could be applied in, and 3) replacing source data with noisy labels to verify whether TAISL can resist noise interference.

Quantifying the class-level domain discrepancy. $\mathcal{A}$-distance has been introduced in [3] as a popular measure of domain discrepancy over two distributions. Estimating this distance involves pseudo-labeling the source domain $\mathcal{P}_s$ and target domain $\mathcal{P}_t$ as a binary classification problem. By learning a linear classifier, $\mathcal{A}$-distance can be estimated as $d_{\mathcal{A}}(\mathcal{P}_s, \mathcal{P}_t) = 2(1 - 2\epsilon)$, where ϵ is the generalization error of the linear classifier. The lower $\mathcal{A}$-distance is, the better two distributions align, and vice versa. Given this measure, we empirically examine the correlation between the classification accuracy and $\mathcal{A}$-distance. Fig. 2(a) and Fig. 2(b) illustrate these two measures of several approaches on two DA tasks. Surprisingly, two measures exhibit a totally adverse tendency. The lowest classification accuracy conversely corresponds to the lowest $\mathcal{A}$-distance. As a consequence, at least for convolutional activations, we consider that *the classification accuracy has low correlations with the domain-level discrepancy*. In an effort to explain such a phenomenon, we consider comparing the class-level domain discrepancy taking source labels into account. Two local versions of $\mathcal{A}$-distance are consequently introduced as

$$\begin{aligned} d^w_{\mathcal{A}} &= \frac{1}{C}\sum_{i}^{C} d_{\mathcal{A}}(\mathcal{P}^i_s, \hat{\mathcal{P}}^i_s) \\ d^b_{\mathcal{A}} &= \frac{1}{C(C-1)}\sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} d_{\mathcal{A}}(\mathcal{P}^i_s, \mathcal{P}^j_s) \end{aligned}, \tag{17}$$

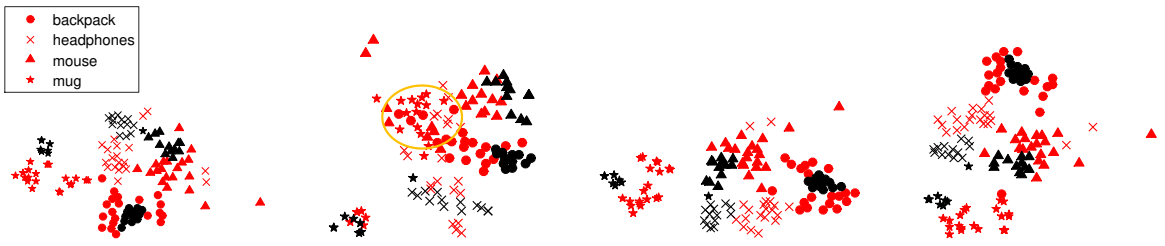
where $d^w_{\mathcal{A}}$ and $d^b_{\mathcal{A}}$ quantifies the within- and between-class divergence, respectively. The superscript in $\mathcal{P}_s$ denotes a specific class in C classes. In particular, considering the fact that, if data can be classified reasonably, it should have small within-class divergence and large between-class divergence. Therefore, $J_s = d^w_{\mathcal{A}}/d^b_{\mathcal{A}}$ is used to score the overall class-level domain discrepancy. Fig. 2(c) shows the value of J_s over the same DA tasks. At this time, the classification accuracy shows a similar trend with the J_s measure. Our analysis justifies the tensor subspace well preserves the discriminative power of source domain.

To give a more intuitive illustration, the data distributions are visualized in Fig. 3. Indeed, the problem occurs during the transfer of source domain. As per the yellow circle in Fig. 3(b), different classes of the source data are overlapped after the adaptation. We call this phenomenon *over-adaptation*. According to a recent study [35], there is a plausible explanation. [35] shows that the feature distributions learned by CNNs are relatively "fat"—the within-class variance is large, while the between-class margin is small. Hence, a slight disturbance would cause the overlaps among different classes. In CORAL, the disturbance perhaps boils down to the inexact estimation of covariance matrices caused by high feature dimensionality and limited source data. In contrast, as shown in Fig. 3(c)-(d), our approach naturally passes the discriminative power of source domain. Notice that, though the adaptation seems not perfect as target data are only aligned close to the source, the margins of different classes are clear so that there still has a high probability for target data to be classified correctly.

Adaptation with limited source/target data. One of the important features of TAISL in practice is the small amount of training data required. In other words, one can characterize a domain, and thus adapt a pre-trained classifier, with very limited data. To demostrate this point, we evaluate the classification accuracy while varying the number of source/target data used for adaptation. The DA task of $D \rightarrow C$ is used. Concretely, we first fix the number of target data and, respectively, randomly choose from 1 to 8 source samples per category. In turn, we fix the number of source data to 8 per category and set the target samples per category to $2^k, k = 0, 1, 2, ..., 7$. Fig. 4(a)-(b) illustrate the results of different approaches. It can be observed that, our approach demonstrates very stable classification performance, while other comparing methods is sensitive to the number of source samples used. Meanwhile, the number of target data seems not to have much impact on the classification accuracy, because in general one prefers to transfer the source domain so that the target domain does not change notably. It is worth noting that TAISL works even with only one source sample per category, which suggests that it can be applied for effective small-sample-size adaptation.

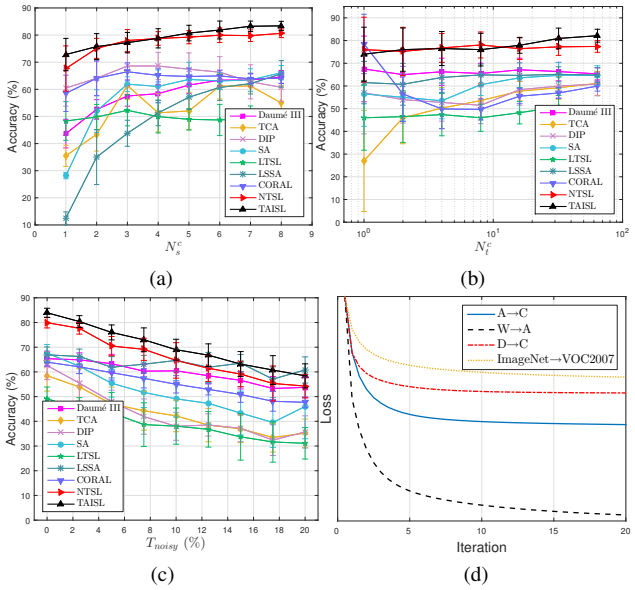
Adaptation with noisy labels. Recent studies [40] demonstrate that tensor representations are inherently robust to noise. To further justify this in the context of DA, we randomly replace the source data with samples that have different labels. We gradually increase the percentage of noisy data T_{noisy} from 0% to 20% and monitor the degradation of classification accuracy. As shown in Fig. 4(c), TAISL consistently demonstrates superior classification performance over other approaches.

Convergence analysis and efficiency comparison. In this part, we empirically analyze the convergence behavior of TAISL. Fig. 4(d) shows the change of loss function as



(a) NA (b) CORAL (c) NTSL (d) TAISL

Figure 3: Class-level data visualization using t-SNE [31] of different methods on the DA task of C (red) → D (black). 4 classes are chosen for better visualization. For CORAL, the data coming from the source domain tend to overlap with each other after the adaptation, a phenomenon we call *over-adaptation*. (Best viewed in color.)



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

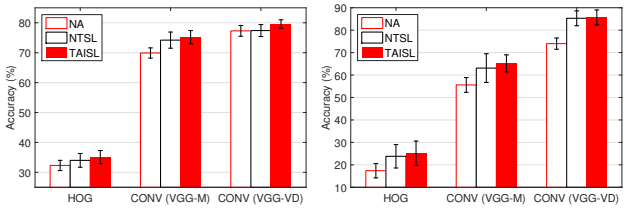
Figure 4: Adaptation on D→C with (a) varying number of source data per class N_s^c, (b) a varying number of target data per class N_t^c, and (c) noisy source labels. (d) Empirical convergence analysis of TAISL over several DA tasks.

the iteration increases. It can be observed that the optimization generally converges in about 10 iterations. In addition, we also compare the efficiency of different approaches. The average evaluation time of each trial is reported. According to Table 1, the efficiency of TAISL is competitive too. TCA and LSSA are fast, because these two methods adopt kernel tricks to avoid high-dimensional computation implicitly. In general, learning a tensor subspace is faster than a vector subspace in the high-dimensional case.

Recognition results. Quantitative results are listed in Table 2. It shows that our approach is on par with or outperforms other related and state-of-the-art methods in terms of both average accuracy and standard deviations. Note that conventional methods that directly adapt vector-form convolutional activations sometimes have a negative effect on the classification, even falling behind the baseline NA. The main reason perhaps is the inexact estimation of a large amount of parameters. For instance, in many subspace-based approaches, one needs to estimate a flattened subspace from the covariance matrix. Given a data matrix $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ with dimension d and n samples, its covariance matrix is estimated as AA^T. Notice that $rank(A) = rank(AA^T) = rank(A^TA) \leq \min(d, n-1)$. If $d \gg n$, the vector subspace will only be spanned by less than n eigenvectors. In addition, one also suffers from the problem of biased estimation [34] (large eigenvalues turn larger, small ones turn smaller) when $d \gg n$. Hence, such vector subspaces are unreliable. In contrast, our approach avoids this problem due to the mode-wise parameters estimation.

Method	Daumé III	TCA	GFK	DIP	SA
Time	0.06	0.05	3.94	9.09	3.40
Method	LTSL	LSSA	CORAL	NTSL	TAISL
Time	12.34	0.59	14.81	0.16	0.92

Table 1: Average evaluation time (min) of each trial of different methods on A→C. (Matlab 2016a, OS: OS X 64-bit, CPU: Intel i5 2.9GHz, RAM: 8 GB)



(a) A→C (b) W→A

Figure 5: Adaptation accuracy of three types of tensor representations on two DA tasks.

5.3. Evaluation on the ImageNet–VOC2007 dataset

Here we evaluate our approach under a more challenging dataset than OC10. As aforementioned, VOC2007 is a multi-label dataset, so many images contain multiple labels. Results are listed in Table 3. Due to the space limitation, we show only results of 10 categories (additional results are attached in the Supplementary). We observe that TAISL still demonstrates the best overall classification performance among comparing approaches. We also notice that NTSL and TAISL show comparable results. We conjecture that, since the target domain contains too many

Method	A→C	C→A	A→D	D→A	A→W	W→A	C→D	D→C	C→W	W→C	D→W	W→D	MEAN
NA	77.3(1.8)	89.0(2.0)	82.8(2.2)	81.1(1.9)	74.6(3.1)	74.0(2.5)	86.2(4.0)	70.5(1.9)	79.4(2.7)	63.7(2.1)	91.1(1.7)	94.9(2.4)	80.4
PCA	36.7(3.0)	57.7(5.2)	23.5(8.1)	50.2(4.8)	18.6(6.2)	51.2(6.0)	29.9(7.8)	51.0(3.3)	26.4(6.7)	51.6(3.6)	49.5(4.2)	50.8(7.0)	41.4
Daumé III	73.1(1.5)	85.9(2.5)	70.9(3.7)	59.9(7.1)	70.6(3.5)	68.7(4.4)	81.4(4.1)	56.2(6.4)	75.4(4.0)	59.6(2.6)	81.5(2.7)	86.5(4.9)	72.5
TCA	56.7(4.5)	78.1(6.1)	59.9(6.7)	61.2(4.2)	55.5(6.4)	68.3(4.1)	74.3(5.2)	51.9(2.2)	69.0(6.6)	54.7(3.8)	89.8(2.2)	90.6(3.2)	67.5
GFK	75.1(3.9)	87.6(2.3)	81.4(4.3)	90.4(1.4)	74.3(5.2)	84.0(4.4)	84.8(4.5)	82.2(2.4)	81.9(4.9)	79.1(2.7)	92.8(2.2)	95.2(2.2)	84.1
DIP	59.8(5.7)	84.8(4.3)	52.2(8.1)	76.4(3.7)	45.5(9.1)	69.3(6.9)	82.8(7.7)	61.9(6.3)	73.5(4.9)	65.2(4.5)	90.9(2.3)	94.1(3.1)	71.4
SA	67.7(4.2)	82.0(2.6)	67.8(4.8)	77.4(6.0)	61.1(5.1)	80.1(4.3)	73.7(4.3)	66.9(3.3)	65.9(4.0)	70.4(4.1)	87.3(3.1)	91.1(3.3)	74.3
LTSL	70.2(2.4)	87.5(2.8)	77.7(4.6)	69.2(4.5)	66.7(4.6)	66.6(5.7)	82.3(4.1)	60.8(3.1)	75.3(4.2)	59.1(4.4)	86.0(2.9)	90.0(3.8)	74.3
LSSA	**80.3(2.3)**	86.4(1.7)	**90.9(1.7)**	**92.3(0.6)**	**84.0(1.7)**	**86.6(4.5)**	73.5(2.3)	65.9(6.5)	45.4(6.6)	29.5(7.0)	93.4(2.2)	85.8(4.7)	76.2
CORAL	77.6(1.2)	80.3(1.9)	64.3(2.9)	74.2(2.2)	61.2(2.4)	69.1(2.6)	62.1(3.0)	72.0(1.7)	63.8(3.1)	66.6(2.2)	89.6(1.6)	82.8(2.8)	72.0
NTSL	78.5(2.3)	89.6(2.2)	83.1(3.3)	87.8(1.4)	77.3(3.1)	85.8(2.8)	87.7(2.9)	79.8(1.5)	80.4(3.8)	80.0(2.0)	95.4(1.4)	**97.8(1.7)**	85.3
TAISL	80.1(1.4)	**90.0(1.9)**	85.1(2.2)	87.6(2.1)	77.9(2.6)	85.6(3.5)	**90.6(1.9)**	**84.0(1.0)**	**85.3(3.1)**	**82.6(2.2)**	**95.9(1.0)**	97.7(1.5)	**86.9**

Table 2: Average multi-class recognition accuracy (%) on Office–Caltech10 dataset over 20 trials. The highest accuracy in each column is boldfaced, the second best is marked in red, and standard deviations are shown in parentheses.

Method	aero	bird	bottle	cat	cow	table	mbike	person	sheep	tv	mAP
NA	66.4(2.1)	65.6(4.0)	29.5(2.1)	70.6(3.4)	30.3(8.0)	35.7(5.5)	47.0(8.0)	69.3(2.9)	44.9(6.9)	56.4(3.3)	51.6
PCA	28.9(5.8)	30.2(3.9)	23.3(5.2)	44.9(4.5)	6.0(1.8)	29.0(6.9)	25.0(5.0)	70.2(1.9)	11.7(3.5)	29.0(6.6)	29.8
Daumé III	64.1(3.7)	59.7(7.4)	26.6(3.3)	65.7(5.3)	26.9(8.5)	30.0(5.4)	40.5(6.6)	68.5(2.5)	37.7(7.4)	51.9(4.4)	47.2
TCA	43.2(9.8)	44.4(10.5)	20.7(1.7)	56.7(8.2)	16.9(6.3)	27.6(8.7)	31.8(10.2)	58.1(5.7)	22.7(8.0)	33.6(10.2)	35.6
GFK	70.0(6.9)	74.6(3.8)	32.5(4.4)	73.1(6.5)	28.9(5.3)	48.3(10.4)	58.3(4.8)	**75.8(3.6)**	52.5(4.8)	57.1(4.5)	57.1
DIP	69.8(5.5)	78.4(4.6)	29.1(5.0)	75.9(3.7)	25.5(5.0)	42.2(8.1)	56.3(5.7)	73.5(3.1)	48.9(4.3)	59.4(5.2)	55.9
SA	64.4(10.1)	69.3(5.4)	34.4(4.6)	67.4(4.9)	18.4(6.6)	36.9(12.8)	53.7(10.9)	68.9(2.4)	31.4(10.2)	55.2(5.7)	50.0
LTSL	56.9(10.4)	61.0(7.7)	34.9(6.2)	70.8(8.8)	21.9(6.3)	43.7(12.4)	52.5(10.7)	69.9(4.3)	38.2(9.5)	54.0(7.5)	50.4
LSSA	**78.7(2.0)**	**79.7(1.2)**	**38.4(4.6)**	**81.7(0.5)**	29.5(1.9)	33.7(3.4)	56.3(9.3)	51.2(2.0)	32.5(10.6)	51.6(4.6)	53.3
CORAL	71.4(3.3)	71.7(3.6)	35.2(2.4)	72.0(4.3)	**36.0(5.7)**	40.6(6.7)	57.3(5.6)	67.6(2.0)	**54.8(2.9)**	56.9(3.6)	56.6
NTSL	76.3(4.3)	71.0(3.9)	35.7(3.7)	71.3(3.2)	34.7(9.8)	49.8(10.4)	59.7(10.2)	72.0(4.6)	53.4(6.0)	60.2(3.5)	58.4
TAISL	76.4(5.1)	71.6(3.1)	36.7(3.5)	72.0(2.1)	33.3(6.6)	**50.7(10.0)**	**60.3(8.7)**	72.2(3.8)	53.6(5.6)	**60.4(3.5)**	**58.7**

Table 3: Average precision (%) on ImageNet–VOC2007 dataset over 10 trials. The highest performance in each column is boldfaced, the second best is marked in red, and standard deviations are shown in parentheses.

noisy labels, it will be hard to determine a global alignment that just matches class-level differences. As a result, the alignment may not work the way it should. In addition, according to Tables 2 and 3, LSSA shows superior accuracy than ours over several DA tasks/categories. It makes sense because LSSA works at different levels with further non-linearity and samples reweighting. However, non-linearity is a double-edged sword. It can improve the accuracy in some situations, while sometimes it may not. For instance, the accuracy of LSSA drops significantly on the W→C task.

5.4. Evaluation with other tensor representations

Finally, we evaluate other types of tensor representations to validate the generality of our approach. We do not limit the representation from deep learning features. Other shallow tensor features also can be adapted by our approach. Specifically, the improved HOG feature [9] and convolutional activations extracted from the CONV5 layer of VGG–M [4] model are further utilized and evaluated on two DA tasks from the OC10 dataset. Results are shown in Fig. 5. We notice that TAISL consistently improves the recognition accuracy with various tensor representations. In addition, a tendency shows that, the better feature representations are, the higher the baseline achieves, which implies a fundamental rule of domain-invariant feature representations for DA.

6. Conclusion

Practical application of machine learning techniques often gives rise to situations where domain adaptation is required, either because acquiring the perfect training data is difficult, the domain shift is unpredictable, or simply because it is easier to re-use an existing model than to train a new one. This is particularly true for CNNs as the training time and data requirements are significant.

The DA method proposed in this work is applicable in the case where a tensor representation naturally captures information that would be difficult to represent using vector arithmetic, but also benefits from the fact that it uses a lower-dimensional representation to achieve DA, and thus is less susceptible to noise. We have shown experimentally that it outperforms the state of the art, most interestingly for CNN DA, but is also much more efficient.

In the future work, discriminative information from source data may be employed for learning a more powerful invariant tensor subspace.

Acknowledgments. This work was supported in part by the National High-tech R&D Program of China (863 Program) under Grant 2015AA015904 and in part by the National Natural Science Foundation of China under Grant 61502187 and Grant 61671385.

References

[1] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 56–63, 2015.

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

[3] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, et al. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 19, page 137, 2007.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. British Machine Vision Conference (BMVC)*, 2014.

[5] W.-S. Chu, F. De La Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[6] H. Daumé III. Frustratingly easy domain adaptation. In *Proc. Association for Computational Linguistics (ACL)*, 2007.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

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

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

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

[13] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006, 2011.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] T. K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, Aug 2009.

[17] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

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

[19] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Robust visual tracking based on incremental tensor subspace learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[21] H. Lu, Z. Cao, Y. Xiao, and Y. Zhu. Two-dimensional subspace alignment for convolutional activations adaptation. *Pattern Recognition*, 71:320–336, 2017.

[22] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, Feb 2011.

[23] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, 2015.

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

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

[26] O. Sener, H. O. Song, A. Saxena, and S. Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2110–2118, 2016.

[27] G. Shakhnarovich and B. Moghaddam. Face recognition in subspaces. In *Handbook of Face Recognition*, pages 19–49. Springer, 2011.

[28] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2):74–93, 2014.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proc. AAAI Conference on Artificial Intelligence*, 2016.

[31] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579–2605):85, 2008.

[32] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. European Conference on Computer Vision (ECCV)*, pages 447–460. Springer, 2002.

[33] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–93. IEEE, 2003.

[34] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li. Beyond covariance: Feature representation with nonlinear kernel matrices. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 4570–4578, 2015.

[35] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 499–515. Springer, 2016.

[36] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NIPS)*, pages 3320–3328, 2014.

[38] L. Zhang, W. Wei, C. Tian, F. Li, and Y. Zhang. Exploring structured sparsity by a reweighted laplace prior for hyperspectral compressive sensing. *IEEE Transactions on Image Processing*, 25(10):4974–4988, 2016.

[39] L. Zhang, W. Wei, Y. Zhang, C. Shen, A. van den Hengel, and Q. Shi. Dictionary learning for promoting structured sparsity in hyperspectral compressive sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7223–7235, 2016.

[40] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, Sept 2015.