Adversarial learning has been successfully embedded into deep networks to learn transferable features for domain adaptation, which reduce distribution discrepancy between the source and target domains and improve generalization performance. Prior domain adversarial adaptation methods could not align complex multimode distributions since the discriminative structures and inter-layer interactions across multiple domain-specific layers have not been exploited for distribution alignment. In this paper, we present randomized multilinear adversarial networks (RMAN), which exploit multiple feature layers and the classifier layer based on a randomized multilinear adversary to enable both deep and discriminative adversarial adaptation. The learning can be performed by stochastic gradient descent with the gradients computed by back-propagation in linear-time. Experiments demonstrate that our models exceed the state-of-the-art results on standard domain adaptation datasets.

1 Introduction

Deep networks have significantly improved the state of the arts for diverse machine learning problems and applications. Unfortunately, the impressive performance gains come only when massive amounts of labeled data are available for supervised learning. Since manual labeling of sufficient training data for diverse application domains on-the-fly is often prohibitive, for a target task short of labeled data, there is strong motivation to build effective learners that can leverage rich labeled data in a different source domain. However, this learning paradigm suffers from the shift in data distributions across different domains, which poses a major obstacle in adapting learning models for a target task [30, 29].

Learning a discriminative model that reduces the dataset shift between training and test distributions is known as transfer learning or domain adaptation [29]. Previous shallow transfer learning methods either bridge the source and target domains by learning invariant feature representations or estimating instance importance without using target labels [18, 28, 11]. Recent deep transfer learning methods leverage deep networks to learn more transferable representations by embedding domain adaptation in the pipeline of deep learning, which can simultaneously disentangle the explanatory factors of variations behind data and match the marginal data distributions across domains [38, 22, 9, 39, 23].

Adversarial adaptation methods [9, 39] are the top-performing architectures for domain adaptation. These methods work similarly as generative adversarial networks [13]: a domain discriminator is learned by minimizing the classification error of distinguishing the source from the target, while a deep network learns transferable representations which are indistinguishable by the domain discriminator. However, existing methods may be confined by two key bottlenecks. First, when data distributions embody complex multimode structures, adversarial adaptation may fail to capture such rich structures to ensure fine-grained alignment of distributions, known as the mode collapse difficulty in generative
adversarial networks [13, 2]. Second, the dataset shifts may linger in multiple domain-specific higher layers [40], and adversarial adaptation of a particular layer is not sufficient to close the domain shifts.

This paper presents Randomized Multilinear Adversarial Networks (RMAN), which largely extends the ability of deep adversarial adaptation [9] to align the joint distributions of multiple domain-specific layers across domains for unsupervised domain adaptation. A key improvement is the exploitation of discriminative structures revealed by the classifier layer to the deep adversarial adaptation modules, which can effectively circumvent the difficulty of matching multi-mode distributions across domains. RMAN admits a simple transfer pipeline, which processes the source and target data by convolutional neural networks (CNN) and aligns the joint distributions of multilayer activations based on a new randomized multilinear adversary. The overall system can be solved efficiently via back-propagation. Experiments show that our models exceed state of the art results on two domain adaptation datasets.

2 Related Work

Transfer learning [29] generalizes a learner across different domains of different data distributions [37, 28, 8, 11, 41], which is widely applied in computer vision [33, 14, 12, 17] and natural language processing [6, 10]. Deep networks learn abstract representations that disentangle the explanatory factors of variations behind data [4] and extract transferable factors underlying different populations [10, 27], which can only reduce, but not remove, the cross-domain discrepancy [40]. Recent work on deep domain adaptation embeds domain-adaptation modules into deep network to boost transfer performance [38, 9, 39, 22, 23]. These methods mainly reduce the data shifts in marginal distributions.

Adversarial learning has been explored for generative modeling. Generative Adversarial Networks (GAN) [13] constitute two networks in a two-player game: a generator that captures data distribution and a discriminator that distinguishes between generated samples and ground truth data. The networks are trained jointly in a mini-max fashion such that the generator is learned to fool the discriminator. Recently, several difficulties of GANs have been addressed, e.g. ease training [2, 1], avoid mode collapse [26, 5, 25]. These technologies have not been leveraged to improve adversarial adaptation.

3 Randomized Multilinear Adversarial Networks

In unsupervised domain adaptation, we are given a source domain \( D_s = \{ (x^s_i, y^s_i) \} \) of \( n_s \) labeled examples and a target domain \( D_t = \{ x^t_i \} \) of \( n_t \) unlabeled examples. The source domain and target domain are sampled from joint distributions \( P(X^s, Y^s) \) and \( Q(X^t, Y^t) \) respectively, and \( P \neq Q \). The goal of this paper is to design a deep network \( y = F(x) \) which formally reduces the shifts in the joint distributions across domains, such that the target risk expectation \( \mathbb{E}_Q(F(x) \neq y) \) can be minimized by jointly minimizing the source risk and the domain discrepancy via adversarial learning.

Deep networks [4] can learn more transferable representations than traditional hand-crafted features [27, 40]. The favorable transferability of deep features leads to several state of the art deep transfer learning methods [9, 39, 22, 23]. This paper also tackles unsupervised domain adaptation by learning transferable features using deep networks. We extend deep convolutional networks (CNNs), e.g. AlexNet [21] and ResNet [16], to novel randomized multilinear adversarial networks (RMANs) as shown in Figure 1. The empirical error of CNN classifier \( F(x) \) on source domain labeled data \( D_s \) is

\[
\min_F \frac{1}{n_s} \sum_{i=1}^{n_s} J(F(x^s_i), y^s_i),
\]

where \( J(\cdot, \cdot) \) is the cross-entropy loss function. Based on the quantification study of feature transferability in deep convolutional networks [40], convolutional layers can learn generic features that are transferable across domains [40]. Thus we opt to fine-tune the features of convolutional layers when transferring deep models pre-trained on large-scale ImageNet from source domain to target domain.

However, the literature findings also reveal that the deep features can reduce, but not remove, the cross-domain distribution discrepancy [40, 22, 23]. The deep features in CNNs eventually transition from general to specific along the network, and the transferability of features and classifiers decreases when the cross-domain discrepancy increases [40]. In other words, even feed-forwarding the source and target domain data through the deep network for multilayer feature abstraction, the shifts in the joint distributions \( P(X^s, Y^s) \) and \( Q(X^t, Y^t) \) still linger in the activations \( Z^1, \ldots, Z^{L-1} \) of the higher
network layers $\mathcal{L}$. Taking AlexNet \cite{21} as an example, the activations in the higher fully-connected layers $\mathcal{L} = \{fc6, fc7, fc8\}$ are not safely transferable for domain adaptation \cite{40}. Thus we can use the joint distributions of the activations in layers $\mathcal{L}$, i.e. $P(Z^s_1, \ldots, Z^s_{|\mathcal{L}|})$ and $Q(Z^t_1, \ldots, Z^t_{|\mathcal{L}|})$ to embody the discrepancy in the original joint distributions $P(X^s, Y^s)$ and $Q(X^t, Y^t)$, respectively. We enable unsupervised domain adaptation by matching $P(Z^s_1, \ldots, Z^s_{|\mathcal{L}|})$ and $Q(Z^t_1, \ldots, Z^t_{|\mathcal{L}|})$.

### 3.1 Randomized Multilinear Adversary

Many existing methods address transfer learning by bounding the target error with the source error plus a discrepancy between the marginal distributions $P(X^s)$ and $Q(X^t)$ of the source and target domains \cite{3}. A rich line of work is based on the Maximum Mean Discrepancy (MMD) \cite{15}, a kernel two-sample test statistic, which measures the discrepancy in marginal distributions $P(X^s)$ and $Q(X^t)$ \cite{38, 22, 23}. Recently, motivated by the success of generative adversarial networks \cite{13}, domain adversarial adaptation has been proposed \cite{9, 39} to measure the discrepancy in distributions $P(X^s)$ and $Q(X^t)$ by a domain discriminator. In domain adversarial adaptation, the domain discriminator minimizes the classification error of distinguishing the source from target, in the meantime, the deep network learns transferable representations that are not distinguishable by the domain discriminator. It has been shown by \cite{9} that the error functional of the domain discriminator is well corresponded to the distribution discrepancy used to bound the target risk in the theory of domain adaptation \cite{13, 24}.

To date, it remains unclear how to reduce the discrepancy in the joint distributions $P(Z^s_1, \ldots, Z^s_{|\mathcal{L}|})$ and $Q(Z^t_1, \ldots, Z^t_{|\mathcal{L}|})$, which is crucial to further improve the performance of domain adaptation. Mode collapse \cite{13, 2} has been a well-known difficulty in adversarial learning, which says that when the distribution is multimode, there is the risk that different modes cannot be matched across domains. By matching the joint distributions $P(Z^1_1, \ldots, Z^1_{|\mathcal{L}|})$ and $Q(Z^t_1, \ldots, Z^t_{|\mathcal{L}|})$ instead of the marginal distributions $P(X^s)$ and $Q(X^t)$, the discriminative information conveyed in the classifier layer $Z^{|\mathcal{L}|}$ may reveal the multimode structure \cite{20}, which can further enable multimode adversarial adaptation.

In order to reduce the discrepancy in the joint distributions $P(Z^s_1, \ldots, Z^s_{|\mathcal{L}|})$ and $Q(Z^t_1, \ldots, Z^t_{|\mathcal{L}|})$, a straightforward solution is to concatenate the activations of layers $\mathcal{L}$ in a single vector $[Z^s_1; \ldots; Z^s_{|\mathcal{L}|}]$ upon which the adversarial adaptation can be performed. However, this solution has two limitations: (1) concatenation cannot capture the full interactions between deep features and classifier predictions, which are important for domain adaptation; (2) the activations of feature layers and the classifier layer are of different magnitudes, while concatenation may not be amenable to different magnitudes.

In this paper, we follow \cite{23} and adopt the tensor product between activations of multiple layers $\mathcal{L}$ to perform lossless multilinear fusion, i.e. $T(Z^s) \triangleq \otimes_{\ell \in \mathcal{L}} Z^{s\ell}$ and $T(Z^t) \triangleq \otimes_{\ell \in \mathcal{L}} Z^{t\ell}$. Due to the multi-linearity of tensor product, it is immune to different magnitudes of the activations in different layers. Tensor product of infinite-dimensional nonlinear feature maps has been successfully applied to embed joint/conditional distributions into reproducing kernel Hilbert spaces (RKHSs) \cite{36, 34, 35} due to its great capability of capturing the full interactions between different sets of random variables.

Figure 1: The architecture of Randomized Multilinear Adversarial Network (RMAN) for deep domain adaptation, where $D$ is the domain discriminator, $R^\ell$ is the random map for the $\ell$-layer hidden features $Z^\ell$, and $\odot$ is the Hadamard product. The joint distributions $P(Z^s_1, \ldots, Z^s_{|\mathcal{L}|})$ and $Q(Z^t_1, \ldots, Z^t_{|\mathcal{L}|})$ of network activations in domain-specific layers $\mathcal{L}$ are aligned by multilinear adversarial adaptation.
A substantial disadvantage of the tensor product is the dimension explosion issue. Denote by $d_\ell$ the dimension of vector $Z^\ell$, then the dimension of tensor product $T(Z) \triangleq \otimes_{\ell \in \mathcal{L}} Z^\ell$ will be $\prod_{\ell=1}^{|\mathcal{L}|} d_\ell$, which is too high-dimension to be embedded into deep networks without causing parameter explosion. This paper addresses the dimension explosion of multilinear fusion by randomized methods \cite{31, 20}. Denote by $z'$ the instantiation of $Z^\ell$, using the property of tensor product in Hilbert space we obtain

$$
\langle T(z), T(z') \rangle = \langle \otimes_{\ell=1}^{|\mathcal{L}|} z^\ell, \otimes_{\ell=1}^{|\mathcal{L}|} z'^\ell \rangle = \prod_{\ell=1}^{|\mathcal{L}|} \langle z^\ell, z'^\ell \rangle \approx \langle \phi_L(z), \phi_L(z') \rangle,
$$

where $\langle \cdot, \cdot \rangle$ is the inner-product, and $\phi_L(z)$ is the explicit $d$-dimensional feature map ($d \ll \prod_{\ell=1}^{|\mathcal{L}|} d_\ell$) so that the linear kernel on $T(z)$ is approximated by the linear kernel on $\phi_L(z)$. We define $\phi_L(z)$ as

$$
\phi_L(z) = \frac{1}{\sqrt{d}} \left( \otimes_{\ell}^{|\mathcal{L}|} R^\ell z^\ell \right),
$$

where $\otimes$ is the element-wise product (Hadamard product), $R^\ell$ is a random matrix, and each of its element $R^\ell_{ij}$ follows a symmetric distribution with uni-variance, i.e. $\mathbb{E} [R^\ell_{ij}] = 0, \mathbb{E} [(R^\ell_{ij})^2] = 1$. Applicable distributions include Bernoulli distribution, Gaussian distribution and uniform distribution.

**Theorem 1.** The expectation and variance of randomized feature map $\phi_L(z)$ \cite{8} over $R^\ell_{ij}, \ell = 1, \ldots, |\mathcal{L}|$, satisfy

$$
\mathbb{E} \left[ \langle \phi_L(z), \phi_L(z') \rangle \right] = \prod_{\ell=1}^{\mathcal{L}} \langle z^\ell, z'^\ell \rangle,
$$

$$
\text{var} \left[ \langle \phi_L(z), \phi_L(z') \rangle \right] = \sum_{i=1}^{d} \left( \otimes_{\ell}^{|\mathcal{L}|} \sum_{j=1}^{d_\ell} \left( (z^\ell_{ij})^2 (z'^\ell_{ij})^2 \mathbb{E} \left[ (R^\ell_{ij})^4 \right] + C' \right) \right) + C,
$$

where $C$ and $C'$ are constants that do not depend on random matrices $R^\ell, \ell = 1, \ldots, |\mathcal{L}|$.

**Proof.** Please refer to the supplemental material for the detailed proof. \hfill \square

Following the idea of domain adversarial adaptation \cite{9, 39}, we train a domain discriminator network $D$ to discriminate the joint distributions $P(Z^{s1}, \ldots, Z^{s|\mathcal{L}|})$ and $Q(Z^{t1}, \ldots, Z^{t|\mathcal{L}|})$ in an optimal way, and use the loss of the domain discriminator $D$ as a reasonable estimate to the domain discrepancy. The empirical risk of domain discriminator $D(\phi_L(z))$ on source and target domain data $D_s$ and $D_t$ is

$$
\min_D - \frac{1}{n_s} \sum_{i=1}^{n_s} \log D(\phi_L(z^s_i)) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log (1 - D(\phi_L(z^t_j))),
$$

where $\phi_L(z^s_i) = \frac{1}{\sqrt{d}} \left( \otimes_{\ell}^{|\mathcal{L}|} R^\ell z^{s\ell}_i \right), \phi_L(z^t_j) = \frac{1}{\sqrt{d}} \left( \otimes_{\ell}^{|\mathcal{L}|} R^\ell z^{t\ell}_j \right)$ are the randomized multilinear fusion of source and target activations in multiple domain-specific layers $\mathcal{L}$. We term Equation (6) as randomized multilinear adversary to emphasize its role for adversarial adaptation of joint distributions.

**Remark:** The randomized multilinear adversary \cite{6} differs from previous domain adversarial adaptation methods \cite{9, 39} based on ordinal adversary in that, for the activations $Z^\ell$ in each layer $\ell \in \mathcal{L}$, the randomized multilinear operation puts non-uniform weights reflecting the influence of other variables in other layers $\mathcal{L}\setminus\ell$. This captures the full interactions between different variables in the joint distributions $P(Z^{s1}, \ldots, Z^{s|\mathcal{L}|})$ and $Q(Z^{t1}, \ldots, Z^{t|\mathcal{L}|})$, which is crucial for domain adaptation.

### 3.2 Randomized Multilinear Adversarial Networks

Denote by $\mathcal{L}$ the domain-specific layers in deep networks where activations are not safely transferable. The features in lower layers are safely transferable and will not need distribution adaptation \cite{40}. We reduce the discrepancy in joint distributions $\mathcal{L}$, i.e. $P(Z^{s1}, \ldots, Z^{s|\mathcal{L}|})$ and $Q(Z^{t1}, \ldots, Z^{t|\mathcal{L}|})$. To enable domain adversarial adaptation, we jointly (1) minimize the CNN error \cite{1} with respect to source network $F(x)$, (2) minimize the adversary error \cite{6} with respect to the domain discriminator $D(\phi_L(z))$, and (3) maximize the adversary error \cite{6} with respect to the source network $F(x)$. This
yields the optimization problem for learning Randomized Multilinear Adversarial Networks (RMAN):

\[
\min_F \frac{1}{n_s} \sum_{i=1}^{n_s} J(F(x_s^i), y_s^i) + \lambda \frac{1}{n_s} \sum_{i=1}^{n_s} \log D(\phi_L(z_s^i)) + \frac{\lambda}{n_t} \sum_{j=1}^{n_t} \log (1 - D(\phi_L(z_t^j)))
\]

\[
\min_D \frac{1}{n_s} \sum_{i=1}^{n_s} \log D(\phi_L(z_s^i)) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log (1 - D(\phi_L(z_t^j))),
\]

where \( \lambda > 0 \) is a balance parameter between the CNN loss and the adversary loss. As shown in Figure [1] we set \( \mathcal{L} = \{f_{c6}, f_{c7}, f_{c8}\} \) for the RMAN model based on AlexNet (last three layers) and we set \( \mathcal{L} = \{pool5, f_c\} \) for the RMAN model based on ResNet (last two layers). Empirical results show that adversarial adaptation of these domain-specific layers is sufficient for domain adaptation.

**Remark:** The RMAN models improve existing domain adversarial adaptation methods [9, 39] by jointly adapting multiple layers (in particular, the classifier layer is jointly included for adaptation), which potentially circumvents the mode collapse difficulty during adversarial training by incorporating discriminative information to the adversary. The RMAN models also improve existing deep adaptation methods [22, 23] by randomized multilinear fusion of multiple domain-specific layers, which removes cumbersome layer-wise hyper-parameters in [22] and tackles dimension explosion difficulty in [23].

## 4 Experiments

We evaluate the randomized multilinear adversarial networks with several state-of-the-art transfer learning and deep learning methods. The codes, datasets, and configurations will be available online.

### 4.1 Setup

**Office-31** [33] is a standard benchmark for visual domain adaptation, comprising 4,652 images and 31 categories collected from three distinct domains: Amazon (A), which contains images downloaded from [amazon.com](http://amazon.com), Webcam (W), and DSLR (D), which contain images respectively taken by web camera and digital SLR camera with different environments. We evaluate all methods across three transfer tasks A → W, D → W and W → D, which are widely used by previous deep transfer learning methods [38, 9], and another three transfer tasks A → D, D → A and W → A as used in [22, 39, 23].

**ImageCLEF-DA** is a benchmark dataset for ImageCLEF 2014 domain adaptation challenge, which is organized by selecting the 12 common categories shared by the following three public datasets, each is considered as a domain: Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). The 12 common categories are aeroplane, bike, bird, boat, bottle, bus, car, dog, horse, monitor, motorbike, and people. There are 50 images in each category and 600 images in each domain. We use all domain combinations and build 6 transfer tasks: I → P, P → I, I → C, C → I, C → P, and P → C. Different from the **Office-31** dataset where different domains are of different sizes, the three domains in this dataset are of equal size, which makes it a good complement to the **Office-31** dataset.

We compare the proposed randomized multilinear adversarial network (RMAN) with both shallow and deep transfer learning methods: Transfer Component Analysis (TCA) [28], Geodesic Flow Kernel (GFK) [12], Deep Domain Confusion (DDC) [38], Deep Adaptation Network (DAN) [22], Residual Transfer Network (RTN) [23], and Reverse Gradient (RevGrad) [9]. TCA learns a shared feature space by Kernel PCA with linear-MMD penalty. GFK interpolates across an infinite number of intermediate subspaces to bridge the source and target subspaces. For these shallow transfer methods, we adopt SVM as the base classifier. DDC maximizes domain confusion by adding to deep networks a single adaptation layer that is regularized by linear-kernel MMD. DAN learns transferable features by embedding deep features of multiple domain-specific layers to reproducing kernel Hilbert spaces (RKHSs) and matching different distributions optimally using multi-kernel MMD. RTN jointly learns transferable features and adapts different source and target classifiers via deep residual learning [16]. RevGrad enables domain adversarial learning [13] by adapting a single layer of deep networks, which matches the source and target domains by making them indistinguishable for a domain discriminator.

We follow standard evaluation protocols for unsupervised domain adaptation [22, 9]. For both **Office-31** and **ImageCLEF-DA** datasets, we use all labeled source examples and all unlabeled target examples.

[^http://imageclef.org/2014/adaptation]: http://imageclef.org/2014/adaptation
We compare the average classification accuracy of each method on three random experiments, and report the standard error of the classification accuracies by different experiments of the same transfer task. For all baseline methods, we either follow their original model selection procedures, or conduct transfer cross-validation [42] if their model selection strategies are not specified. We also adopt transfer cross-validation [42] to select parameter $\lambda$ for the RMAN models. Fortunately, our models perform very stably under different parameter values, thus we fix $\lambda = 1$ throughout all experiments. For MMD-based methods (TCA, DDC, DAN, and RTN), we use Gaussian kernel with bandwidth set to the median pairwise squared distances on the training data, i.e. median trick [15, 22]. We examine the influence of deep representations for domain adaptation by exploring AlexNet [21] and ResNet [16] as base architectures for learning deep representations. For shallow methods, we follow DeCAF [7] and use as deep representations the activations of the $fc7$ (AlexNet) and $pool5$ (ResNet) layers. To see the effectiveness of different sampling distributions for randomized multilinear, we testify the variants of RMAN using Bernoulli, Gaussian, and Uniform distributions as the sampling distributions, which are denoted by RMAN (bernoulli), RMAN (gaussian), and RMAN (uniform), respectively.

We implement all deep methods based on the Caffe [19] framework, and fine-tune from AlexNet [21] and ResNet [16] models pre-trained on the ImageNet dataset [32]. We fine-tune all convolutional and pooling layers and train the classifier layer via back propagation. Since the classifier is trained from scratch, we set its learning rate to be 10 times that of the lower layers. We employ the mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and the learning rate strategy implemented in RevGrad [9]: the learning rate is not selected by a grid search due to high computational cost—it is adjusted during SGD using these formulas: $\eta_p = \frac{\eta_0}{(1 + \gamma p)^{\frac{2}{\tau - \gamma}}}$, where $p$ is the training progress linearly changing from 0 to $\eta_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$, which is optimized to promote convergence and low error on source domain. To suppress noisy activations at the early stages of training, instead of fixing parameter $\lambda$, we gradually change it by multiplying $\frac{2}{(1 + \exp(-\delta p))} - 1$, where $\delta = 10$ [9]. This progressive training strategy significantly stabilizes the parameter sensitivity of the proposed models.

| Method              | A $\rightarrow$ W | D $\rightarrow$ W | W $\rightarrow$ D | A $\rightarrow$ D | D $\rightarrow$ A | W $\rightarrow$ A | Avg  |
|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------|
| AlexNet [21]        | 60.6±0.0          | 95.4±0.2          | 99.0±0.1          | 64.2±0.3          | 45.5±0.5          | 48.3±0.5          | 68.8 |
| TCA [23]            | 59.0±0.0          | 90.2±0.0          | 88.2±0.0          | 57.8±0.0          | 51.6±0.0          | 47.9±0.0          | 65.8 |
| GFK [12]            | 58.4±0.0          | 93.6±0.0          | 91.0±0.0          | 58.6±0.0          | 52.4±0.0          | 46.1±0.0          | 66.7 |
| DDC [33]            | 61.0±0.5          | 95.0±0.3          | 98.5±0.3          | 64.9±0.4          | 47.2±0.5          | 49.4±0.4          | 69.3 |
| DAN [22]            | 68.5±0.3          | 96.0±0.1          | 99.0±0.1          | 66.8±0.2          | 50.0±0.4          | 49.8±0.3          | 71.7 |
| RTN [23]            | 73.3±0.2          | 96.8±0.2          | 99.6±0.1          | 71.0±0.2          | 50.5±0.3          | 51.0±0.1          | 73.7 |
| RevGrad [9]         | 73.0±0.5          | 96.4±0.3          | 99.2±0.3          | 72.3±0.3          | 52.4±0.4          | 50.4±0.5          | 74.1 |
| RMAN (bernoulli)    | 76.7±0.5          | 96.1±0.3          | 100.0±0.0         | 73.1±0.3          | 56.3±0.5          | 54.6±0.5          | 76.3 |
| RMAN (gaussian)     | 77.8±0.4          | 96.7±0.2          | 100.0±0.0         | 74.9±0.2          | 56.4±0.3          | 53.4±0.4          | 76.5 |
| RMAN (uniform)      | 77.9±0.2          | 96.9±0.1          | 100.0±0.0         | 74.6±0.1          | 55.1±0.2          | 57.5±0.3          | 77.0 |

| Method              | A $\rightarrow$ W | D $\rightarrow$ W | W $\rightarrow$ D | A $\rightarrow$ D | D $\rightarrow$ A | W $\rightarrow$ A | Avg  |
|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------|
| ResNet [16]         | 68.4±0.2          | 96.7±0.1          | 99.3±0.1          | 68.9±0.2          | 62.5±0.3          | 60.7±0.3          | 76.1 |
| TCA [28]            | 74.7±0.0          | 96.7±0.0          | 99.6±0.0          | 76.1±0.0          | 63.7±0.0          | 62.9±0.0          | 79.3 |
| GFK [12]            | 74.8±0.0          | 95.0±0.2          | 98.2±0.0          | 76.5±0.0          | 65.4±0.0          | 63.0±0.0          | 78.8 |
| DDC [35]            | 75.8±0.2          | 95.0±0.2          | 98.2±0.1          | 77.3±0.3          | 67.4±0.4          | 64.0±0.5          | 79.7 |
| DAN [22]            | 83.8±0.4          | 96.8±0.2          | 99.5±0.1          | 78.4±0.2          | 66.7±0.3          | 62.7±0.2          | 81.3 |
| RTN [23]            | 84.5±0.2          | 96.8±0.1          | 99.4±0.1          | 77.5±0.3          | 66.2±0.2          | 64.8±0.3          | 81.6 |
| RevGrad [9]         | 82.0±0.4          | 96.9±0.2          | 99.1±0.1          | 79.7±0.4          | 68.2±0.4          | 67.4±0.5          | 82.2 |
| RMAN (bernoulli)    | 84.1±0.3          | 97.4±0.2          | 99.4±0.1          | 83.1±0.4          | 67.9±0.4          | 68.3±0.5          | 83.1 |
| RMAN (gaussian)     | 87.8±0.2          | 97.2±0.2          | 99.4±0.1          | 85.1±0.3          | 70.9±0.3          | 71.7±0.4          | 85.3 |
| RMAN (uniform)      | 88.0±0.1          | 97.4±0.1          | 99.7±0.1          | 86.4±0.2          | 70.6±0.3          | 71.4±0.3          | 85.6 |

### 4.2 Results

The classification accuracy results on the Office-31 dataset for unsupervised domain adaptation based on AlexNet and ResNet are shown in Table 1. For fair comparison, the results of DAN [22], RTN [23], and RevGrad [9] are directly reported from their original papers. The RMAN models outperform all comparison methods on most transfer tasks, where RMAN (uniform) is the top-performing variant and RMAN (gaussian) is the second-best variant, which is consistent with our theoretical result on the approximation quality. It is noteworthy that RMAN promotes the classification accuracies substantially on hard transfer tasks, e.g. $A \rightarrow W$, $A \rightarrow D$, $D \rightarrow A$, and $W \rightarrow A$, where the source
and target domains are substantially different, and produce comparable classification accuracies on easy transfer tasks, $D \rightarrow W$ and $W \rightarrow D$, where the source and target domains are similar \[53\]. The three domains in the ImageCLEF-DA dataset are balanced in each category. As reported in Table 2, the RMAN models outperform the comparison methods on most transfer tasks. The encouraging results highlight the importance of randomized multilinear adaptation in deep neural networks, and suggest that RMAN is able to learn more transferable representations for effective domain adaptation.

The experimental results reveal several insightful observations. (1) Standard deep learning methods (AlexNet and ResNet) either outperform or underperform traditional shallow transfer learning methods (TCA and GFK) using deep features as input. This confirms the current practice that deep networks, even the extremely deep ones (ResNet), can learn abstract feature representations that only reduce but not remove the cross-domain discrepancy \[40\]. (2) Deep transfer learning methods substantially outperform both standard deep learning methods and traditional shallow transfer learning methods with deep features as input. This validates that explicitly reducing the cross-domain discrepancy by embedding domain-adaptation modules into deep networks (DDC, DAN, RTN, and RevGrad) can learn more transferable features. (3) RMAN substantially outperforms previous methods based on either multilayer adaptation (DAN), semi-supervised adaptation (RTN), and domain adversarial training (RevGrad). Although both RMAN and DAN \[22\] adapt multiple domain-specific layers, the improvement from DAN to RMAN is crucial for domain adaptation: DAN imposes multiple MMD penalties, each independently reducing the distribution shift in a single layer; RMAN enables domain adaptation by making the source and target domains indistinguishable for a domain discriminator under the randomized multilinear fusion of deep features and classifier predictions in multiple layers, which can essentially reduce the dataset shift in the joint distributions of multiple task-specific layers.

### 4.3 Analysis

**Feature Visualization:** We go deeper into the feature transferability by visualizing in Figures 2(a)–2(d) the network activations of task $A \rightarrow W$ (31 classes) learned by ResNet (the last feature layer pool5), RevGrad (the bottleneck layer $f_{cb}$), RMAN (the bottleneck layer $f_{cb}$), and RMAN-rm (the randomized multilinear fusion of the bottleneck layer $f_{cb}$ and the the classifier layer $f_{cc}$) respectively.
using t-SNE embeddings [7]. The visualization results reveal several interesting observations. (1) Under ResNet features, the source and target domains are not aligned well, which results in inferior accuracy when classifying the target data using the source classifier. (2) Under RevGrad features, the source and target domains are made indistinguishable; however, different categories are not well discriminated clearly. The reason is that domain adversarial learning is performed only at the feature layer \( f_{cb} \), while the discriminative information \( f_{cc} \) is not taken into account by the domain adversary. (3) Under RMAN features, not only the source and target domains are made more indistinguishable but also different categories are made more discriminated, which implies the best adaptation accuracy. This superior results benefits from the integration of discriminative information \( f_{cc} \) into the domain adversarial learning framework. (4) Under RMAN-rm features, the source and target domains are perfectly aligned and different categories are perfectly discriminated. This is not surprising, because the randomized multilinear fusion of deep features \( f_{cb} \) and classifier predictions \( f_{cc} \) are directly adapted by the proposed RMAN model, which jointly yields more transferable features and classifiers.

**Fusion Strategies:** Besides the proposed randomized multilinear fusion strategy, one may concern on other fusion strategies such as element-wise sum/product, concatenation, etc. We have to note that, due to different dimensions in the domain-specific layers (involving feature layers and the classifier layer), element-wise operations are not applicable. To examine different strategies, we compare RMAN with ResNet, RevGrad-fcb (the domain discriminator is imposed at the last feature layer \( f_{cb} \)), RevGrad-fcc (the domain discriminator is imposed at the classifier layer \( f_{cc} \)), RevGrad-concat (the domain discriminator is imposed on the concatenation of \( f_{cb} \) and \( f_{cc} \)). The accuracy results of tasks \( A \to W \) and \( A \to D \) in Figure 3(a) reveal that the concatenation strategy is not successful for two reasons: (i) concatenation cannot capture the full interactions between deep features and classifier predictions, which are important for domain adaptation; (ii) the feature layers and the classifier layer are of different magnitudes, which require multilinear operations to eliminate magnitudes side-effects.

**Distribution Discrepancy:** The domain adaptation theory [3, 24] suggests \( A \)-distance as a measure of cross-domain discrepancy, which, together with the source risk, will bound the target risk. The proxy \( A \)-distance is defined as \( d_A = 2 (1 - 2 \epsilon) \), where \( \epsilon \) is the generalization error of a classifier (e.g. kernel SVM) trained on the binary task of discriminating source and target. Figure 3(b) shows \( d_A \) on tasks \( A \to W, W \to D \) with features of ResNet, RevGrad, and RMAN. We observe that \( d_A \) using RMAN features is much smaller than \( d_A \) using ResNet and RevGrad features, which suggests that RMAN features can reduce the cross-domain gap more effectively. As domains \( W \) and \( D \) are similar, \( d_A \) of task \( W \to D \) is smaller than that of \( A \to W \), which well explains better accuracy of \( W \to D \).

**Convergence Performance:** Since RMAN involves randomized procedures, we testify the convergence performance of its variants, RMAN (bernoulli), RMAN (gaussian), and RMAN (uniform) in comparison with ResNet and RevGrad. Figure 3(c) shows the test errors of different methods on task \( A \to W \), which suggests that all RMAN variants have similarly stable convergence performance as RevGrad while the ones based on Gaussian and Uniform sampling significantly outperform RevGrad. Furthermore, the computational complexity of RMAN is similar to RevGrad: we set the dimension after the randomized multilinear fusion as 1024, which is a typical size for standard deep networks.

## 5 Conclusion

This paper presented a novel randomized multilinear adversarial adaptation approach to deep transfer learning. Unlike previous adversarial adaptation methods that only match the marginal distributions
of features across domains and may be trapped by the multimode collapse difficulty, the proposed approach further exploits the discriminative structures to enable fine-grained distribution alignment. The discrepancy between joint distributions of feature layers and classifier layer can be computed by a new randomized multilinear adversary. Experiments testified the efficacy of the proposed approach.

6 Appendix: Proof of Theorem 1

This supplemental material provides detailed proof to Theorem 1 in the main paper. To enable better readability, we first echo the randomized feature map for multilinear fusion of different vectors as

$$\phi_L(z) = \frac{1}{\sqrt{d}} \left( \otimes_{\ell}^{|L|} R^\ell z^\ell \right). \tag{8}$$

**Theorem 2.** The expectation and variance of the inner products between the randomized feature maps $\phi_L(z)$ generated by random matrices $R^\ell$, $\ell = 1, \ldots, |L|$ are

$$\mathbb{E} [\langle \phi_L(z), \phi_L(z') \rangle] = \prod_{\ell=1}^{|L|} \langle z^\ell, z'^\ell \rangle, \tag{9}$$

$$\text{var} [\langle \phi_L(z), \phi_L(z') \rangle] = \sum_{i=1}^d \otimes_{\ell}^{|L|} \left[ \sum_{j=1}^{d_{\ell}} \left( z_{ij}^\ell \right)^2 \left( z_{ij}'^\ell \right)^2 \mathbb{E} \left[ (R_{ij}^\ell)^4 \right] + C' \right] + C, \tag{10}$$

where $C$ and $C'$ are constants that do not depend on the random matrices $R^\ell$, $\ell = 1, \ldots, |L|$.

Theorem 1 reveals that the inner product between the randomized feature maps $\phi_L(z)$ is an unbiased estimate of the inner product between the original multilinear fusions based on tensor products $T(z)$. The variance of the inner product between the randomized feature maps $\phi_L(z)$ is depending only on the moments $\mathbb{E}[(R_{ij}^\ell)^4]$, which are constants for many symmetric distributions with uni-variance, $\mathbb{E} [R_{ij}^\ell] = 0$, $\mathbb{E}[(R_{ij}^\ell)^2] = 1$. We can verify that: (1) for Bernoulli distribution, $\mathbb{E} (R_{ij}^\ell)^4 = 1$; (2) for standard normal distribution, $\mathbb{E} (R_{ij}^\ell)^4 = 3$; (3) for uniform distribution, $\mathbb{E} (R_{ij}^\ell)^4 = 1.8$. Therefore, for continuous sampling distributions, uniform distribution will yield the lowest estimation variance. The empirical study confirms that uniform distribution leads to the best multilinear fusion accuracy.

**Proof.**

$$\mathbb{E} [\langle \phi_L(z), \phi_L(z') \rangle] = \mathbb{E} \left[ \left\langle \frac{1}{\sqrt{d}} \left( \otimes_{\ell}^{|L|} R^\ell z^\ell \right), \frac{1}{\sqrt{d}} \left( \otimes_{\ell}^{|L|} R^\ell z'^\ell \right) \right\rangle \right]$$

$$= \frac{1}{d} \mathbb{E} \left[ \left( \otimes_{\ell}^{|L|} R^\ell z^\ell, \otimes_{\ell}^{|L|} R^\ell z'^\ell \right) \right]$$

$$= \frac{1}{d} \mathbb{E} \left[ \sum_{i=1}^d \otimes_{\ell}^{|L|} \left( R_{i1}^\ell z^i \right) \left( R_{i1}^\ell z'^i \right) \right] = \frac{1}{d} \sum_{i=1}^d \mathbb{E} \left[ \otimes_{\ell}^{|L|} \left( R_{i1}^\ell z^i \right) \left( R_{i1}^\ell z'^i \right) \right] \tag{11}$$

$$= \frac{1}{d} \sum_{i=1}^d \left[ \otimes_{\ell}^{|L|} z^T E \left[ R_{i1}^\ell R_{i1}^\ell \right] z'^T \right] = \frac{1}{d} \sum_{i=1}^d \left[ \otimes_{\ell}^{|L|} z^T z'^T \right] = \prod_{\ell=1}^{|L|} \langle z^\ell, z'^\ell \rangle.$$
\[ \text{var} [\langle \phi_L(z), \phi_L(z') \rangle] = \mathbb{E} \left[ (\phi_L(z), \phi_L(z'))^2 \right] - \mathbb{E}[\langle \phi_L(z), \phi_L(z') \rangle]^2 \]

\[ = \mathbb{E} \left[ \left( \frac{1}{\sqrt{d}} \left( \sum_{i=1}^{d} \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right) \right)^2 \right] - \left( \prod_{i=1}^{d} \langle z', z' \rangle \right)^2 \]

\[ = \mathbb{E} \left( \frac{1}{d^2} \left( \sum_{i=1}^{d} \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right)^2 \right) - C_1 \]

\[ = \frac{1}{d^2} \mathbb{E} \left[ \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} E \left( \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right) \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right] - C_1 \]

\[ = \frac{1}{d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} E \left( \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right) \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right] - C_1 \]

\[ = \frac{1}{d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} E \left( \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right) \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right] - C_1 \]

\[ = \frac{1}{d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} E \left( \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right) \phi_L^{[i]}(R_{\ell}^{[i]} z') \phi_L^{[i]}(R_{\ell}^{[i]} z') \right] - C_1 \]

Since the equations in the proof are a bit lengthy, we simply denote any parts of the equations independent on random matrices \( R_{\ell}, \forall \ell \in L \) as constants, e.g. \( C_1 \sim C_4, C \), and \( C' \).

References

[1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In ICLR, 2017.
[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.
[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1798–1828, 2013.
[5] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *ICLR*, 2017.
[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research (JMLR)*, 12:2493–2537, 2011.
[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.
[8] L. Duan, I. W. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(3):465–479, 2012.
[9] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015.
[10] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *International Conference on Machine Learning (ICML)*, 2011.
[11] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2013.
[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.
[14] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.
[15] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*, 13:723–773, 2012.
[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[17] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.
[18] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.
[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *MM*, 2014.
[20] P. Kar and H. Karmick. Random feature maps for dot product kernels. In *AISTATS*, volume 22, pages 583–591, 2012.
[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.
[22] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015.
[23] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 136–144, 2016.
[24] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Computational Learning Theory (COLT)*, 2009.
[25] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *ICLR*, 2017.
[26] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
[28] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks (TNN)*, 22(2):199–210, 2011.
[29] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2010.
[30] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.
[31] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.
[32] 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. 2014.

[33] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In European Conference on Computer Vision (ECCV), 2010.

[34] L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. Smola. Hilbert space embeddings of hidden markov models. In International Conference on Machine Learning (ICML), 2010.

[35] L. Song and B. Dai. Robust low rank kernel embeddings of multivariate distributions. In Advances in Neural Information Processing Systems (NIPS), pages 3228–3236, 2013.

[36] L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In International Conference on Machine Learning (ICML), 2009.

[37] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In Advances in Neural Information Processing Systems (NIPS), 2008.

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

[39] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Simultaneous deep transfer across domains and tasks. In IEEE International Conference on Computer Vision (ICCV), 2015.

[40] 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), 2014.

[41] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In International Conference on Machine Learning (ICML), 2013.

[42] E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In ECML/PKDD, pages 547–562. Springer, 2010.