Revisiting and Accelerating Transfer Component Analysis

Zhanbo Feng[*1] **Yuanjie Wang**[*2] **Jie Li**[1] **Robert C. Qiu**[2]

Zhenyu Liao[†2]
[1]CSE, Shanghai Jiao Tong University, Shanghai, China
[2]EIC, Huazhong University of Science and Technology, Wuhan, China

Abstract

Transfer learning allows one to apply existing knowledge acquired when solving one machine learning problem to a different but related problem, and largely eliminates the need for manual labeling that is both expansive and burdensome. Kernel- and neural network-based models have been proposed to learn invariant (and thus "transferable") features for both labelled source and unlabelled target data. In this paper, we revisit the popular kernel-based unsupervised transfer component analysis (TCA) approach, and show, perhaps surprisingly, that it is closely related to kernel principal component analysis (kernel PCA). Built upon such a connection, we propose a random feature-based TCA (RF-TCA) approach to alleviate the computational burden of the original TCA method when the number of source and/or target data is large. We further exploit matrix concentration results to show that a number of $N = O(n \log(n))$ random features is sufficient for the transferred features obtained from RF-TCA to be close to those from TCA. Experiments on standard datasets testify that the proposed RF-TCA approach yields comparable performance with respect to various kernel- and neural network-based transfer learning schemes, with an average of $50+$ times computational speedup.

1 Introduction

Modern machine learning (ML) has achieved remarkable success in applications ranging from computer vision, natural language processing, to information retrieval and speech recognition. It is believed[1] that modern ML heavily relies on, and owns its success to, the exponentially growing of (the amount of publicly available) data and the exponentially falling cost per unit of computation. While the computational resources may continue to grow, as long as the Moore's law, or its variants, remain valid, this may *not* be the case for the data, and even if they were, such seemingly endless and gratuitous growth should *not* be expected for the labels of those data. For non-trivial tasks such as disease identification, manual labeling of a sufficiently large number of training data requires experts' knowledge and labor, can be both expensive and cumbersome, but is of crucial importance to the performance of ML models. The transfer learning approach was proposed to address this issue [23].

Domain adaption, as an important subcategory of transfer learning, aims to learn invariant features/representations for both source and target data, so that downstream ML tasks can be performed in this *common* feature space. To obtain rich data representations that are efficient and effective for

[*]Equal contribution.

[†]Author to whom any correspondence should be addressed. Email: zhenyu_liao@hust.edu.cn

[1]See, for example the blog of Rich Sutton on the "bitter lesson."

Preprint. Under review.

the ML task at hand, transfer learning paradigm can cooperate with different representation learning approaches such as kernel- [28] and neural network-based [10] methods.

Transfer Component Analysis (TCA) [22] is among the most popular and powerful kernel-based domain adaptation techniques, and aims to "transfer" both source and target data into a common feature space by linearly combing their kernel features using a weight matrix $\mathbf{W}$. In this common feature space, standard ML approaches (such as classifiers learned with labelled source features) can then be applied on unlabelled target features. However, kernel methods are known to suffer from scalability issues in large-scale problems due to their huge space and time complexity with respect to the number of data n. Concretely, for a kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ constructed from n data points, kernel spectra-based learning approaches such as kernel TCA, kernel spectral clustering [19], and kernel principal component analysis (PCA) [27] generally suffer from an $O(n^2)$ complexity with Lanczos iteration [8] and quickly becomes burdensome for n large, which is a common practice today.

In this paper, we focus on unsupervised TCA, discuss its computational and regularization properties, and show its close connection to the kernel PCA approach [27]. To address the scalability issue of kernel TCA for n large, we leverage random features-based kernel approximation technique proposed in [24] to design a random features-based TCA, referred to as RF-TCA, that shows huge savings in both space and time, with virtually no performance degradation. We further exploit standard concentration results to derive bounds for the difference between the learned *transfer components* of RF-TCA and those of the original TCA, thereby providing theoretical guarantee for the proposed RF-TCA method. Experiments on Office-Caltech [9] and Office-31 datasets [25] are performed, demonstrating the numerical advantages of RF-TCA with comparable performance and an average of $50+$ times computational speedup with respect to various kernel- and neural network-based transfer learning methods.

Our main results are summarized as follows.

1. We discuss, in Section 3, the computational and regularization properties of the original TCA and how the regularization penalty γ impacts the transfer learning performance.
2. We propose, in Section 4, an alternative regularization scheme for TCA, by penalizing the "transferring" weight matrix $\mathbf{W}$, but in the kernel feature space.
3. This further allows us to propose, in Algorithm 1 of Section 5, the random feature-based TCA approach (RF-TCA) that significantly accelerates the original TCA computation, with virtually no performance degradation. We then exploit standard matrix concentration result in Theorem 1 to show, in Theorem 2, that it suffices to have $N = O(n \log(n))$ random features for the proposed RF-TCA approach to match the performance of TCA.
4. Experiments are conducted on the popular Office-Caltech and Office-31 datasets, showing an average of $50+$ times computational speedup with respect to other baseline transfer learning methods, see Figure 3.

Notations. We denote scalars by lowercase letters, vectors by bold lowercase, and matrices by bold uppercase. We denote the transpose operator by $(\cdot)^\mathsf{T}$, and use $\|\cdot\|$ to denote the Euclidean norm for vectors and spectral/operator norm for matrices. For a random variable z, $\mathbb{E}[z]$ denotes the expectation of z. We use $\mathbf{1}_p$ and $\mathbf{I}_p$ for the vector of all one's of dimension p and the identity matrix of dimension $p \times p$, respectively. We denote n_S the number of source data, n_T the number of target data, and $n = n_S + n_T$ the total number of available data, respectively. We use Θ, O and Ω notations as in classical computer science literature.

2 Related works and preliminaries

Here, we provide a brief review of related previous efforts, with a particular focus on the TCA and random features approach in Section 2.1 and 2.2, respectively.

Transfer learning and domain adaptation. Transfer learning, and in particular, domain adaptation schemes aims to "align" source and target data representations in a common features space. In this respect, Geodesic Flow Kernel (GFK) [9] exploits intrinsic low-dimensional structures of the data to perform subspace feature alignment; while CoRelation Alignment (CORAL) [30] generates novel

data representations after feature alignment by exploring their second-order statistics. Different feature alignment strategies can also be considered: For example, Joint Distribution Adaptation (JDA) [17] minimizes the difference between source and target marginal distributions while preserving the "structure" of the data; and Domain Adaptive Neural Network (DaNN) [7] combines neural network with the Maximum Mean Discrepancy (MMD) principle [29, 11] to reduce the source-target distribution mismatch in some latent space.

Kernel method and random features. Random features methods were first proposed to relieve the computational and storage burden of kernel methods in large-scale problems when the number of data n is large [28, 24, 16]. For instance, Random Fourier Features can be used to approximate the popular Gaussian kernel, with a sufficiently large N number of random features [24, 15]. Other random features-based techniques have also been proposed for more involved kernels, e.g., [32], and we refer interested readers to [16] for a review of these approaches.

Spectra method in machine learning. Spectral method, as a simple and surprisingly effective scheme to extract useful information from massive and noisy data, is playing an important role in modern statistics, machine learning, and data science, with successful applications ranging from dimension reduction (e.g., PCA), to community detection in networks [20], clustering [19], and ranking [3], to name a few [2]. In this respect, random matrix theory emerges as a powerful and flexible tool to assess and improve these methods when large dimensional data are considered [4].

2.1 Transfer component analysis

The transfer component analysis (TCA) approach [22] proposes to align the (features of the) source data $\mathbf{x}_S \in \mathbb{R}^p$ and target data $\mathbf{x}_T \in \mathbb{R}^p$ via the following two-step transformation:

1. "Transform" source $\mathbf{x}_S \mapsto \phi(\mathbf{x}_S)$ and target data $\mathbf{x}_T \mapsto \phi(\mathbf{x}_T)$ via the kernel feature map $\phi(\cdot)\colon \mathbb{R}^p \to \mathcal{H}$ that maps the input (raw) data into some kernel feature space $\mathcal{H}$ (e.g., the so-called Reproducing Kernel Hilbert Space, RKHS [28]), to form the source-target kernel matrix
$$\mathbf{K} \equiv \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{i,j=1}^n \in \mathbb{R}^{n \times n}, \tag{1}$$
with $n_S + n_T = n$, $\langle , \rangle$ denotes the dot product in $\mathcal{H}$; and
2. find, in an unsupervised manner, a *linear* transformation $\mathbf{W} \in \mathbb{R}^{n \times m}$ that "projects" the kernel features of both source and target data onto a low dimensional space $\mathbb{R}^m$ with $m \ll n$ (that can be, for example, computed or stored much more efficiently), according to the Maximum Mean Discrepancy (MMD) principle [29, 11], by solving the following constrained (and regularized) trace optimization problem
$$\begin{aligned} \min_{\mathbf{W} \in \mathbb{R}^{n \times m}} & L_\gamma(\mathbf{W}) = \operatorname{tr}(\mathbf{W}^\mathsf{T} \mathbf{K} \mathbf{y} \mathbf{y}^\mathsf{T} \mathbf{K} \mathbf{W}) + \gamma \operatorname{tr}(\mathbf{W}^\mathsf{T} \mathbf{W}), \\ \text{s.t.} \quad & \mathbf{W}^\mathsf{T} \mathbf{K} \mathbf{H} \mathbf{K} \mathbf{W} = \mathbf{I}_m \end{aligned} \tag{2}$$
with some regularization penalty $\gamma \geq 0$, $\mathbf{H} = \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\mathsf{T} \in \mathbb{R}^{n \times n}$, and $\mathbf{y} \in \mathbb{R}^n$ the (normalized) "label" vector $\mathbf{y} \in \mathbb{R}^n$ with its ith entry given by
$$[\mathbf{y}]_i = \frac{1}{n_S} 1_{\mathbf{x}_i \in \mathbf{X}_S} - \frac{1}{n_T} 1_{\mathbf{x}_i \in \mathbf{X}_T}, \tag{3}$$
for $\mathbf{X}_S \in \mathbb{R}^{p \times n_S}$ the (set of) source data and $\mathbf{X}_T \in \mathbb{R}^{p \times n_T}$ the (set of) target data, so that $\|\mathbf{y}\|^2 = \frac{n}{n_S n_T}$. The solution to the optimization problem in (2) is explicitly given by the m top eigenvectors (that correspond to the largest m eigenvalues) of $(\gamma \mathbf{I}_n + \mathbf{K} \mathbf{y} \mathbf{y}^\mathsf{T} \mathbf{K})^{-1} \mathbf{K} \mathbf{H} \mathbf{K}$, that is, $\mathbf{W} \in \mathbb{R}^{n \times m}$ satisfying
$$(\gamma \mathbf{I}_n + \mathbf{K} \mathbf{y} \mathbf{y}^\mathsf{T} \mathbf{K})^{-1} \mathbf{K} \mathbf{H} \mathbf{K} \cdot \mathbf{W} = \mathbf{W} \cdot \mathbf{\Lambda} \tag{4}$$
with diagonal $\mathbf{\Lambda} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_m\} \in \mathbb{R}^{m \times m}$ containing the largest m eigenvalues of $(\gamma \mathbf{I}_n + \mathbf{K} \mathbf{y} \mathbf{y}^\mathsf{T} \mathbf{K})^{-1} \mathbf{K} \mathbf{H} \mathbf{K}$, and the resulting low dimensional *common* representations of both source and target data are the columns of $\mathbf{W}^\mathsf{T} \mathbf{K} \in \mathbb{R}^{m \times n}$.

This two-step feature "transformation" is visually displayed in Figure 1.

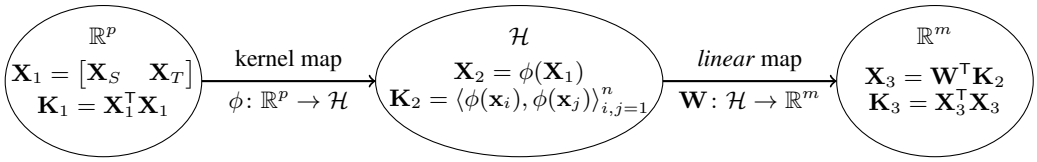


Figure 1: The two-step feature "transformation" of TCA.

2.2 Random features maps

In [24] the authors proposed the random Fourier features (RFFs) scheme to approximate shift-invariant, so specifically, Gaussian kernel matrices. More precisely, the random Fourier features (RFFs) matrix of data $\mathbf{X} \in \mathbb{R}^{p\times n}$ is given by

$$\mathbf{\Sigma} = \frac{1}{\sqrt{N}}\begin{bmatrix}\cos(\mathbf{\Omega X})\\ \sin(\mathbf{\Omega X})\end{bmatrix} \in \mathbb{R}^{2N\times n}, \quad [\mathbf{\Omega}]_{ij} \sim \mathcal{N}(0, 1/\sigma^2) \tag{5}$$

with N the number of random features, $\mathbf{\Omega} \in \mathbb{R}^{N\times p}$ having i.i.d. standard Gaussian entries with mean zero and variance $1/\sigma^2$, and nonlinear functions $\cos(\cdot), \sin(\cdot)$ applied entry-wise. It has been shown in [24] that, for a sufficiently large N number of random features, the RFF Gram matrix can "well approximate" the Gaussian kernel matrix in some sense, that is

$$\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} \approx \mathbf{K}_{\mathrm{Gauss}}, \quad [\mathbf{K}_{\mathrm{Gauss}}]_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)). \tag{6}$$

The case of spectral norm approximation is *quantitatively* characterized in the following result.

Theorem 1 (RFFs approximation, [31, Section 6.5])**.** *For random Fourier features $\mathbf{\Sigma} \in \mathbb{R}^{2N\times n}$ defined in* (5)*, and the associated Gaussian kernel matrix $\mathbf{K}_{\mathrm{Gauss}} \in \mathbb{R}^{n\times n}$ defined in* (6)*, one has,*

$$\mathbb{E}\|\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} - \mathbf{K}_{\mathrm{Gauss}}\| \leq C\left(\sqrt{\frac{n}{N}\|\mathbf{K}_{\mathrm{Gauss}}\|\log(n)} + \frac{n}{N}\log(n)\right) \tag{7}$$

holds for some universal constant $C > 0$.

It then follows from a direct application of Markov's inequality ($\Pr(X \geq a) \leq \mathbb{E}[X]/a$) that, for any given $\delta \in (0,1)$ and $\varepsilon \in (0,1)$, it suffices to have a number

$$N \geq C(\|\mathbf{K}_{\mathrm{Gauss}}\| + 1)\frac{n\log(n)}{\delta\varepsilon} \tag{8}$$

random features so that the event $\|\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} - \mathbf{K}_{\mathrm{Gauss}}\| \leq \varepsilon$ happens with probability at least $1-\delta$. In plain words, if the number of random features N is sufficiently large (compared to, e.g., $n\log(n)$), then the random feature Gram matrix $\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}$ is, with high probability, a good approximation of the Gaussian kernel matrix $\mathbf{K}_{\mathrm{Gauss}}$, in a spectral norm sense.[2]

Here and in the remainder of this paper, we focus on the case of RFF as a illustrating example, but the proposed analysis and RF-TCA method extends *directly* at least to Cauchy, Laplacian, and at other types of shift-invariant kernels [24]. We refer the interested readers to [16] for a review of general random features-based approaches and to Section C for more numerical results and discussions.

3 A few facts about vanilla TCA

To distinguish the TCA approach originally proposed in [22] from the upcoming "variants" of it, we will, in the remainder of this paper, refer the original TCA formulation (as recalled in Section 2.1 above) as the *vanilla* TCA.

For the ease of exposition, we position ourselves under the following technical assumption.

Assumption 1 (On kernel matrix)**.** *Let the kernel matrix $\mathbf{K}$ be defined as in* (1) *and let $\lambda_{\min}(\mathbf{K})$ and $\lambda_{\max}(\mathbf{K})$ be its minimum and maximum eigenvalue, respectively. Then, there exists some universal constants $C_{\mathbf{K}} \geq c_{\mathbf{K}} > 0$* independent *of n, p such that $0 < c_{\mathbf{K}} \leq \lambda_{\min}(\mathbf{K}) \leq \lambda_{\max}(\mathbf{K}) \leq C_{\mathbf{K}}$.*

[2] As a technical side remark, the dependence on δ, ε can be improved by establishing a higher-order (beyond the first-order moment) version of Theorem 1.

Assumption 1 is rather standard in kernel learning literature. As an example, in the case of Gaussian kernel matrix $\mathbf{K} = \mathbf{K}_{\text{Gauss}} = \{\exp(\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2))\}_{i,j=1}^n$, the lower bound holds when the data vectors are linearly independent (see, e.g., [28, Theorem 2.18]), and the upper bound holds, at least with high probability, when the input data have i.i.d. sub-Gaussian entries, see [14, Section 3], or the kernel eigenvalues decay sufficiently slowly (since $\operatorname{tr} \mathbf{K}_{\text{Gauss}} = n$) and corresponds to a large Reproducing Kernel Hilbert Space, see [1].

Note that for the unsupervised TCA formulation in (2) and (4), one has $\mathbf{Hy} = \mathbf{y}$ so that by pre-multiplying (4) by $\mathbf{HK}$, one gets

$$\mathbf{HK}(\gamma \mathbf{I}_n + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{KH} \cdot \mathbf{HKW} = \mathbf{HKW} \cdot \mathbf{\Lambda} \tag{9}$$

so that $\mathbf{HKW} \in \mathbb{R}^{n \times m}$ is the top eigenspace that corresponds to the largest m eigenvalues of the *symmetric* matrix $\mathbf{HK}(\gamma \mathbf{I}_n + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{KH}$ (the eigenvalues of which coincide with those of $(\gamma \mathbf{I}_n + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{KHK}$), where we used the fact that $\mathbf{H}$ is a projection matrix with $\mathbf{H}^2 = \mathbf{H}$.

Since $\mathbf{y}\mathbf{y}^\mathsf{T}$ is a rank-one matrix, one has the following result, as a direct consequence of the Sherman–Morrison formula (i.e., Lemma 2 in Section A of the appendix), that further simplifies the vanilla TCA formulation.

Lemma 1 (Equivalent form of vanilla TCA)**.** *The matrix $\gamma \mathbf{I}_n + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}$ is invertible if and only if $\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y} \neq 0$, and*

$$\left(\gamma \mathbf{I}_n + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\right)^{-1} = \frac{1}{\gamma}\left(\mathbf{I}_n - \frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}\right), \tag{10}$$

so that

$$\mathbf{K}(\gamma \mathbf{I}_n + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K} = \frac{1}{\gamma}\left(\mathbf{K}^2 - \frac{\mathbf{K}^2\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}^2}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}\right). \tag{11}$$

The advantage of the formulation in Lemma 1 is that one no longer needs to compute the matrix inverse ($O(n^3)$ time complexity), or to solve the generalized eigenvector (that can be solved, but only approximately, in $O(n^2)$ time, see, e.g., [6, Theorem 2]) for vanilla TCA in (4), but only to perform matrix additions and, e.g., Lanczos iteration that takes $O(n^2)$ time [8] to retrieve the top eigenvectors.

With Lemma 1, we have the following remark on the regularization parameter γ for vanilla TCA.

Remark 1 (On regularization of vanilla TCA)**.** *By definition in* (3) *and under Assumption 1, one has $\|\mathbf{y}\|^2 = \frac{n}{n_S n_T}$, and that the (only) non-zero eigenvalue of $\frac{\mathbf{K}^2\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}^2}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}$ is $\frac{\mathbf{y}^\mathsf{T}\mathbf{K}^4\mathbf{y}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}$ with*

$$\frac{c_{\mathbf{K}}^4 n}{\gamma n_S n_T + c_{\mathbf{K}}^2 n} \leq \frac{\mathbf{y}^\mathsf{T}\mathbf{K}^4\mathbf{y}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}} \leq \frac{C_{\mathbf{K}}^4 n}{\gamma n_S n_T + C_{\mathbf{K}}^2 n}. \tag{12}$$

As such,

1. *if the number of source and target data are "balanced", in the sense that both n_S, n_T are of order $\Theta(n)$, then the rank-one matrix $\frac{\mathbf{K}^2\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}^2}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}$ is of spectral norm order $\Theta(1)$, and thus "on even ground" with $\mathbf{K}^2$ under Assumption 1,* if and only if *one sets $\gamma = \Theta(n^{-1})$; and*
2. *if the number of source and target data are "unbalanced" with $n_S = \Theta(1)$ and $n_T = \Theta(n)$, or $n_S = \Theta(n)$ and $n_T = \Theta(1)$ (recall that $n_S + n_T = n$), then $\frac{\mathbf{K}^2\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}^2}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}$ is "on even ground" with $\mathbf{K}^2$ in a spectral norm sense* if and only if $\gamma = \Theta(1)$.

The above remark is of direct algorithmic use in the search of optimal regularization parameter γ for vanilla TCA and inspires the following alternative TCA regularization scheme.

4 An alternative regularization scheme for TCA: R-TCA

Instead for regularizing the Frobenius norm of the linear transformation $\mathbf{W} \in \mathbb{R}^{n \times m}$ as in the case of vanilla TCA [21], one may want to regularize the weight matrix $\tilde{\mathbf{W}} = \mathbf{K}^{1/2}\mathbf{W} \in \mathbb{R}^{n \times m}$ that acts

on the empirical kernel map $\mathbf{K}^{1/2}$. This approach, referred to as Regularized TCA (R-TCA), aims to solve the following optimization problem:

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times m}} \tilde{L}_\gamma(\mathbf{W}) = \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{W}) + \gamma \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{K}\mathbf{W}), \quad \text{s.t.} \quad \mathbf{W}^\mathsf{T}\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I}_m \tag{13}$$

and differs from the vanilla TCA formulation in (2) in the regularization term $\gamma \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{K}\mathbf{W})$. It can be easily checked that, under Assumption 1, the TCA and R-TCA approaches are closely related. Precisely, for a given $\mathbf{W} \in \mathbb{R}^{n\times m}$, one has

$$\begin{aligned} L_{\gamma c_\mathbf{K}}(\mathbf{W}) &= \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{W}) + \gamma c_\mathbf{K} \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{W}) \\ &\le \tilde{L}(\mathbf{W}) \le \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{W}) + \gamma C_\mathbf{K} \operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{W}) = L_{\gamma C_\mathbf{K}}(\mathbf{W}) \end{aligned}$$

so that the two optimization problems in (2) and (13) are "equivalent," in terms of the loss function, up to a proper scaling of the regularization parameter γ (that depends on the minimum and maximum eigenvalues of the kernel matrix $\mathbf{K}$).

As in the case for vanilla TCA, the Regularized TCA approach in (13) also admits an *explicit* solution $\mathbf{W} \in \mathbb{R}^{n\times m}$ given by the top m eigenvector of $(\gamma\mathbf{K} + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K}\mathbf{H}\mathbf{K}$ (the proof of which follows from that of [22, Proposition 1], and is omitted here), that is, $\mathbf{W}$ satisfying

$$\mathbf{A}\cdot\mathbf{H}\mathbf{K}\mathbf{W} \equiv \mathbf{H}\mathbf{K}(\gamma\mathbf{K} + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K}\mathbf{H}\cdot\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{H}\mathbf{K}\mathbf{W}\cdot\mathbf{\Lambda}, \tag{14}$$

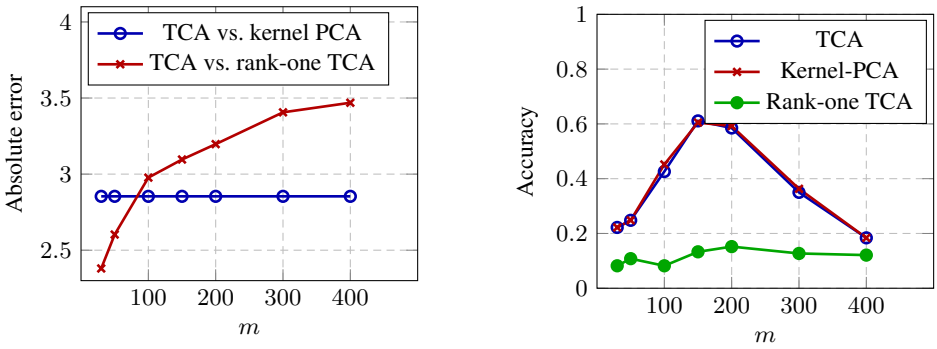
where we denote, with a slight abuse of notations, $\mathbf{\Lambda} \in \mathbb{R}^{m\times m}$ the diagonal matrix containing the largest m eigenvalues of $(\gamma\mathbf{K}+\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K}\mathbf{H}\mathbf{K}$. It then follows again from the Sherman–Morrison formula (as in the proof of Lemma 1) that

$$\mathbf{K}(\gamma\mathbf{K} + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K} = \frac{1}{\gamma}\left(\mathbf{K} - \frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}\right), \tag{15}$$

so as in Lemma 1, one does *not* to inverse the n-by-n matrix in solving R-TCA.

Comparing (15) to (11), one observes that the two forms are almost the same, by replacing $\mathbf{K}^2$ in (11) with $\mathbf{K}$. Also, since $\frac{c_\mathbf{K}^2 n}{\gamma n_S n_T + c_\mathbf{K} n} \le \frac{\mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}} \le \frac{C_\mathbf{K}^2 n}{\gamma n_S n_T + C_\mathbf{K} n}$, depending on the number of source or target data n_S and n_T, the scaling of regularization γ as in Remark 1 should also be adopted.

In both vanilla TCA and R-TCA, we see that the kernel matrix $\mathbf{K}$ or its square $\mathbf{K}^2$ plays an important role. Indeed, it follows from (9) and (14) that for both vanilla TCA and R-TCA, the obtained "transferred" features are strongly connected to the (top m) eigenvectors of $\mathbf{K}^2$ or $\mathbf{K}$, that are essentially the same and are identical to the kernel PCA solution [26].



(a) Absolute error between features

(b) Accuracy of different approaches

Figure 2: Results of transfer from Webcam to Dslr on Decaf6 features of Office-Caltech [9] with a SVM classifier, see more experimental details in Section 6 below. Absolute spectral norm error (**left**) and accuracy (**right**) between different transferred features, as a function of the latent space size m.

To have an empirical grasp of the impact of the kernel matrix $\mathbf{K}^2$ and of the rank-one matrix $\frac{\mathbf{K}^2\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}^2}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}$ (that contains the crucial source-target "label" information $\mathbf{y}$!), we compare, in Figure 2, the absolute

error and the resulting classification accuracy of the "transferred" features obtained from vanilla TCA, kernel PCA (so by using the top m eigenvectors of the centered kernel matrix $\mathbf{HKH}$), as well as the "rank-one TCA" that exploits the top m eigenvectors of $\frac{\mathbf{HK}^2\mathbf{yy}^\mathsf{T}\mathbf{K}^2\mathbf{H}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}^2\mathbf{y}}$ (which is of rank one, so taking $m > 1$ does not, in theory, bring additional information). In the case of Figure 2, we see that the major power of vanilla TCA comes from the top eigenvectors of $\mathbf{K}^2$ (so those of $\mathbf{K}$, and thus the kernel PCA solution). In other scenarios (see, e.g., Section C in the appendix), we also observe the case where vanilla TCA outperforms kernel PCA by a large margin. This motivates the following random features-improved TCA method.

5 A random feature-based TCA scheme

In this section, we present a random features-based approach to computationally more efficient TCA, by focusing on the example of random Fourier features (RFFs) in the case of Gaussian kernel. According to our discussions in Section 2.2, the RFFs of data $\mathbf{X} \in \mathbb{R}^{p\times n}$ is *explicitly* given by $\mathbf{\Sigma}^\mathsf{T} = \left[\cos(\mathbf{\Omega X})^\mathsf{T} \quad \sin(\mathbf{\Omega X})^\mathsf{T}\right]/\sqrt{N} \in \mathbb{R}^{n\times 2N}$, with $[\mathbf{\Omega}]_{ij} \sim \mathcal{N}(0, 1/\sigma^2)$, with *empirical* kernel matrix given by $\hat{\mathbf{K}} = \mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$, which approximates the desired Gaussian kernel matrix $\mathbf{K}_{\rm Gauss}$ for N sufficiently large, as quantitatively characterized in Theorem 1.

Following the idea of the vanilla TCA in [22], one aims to find a matrix $\mathbf{W}_{\rm RF} \in \mathbb{R}^{2N\times m}$ that "projects" the RFFs $\mathbf{\Sigma}$ onto an m-dimensional space, with $m \ll \min(2N, n)$, to obtain the "transferred" representations $\mathbf{W}_{\rm RF}^\mathsf{T}\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ with associated matrix $\tilde{\mathbf{K}} = \mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\rm RF}\mathbf{W}_{\rm RF}^\mathsf{T}\mathbf{\Sigma}$, see again Figure 1. As a result, the (unsupervised) random feature-based TCA (RF-TCA) approach writes

$$\begin{aligned}\mathbf{W}_{\rm RF} = \operatorname{argmin}_{\mathbf{W}\in\mathbb{R}^{2N\times m}} &\operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T}\mathbf{W}) + \gamma\operatorname{tr}(\mathbf{W}^\mathsf{T}\mathbf{W}) \\ \text{s.t.} \quad &\mathbf{W}^\mathsf{T}\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T}\mathbf{W} = \mathbf{I}_m \end{aligned} \tag{16}$$

with some regularization penalty $\gamma \geq 0$, and $\mathbf{H}, \mathbf{L} = \mathbf{yy}^\mathsf{T}$ as defined in vanilla TCA in Section 3.

Let $\tilde{\mathbf{W}} \equiv (\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T})^{\frac{1}{2}}\mathbf{W} \in \mathbb{R}^{2N\times m}$, the optimization in (16) writes

$$\min_{\mathbf{W}\in\mathbb{R}^{2N\times m}} \operatorname{tr}\left(\tilde{\mathbf{W}}^\mathsf{T}(\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T})^{-\frac{1}{2}}(\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})(\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T})^{-\frac{1}{2}}\tilde{\mathbf{W}}\right) \quad \text{s.t. } \tilde{\mathbf{W}}^\mathsf{T}\tilde{\mathbf{W}} = \mathbf{I}_m \tag{17}$$

which also takes the standard form of a trace minimization problem, see [22, Proposition 1] and [19]. It thus follows from the Rayleigh-Ritz theorem that the optimal solution $\mathbf{W}_{\rm RF}$ is given by the top m eigenvectors of $(\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1}\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T}$ associated to its m largest eigenvalues.

Again, since $\mathbf{L} = \mathbf{yy}^\mathsf{T}$ is of rank one, using Lemma 2 we obtain

$$(\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1} = (\mathbf{\Sigma yy}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1} = \frac{1}{\gamma}\left(\mathbf{I}_{2N} - \frac{\mathbf{\Sigma yy}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma y}}\right). \tag{18}$$

This remark leads to the (simplified) RF-TCA procedure summarized as follows.

Algorithm 1: Unsupervised random random-based TCA (RF-TCA)

Input: Source data $\mathbf{X}_S \equiv \{\mathbf{x}_{Si}\}_{i=1}^{n_S}$, target data $\mathbf{X}_T \equiv \{\mathbf{x}_{Tj}\}_{j=1}^{n_T}$, and feature dimension m.

Output: "Invariant" features $\mathbf{F}_{\rm RF} = [\mathbf{F}_S \quad \mathbf{F}_T] \in \mathbb{R}^{m\times n}$ obtained via RF-TCA.

1 Compute RFFs $\mathbf{\Sigma} \in \mathbb{R}^{2N\times n}$ of source and target data $\mathbf{X} = [\mathbf{X}_S \quad \mathbf{X}_T] \in \mathbb{R}^{p\times n}$ as in (5);

2 Compute $\mathbf{W}_{\rm RF} \in \mathbb{R}^{2N\times m}$ the m dominant eigenvectors (that correspond to the largest m eigenvalues) of $\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T} - \frac{\mathbf{\Sigma yy}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma y}}$ for $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\mathsf{T}$ and $\mathbf{y} \in \mathbb{R}^n$ defined in (3);

3 **return** RF-TCA "transferred" features $\mathbf{F}_{\rm RF} = \mathbf{W}_{\rm RF}^\mathsf{T}\mathbf{\Sigma}$.

Similar to Lemma 1 for vanilla TCA and (15) for R-TCA, it can also be seen that the transferred features obtained by RF-TCA are, up to a centering via $\mathbf{H}$, the columns of $\mathbf{W}_{\rm RF}^\mathsf{T}\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ that

$$\mathbf{A}_{\rm RF}\cdot\mathbf{H}\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\rm RF} \equiv \mathbf{H}\mathbf{\Sigma}^\mathsf{T}(\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1}\mathbf{\Sigma H}\cdot\mathbf{H}\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\rm RF} = \mathbf{H}\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\rm RF}\cdot\mathbf{\Lambda}_{\rm RF} \tag{19}$$

with diagonal $\mathbf{\Lambda}_{\rm RF} \in \mathbb{R}^{m\times m}$ the largest m eigenvalues of $(\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1}\mathbf{\Sigma H}\mathbf{\Sigma}^\mathsf{T}$, and thus of the symmetric matrix

$$\mathbf{\Sigma}^\mathsf{T}(\mathbf{\Sigma L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1}\mathbf{\Sigma} = \frac{1}{\gamma}\left(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} - \frac{\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma yy}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma y}}\right), \tag{20}$$

again up to row and column centering via $\mathbf{H}$ on the left- and right-hand side. The RF-TCA formulation in (20) is reminiscent of the R-TCA formulation in (15), since the RFF Gram matrix $\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}$ well approximates the Gaussian kernel matrix $\mathbf{K}_{\text{Gauss}} = \mathbf{K}$ in a spectral norm sense for N sufficiently, according to Theorem 1. It turns out the two formulations are, for N large, also close in a spectral norm sense, as precisely given in the following result, the proof of which follows from an almost immediate application of Theorem 1 and is deferred to Section B.1 of the appendix.

Corollary 1. *For a given data matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$, define the associated Gaussian kernel matrix $\mathbf{K}_{\text{Gauss}} = \mathbf{K} \in \mathbb{R}^{n\times n}$ as in* (6) *and the random Fourier features matrix $\mathbf{\Sigma} \in \mathbb{R}^{2N\times n}$ as in* (5)*. Then, under Assumption 1, for any given $\delta \in (0,1)$ and $\varepsilon \in (0, \min(c_{\mathbf{K}}, 1))$, there exists a universal constant $C > 0$ that only depends on $c_{\mathbf{K}}$ and $C_{\mathbf{K}}$ such that if $N \geq C\frac{n\log(n)}{\delta\varepsilon}$, then $\|\mathbf{K} - \frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}} - (\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} - \frac{\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}})\| \leq \varepsilon$ holds true with probability at least $1-\delta$.*

With Corollary 1 at hand, we are in position to present our result on the performance guarantee for the proposed RF-TCA approach. To that end, the following technical assumption, often referred to as the eigen-gap condition [13, 19], is needed.

Assumption 2. *For symmetric matrix $\mathbf{A} \equiv \mathbf{H}\mathbf{K}(\gamma\mathbf{K} + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K}\mathbf{H}$ defined in* (14)*, denote $\lambda_1(\mathbf{A}) > \cdots > \lambda_m(\mathbf{A})$ its largest m eigenvalues listed in a non-increasing order, then there exists a universal constant $\Delta_{\mathbf{K}} > 0$* independent *of n such that $\min_{1\leq i\leq m} |\lambda_i(\mathbf{A}) - \lambda_{i-1}(\mathbf{A})| \geq \Delta_{\mathbf{K}}$.*

Theorem 2 (Performance guarantee for RF-TCA)**.** *For the random Fourier features matrix $\mathbf{\Sigma} \in \mathbb{R}^{2N\times n}$ defined in* (5) *and the associated kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ defined in* (6) *that satisfy Assumption 1 and 2, let the TCA transferring matrix $\mathbf{W} \in \mathbb{R}^{n\times m}$ be defined in* (13) *and let the associated RF-TCA transferring matrix $\mathbf{W}_{RF} \in \mathbb{R}^{2N\times m}$ be defined in* (16)*. One has, for any given $\delta \in (0,1)$ and $\varepsilon \in (0, \min(c_{\mathbf{K}}, 1))$ that, there exists a universal constant $C > 0$ only depends on $c_{\mathbf{K}}$ and $C_{\mathbf{K}}$ such that if $N \geq C\frac{n\log(n)\sqrt{m}}{\Delta_{\mathbf{K}}\delta\varepsilon}$, then*

$$\|\mathbf{H}\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\text{RF}} - \mathbf{H}\mathbf{K}\mathbf{W}\|_F \leq \varepsilon \tag{21}$$

holds true with probability at least $1-\delta$.

Proof of Theorem 2. Note that the columns of $\mathbf{H}\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\text{RF}} \in \mathbb{R}^{n\times m}$ and $\mathbf{H}\mathbf{K}\mathbf{W} \in \mathbb{R}^{n\times m}$ are in fact the top m eigenvectors $\mathbf{u}_i(\mathbf{A}_{\text{RF}}), \mathbf{u}_i(\mathbf{A}) \in \mathbb{R}^n, i \in \{1,\ldots,m\}$, of $\mathbf{A}_{\text{RF}} \equiv \mathbf{H}\mathbf{\Sigma}^\mathsf{T}(\mathbf{\Sigma}\mathbf{L}\mathbf{\Sigma}^\mathsf{T} + \gamma\mathbf{I}_{2N})^{-1}\mathbf{\Sigma}\mathbf{H}$ and $\mathbf{A} \equiv \mathbf{H}\mathbf{K}(\gamma\mathbf{K} + \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})^{-1}\mathbf{K}\mathbf{H}$, defined respectively in (19) and (14), so the Frobenius norm error satisfies

$$\|\mathbf{H}(\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\text{RF}} - \mathbf{K}\mathbf{W})\|_F^2 = \sum_{i=1}^m \|\mathbf{u}_i(\mathbf{A}_{\text{RF}}) - \mathbf{u}_i(\mathbf{A})\|^2. \tag{22}$$

It then follows from Davis–Kahan theorem [33], Theorem 4 in Appendix A, that

$$\begin{aligned}\|\mathbf{u}_i(\mathbf{A}_{\text{RF}}) - \mathbf{u}_i(\mathbf{A})\| &\leq \sqrt{2}\sin\Theta\left(\mathbf{u}_i(\mathbf{A}_{\text{RF}}), \mathbf{u}_i(\mathbf{A})\right) \\ &\leq \frac{2\sqrt{2}\,\|\mathbf{A}_{\text{RF}} - \mathbf{A}\|}{\min\{|\lambda_{i-1}(\mathbf{A}) - \lambda_i(\mathbf{A})|, |\lambda_{i+1}(\mathbf{A}) - \lambda_i(\mathbf{A})|\}},\end{aligned} \tag{23}$$

with '$\sin\Theta(\mathbf{u}_1, \mathbf{u}_2)$ the "sine similarity" between two vectors $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n$ with $\Theta(\mathbf{u}_1, \mathbf{u}_2) \equiv \arccos\left(\frac{\mathbf{u}_1^\mathsf{T}\mathbf{u}_2}{\|\mathbf{u}_1\|\cdot\|\mathbf{u}_2\|}\right)$ that satisfies $\|\mathbf{u}_1 - \mathbf{u}_2\| \leq \sqrt{2}\Theta(\mathbf{u}_1, \mathbf{u}_2)$.

Further note that under Assumption 2 for $\|\mathbf{K} - \frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}} - (\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma} - \frac{\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}})\| \leq \varepsilon$, one has

$$\|\mathbf{A}_{\text{RF}} - \mathbf{A}\| \leq \|\mathbf{H}\| \cdot \varepsilon \cdot \|\mathbf{H}\| \leq \varepsilon, \tag{24}$$

where we use the fact that $\|\mathbf{H}\| = 1$, and therefore

$$\|\mathbf{H}(\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\text{RF}} - \mathbf{K}\mathbf{W})\|_F^2 = \sum_{i=1}^m \|\mathbf{u}_i(\mathbf{A}_{\text{RF}}) - \mathbf{u}_i(\mathbf{A})\|^2 \leq \frac{8m\varepsilon^2}{\Delta_{\mathbf{K}}^2}, \tag{25}$$

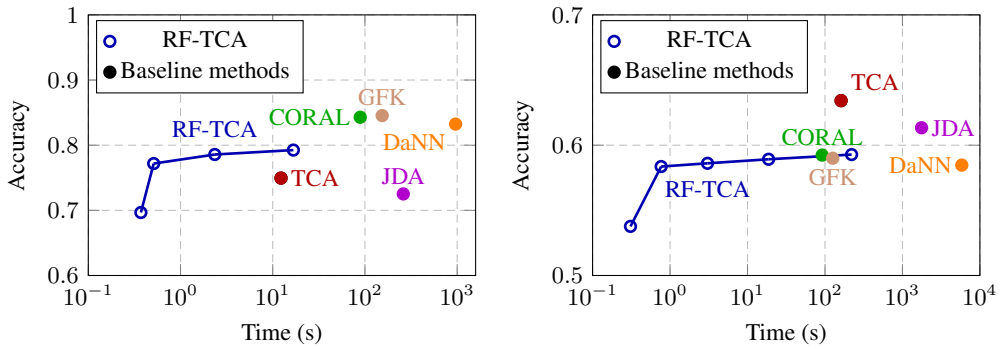
so it suffices to apply Corollary 1 to see that, there exists $C > 0$ so that for $N \geq C\frac{n\log(n)\sqrt{m}}{\Delta_{\mathbf{K}}\delta\varepsilon}$,

$$\|\mathbf{H}(\mathbf{\Sigma}^\mathsf{T}\mathbf{W}_{\text{RF}} - \mathbf{K}\mathbf{W})\| \leq \varepsilon \tag{26}$$

holds with probability at least $1-\delta$. This concludes the proof of Theorem 2. □

6 Numerical experiments

In this section, we compare the performance of the proposed random feature-based TCA (RF-TCA) approach with various baseline transfer learning methods including Transfer Component Analysis (TCA) [21], Joint Distribution Adaptation (JDA) [17], Geodesic Flow Kernel (GFK) [9], CoRelation Alignment (CORAL) [30], Domain Adaptive Neural Networks (DaNN) [7], on Office-Caltech [9] as well as Office-31 [25] datasets. To compare with different methods in both of time complexity and accuracy, Figure 3 depicts the results of RF-TCA by varying the number of random features N. We observe that using RF-TCA significantly decreases the time of transfer learning, with comparable or sometimes even better performance. We also observe that it (empirically) suffices to have a relatively small number of random features N to ensure satisfactory performance, as in line with our theory, and the continuous increase in N does *not* seem to significantly impact the performance.



(a) Office-Caltech (b) Office-31

Figure 3: Test accuracy and running time of RF-TCA versus baseline methods on Office-Caltech and Office-31 datasets. **Blue** circles represent the proposed RF-TCA approach using a different number of random features $N \in \{100, 500, 1\,000, 2\,000, 5\,000\}$, the **red**, **purple**, **green**, **brown** and **orange** dots represent **TCA**, **JDA**, **CORAL**, **GFK**, and **DaNN** approach, respectively.

The details of experiment are described as follows, and we refer the readers to Section C of the appendix for more experimental results and discussions.

- **Datasets:** The Office-Caltech [9] dataset has 4 subsets (Amazon, Caltech, Dslr and Webcam) with 10 classes in each. The Office-31 [25] dataset has 3 subsets (Amazon, Dslr and Webcam) with 31 classes in each. Experiments are performed on Decaf6 features [5] of both datasets. The data vectors are normalized to have unit Euclidean norms.
- **Experiment setting:** The obtained transferred features are classified using a fully-connected neural network with two hidden layers (having 100 neurons per layer). DaNN is the only end-to-end model, for which no (additional) classifier is applied. Each method is tested on each pair of subsets (so 12 tests for Office-Caltech and 6 tests for Office-31 data), and the final results are obtained by averaging over these tests.
- **Hyperparameters:** There are 5 hyperparameters in RF-TCA: the number of random features N, the dimension of latent space m, the regularization parameter γ, and the (Gaussian width) kernel parameter σ as in (6). We choose $m = 100$, search γ in the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$, the Gaussian (width) parameter σ in the set $\{5, 6, \cdots, 14, 15\}$. For each test, we perform hyperparameter search in the range above and report the best performance.

In this paper, we investigate the computational and generalization properties of TCA approach, and propose the computationally efficient random features-based TCA (RF-TCA) approach with a significant save in both space and time, and without severe performance degradation. Experiments on standard datasets are conducted to demonstrate the advantageous performance of the proposed method.

Acknowledgments and Disclosure of Funding

ZL would like to acknowledge the CCF-Hikvision Open Fund (20210008), the National Natural Science Foundation of China (NSFC-12141107), the Fundamental Research Funds for the Central Universities of China (2021XXJS110), the Key Research and Development Program of Hubei (2021BAA037), and the Key Research and Development Program of Guangxi (GuiKe-AB21196034) for providing partial support.

References

[1] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30, pages 185–209, 2013.

[2] Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma. Spectral Methods for Data Science: A Statistical Perspective. *Foundations and Trends® in Machine Learning*, 14(5):566–806, 2021.

[3] Yuxin Chen, Jianqing Fan, Cong Ma, and Kaizheng Wang. Spectral method and regularized MLE are both optimal for top-K ranking. *The Annals of Statistics*, 47(4):2204–2235, 2019.

[4] Romain Couillet and Zhenyu Liao. *Random Matrix Methods for Machine Learning*. Cambridge University Press.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[6] Rong Ge, Chi Jin, Sham, Praneeth Netrapalli, and Aaron Sidford. Efficient Algorithms for Large-scale Generalized Eigenvector Computation and Canonical Correlation Analysis. 48:2741–2750, 2016.

[7] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. In *Pacific Rim international conference on artificial intelligence*, pages 898–904. Springer, 2014.

[8] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, third edition, 2013.

[9] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016.

[11] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A Kernel Method for the Two-Sample-Problem. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

[12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012.

[13] Antony Joseph and Bin Yu. Impact of regularization on spectral clustering. *The Annals of Statistics*, 44(4):1765–1791, 2016.

[14] Shiva Prasad Kasiviswanathan and Mark Rudelson. Spectral Norm of Random Kernel Matrices with Applications to Privacy. *arXiv*, 2015.

[15] Zhenyu Liao, Romain Couillet, and Michael W. Mahoney. A random matrix analysis of random Fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent. In *Advances in Neural Information Processing Systems*, volume 33 of *NIPS'20*, pages 13939—13950. Curran Associates, Inc., 2020.

[16] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan A. K. Suykens. Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2021.

[17] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[19] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[20] Mark E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[21] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010.

[22] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[23] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[24] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20 of *NIPS'08*, pages 1177–1184. Curran Associates, Inc., 2008.

[25] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[26] Bernhard Schlkopf, Alexander Smola, and Klaus-Robert Mller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.

[27] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International conference on artificial neural networks*, pages 583–588. Springer, 1997.

[28] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2018.

[29] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. Algorithmic Learning Theory. *Lecture Notes in Computer Science*, pages 13–31, 2007.

[30] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[31] Joel A. Tropp. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[32] Andrea Vedaldi and Andrew Zisserman. Efficient Additive Kernels via Explicit Feature Maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012.

[33] Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material

Revisiting and Accelerating Transfer Component Analysis

A Useful lemmas

Lemma 2 (Sherman–Morrison). *For $\mathbf{A} \in \mathbb{R}^{p\times p}$ invertible and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p$, $\mathbf{A} + \mathbf{u}\mathbf{v}^\mathsf{T}$ is invertible if and only if $1 + \mathbf{v}^\mathsf{T}\mathbf{A}^{-1}\mathbf{u} \neq 0$ and*

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^\mathsf{T})^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\mathsf{T}\mathbf{A}^{-1}}{1 + \mathbf{v}^\mathsf{T}\mathbf{A}^{-1}\mathbf{u}}.$$

Theorem 3 (Weyl's inequality, [12, Theorem 4.3.1]). *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{p\times p}$ be symmetric matrices and let the respective eigenvalues of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{A} + \mathbf{B}$ be arranged in non-decreasing order, i.e., $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{p-1} \leq \lambda_p$. Then, for all $i \in \{1, \ldots, p\}$,*

$$\begin{aligned}
&\lambda_i(\mathbf{A} + \mathbf{B}) \leq \lambda_{i+j}(\mathbf{A}) + \lambda_{p-j}(\mathbf{B}), \quad j = 0, 1, \ldots, p - i,\\
&\lambda_{i-j+1}(\mathbf{A}) + \lambda_j(\mathbf{B}) \leq \lambda_i(\mathbf{A} + \mathbf{B}), \quad j = 1, \ldots, i.
\end{aligned}$$

In particular, taking $i = 1$ in the first equation and $i = p$ in the second equation, together with the fact $\lambda_j(\mathbf{B}) = -\lambda_{p+1-j}(-\mathbf{B})$ for $j = 1, \ldots, p$, implies

$$\max_{1\leq j\leq p} |\lambda_j(\mathbf{A}) - \lambda_j(\mathbf{B})| \leq \|\mathbf{A} - \mathbf{B}\|.$$

Theorem 4 (Davis–Kahan theorem, [33]). *Under the same notation as in Theorem 3, assume that for a given $i \in \{1, \ldots, n\}$ one has $\min\{|\lambda_{i-1}(\mathbf{A}) - \lambda_i(\mathbf{A})|, |\lambda_{i+1}(\mathbf{A}) - \lambda_i(\mathbf{A})| > 0$, then the corresponding eigenvectors satisfy*

$$\sin\Theta\left(\mathbf{u}_i(\mathbf{A} + \mathbf{B}), \mathbf{u}_i(\mathbf{A})\right) \leq \frac{2\|\mathbf{B}\|}{\min\{|\lambda_{i-1}(\mathbf{A}) - \lambda_i(\mathbf{A})|, |\lambda_{i+1}(\mathbf{A}) - \lambda_i(\mathbf{A})|\}}, \tag{27}$$

where we denote '$\sin\Theta$' the alignment between two vectors $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^n$ as

$$\Theta(\mathbf{u}_1, \mathbf{u}_2) \equiv \arccos\left(\frac{\mathbf{u}_1^\mathsf{T}\mathbf{u}_2}{\|\mathbf{u}_1\| \cdot \|\mathbf{u}_2\|}\right), \tag{28}$$

that satisfies $\|\mathbf{u}_1 - \mathbf{u}_2\| \leq \sqrt{2}\sin\Theta(\mathbf{u}_1, \mathbf{u}_2)$.

B Proofs

B.1 Proof of Corollary 1

In the sequel, we will use the shortcut notation $\hat{\mathbf{K}} \equiv \boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$. We aim to bound the following operator norm difference

$$\left\| \mathbf{K} - \frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}} - \left(\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma} - \frac{\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{y}\mathbf{y}^\mathsf{T}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma}}{\gamma + \mathbf{y}^\mathsf{T}\boldsymbol{\Sigma}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{y}} \right) \right\|. \tag{29}$$

First, we have, for $\|\mathbf{K} - \hat{\mathbf{K}}\| \leq \varepsilon$, by $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$, $\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$, and $\gamma > 0$ that

$$\begin{aligned}
&\left\| \frac{\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}}{\gamma + \mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}} - \frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}} \right\| \leq \frac{\left\| \gamma(\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}} - \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}) + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}} - \mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K} \right\|}{(\gamma + \mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})} \\
&\leq \frac{\left\| \gamma(\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}} - \mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}) \right\|}{(\gamma + \mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})} + \frac{\left\| \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}} - \mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K} \right\|}{(\gamma + \mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma + \mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}.
\end{aligned} \tag{30}$$

Then, for the first right-hand side term in (30), we have

$$\begin{aligned}\frac{\left\|\gamma(\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}-\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K})\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})} &= \frac{\gamma\left\|(\hat{\mathbf{K}}-\mathbf{K})\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}+\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}-\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}\\ &\leq \gamma\frac{\left\|\hat{\mathbf{K}}-\mathbf{K}\right\|\cdot\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}+\gamma\frac{\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}\cdot\left\|\hat{\mathbf{K}}-\mathbf{K}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}\\ &\leq \frac{\gamma\varepsilon}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}+\frac{\gamma\varepsilon}{\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}}\\ &\leq 2\varepsilon, \end{aligned} \tag{31}$$

where we consider $\|\mathbf{K}-\hat{\mathbf{K}}\|\leq\varepsilon$ and use the fact that $\frac{\mathbf{y}^\mathsf{T}\mathbf{Z}\mathbf{y}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{Z}\mathbf{y}}\leq 1$ for $\mathbf{Z}=\hat{\mathbf{K}}$ and $\mathbf{K}$ in the third inequality, as well as $0<c_{\mathbf{K}}\leq\lambda_{\min}(\mathbf{K})\leq\lambda_{\max}(\mathbf{K})\leq C_{\mathbf{K}}$ from Assumption 1 in the fourth inequality.

For the second right-hand side term in (30), we have, again for $\|\mathbf{K}-\hat{\mathbf{K}}\|\leq\varepsilon$ that

$$\begin{aligned}&\frac{\left\|\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}-\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})} = \frac{\left\|\mathbf{y}^\mathsf{T}(\mathbf{K}-\hat{\mathbf{K}})\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}-\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}\\ &\leq \frac{\left\|\mathbf{y}^\mathsf{T}(\mathbf{K}-\hat{\mathbf{K}})\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}+\frac{\left\|\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}-\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}\\ &\leq \frac{\|\hat{\mathbf{K}}-\mathbf{K}\|\cdot(\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})^2}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}+\frac{\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}\cdot\left\|\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}-\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}\right\|}{(\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y})(\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y})}\\ &\leq \varepsilon\left(\frac{2\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}+1\right)=2\varepsilon\cdot\frac{\mathbf{y}^\mathsf{T}(\hat{\mathbf{K}}-\mathbf{K})\mathbf{y}+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}+\varepsilon\leq 2\varepsilon\left(\frac{\mathbf{y}^\mathsf{T}(\hat{\mathbf{K}}-\mathbf{K})\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}+1\right)+\varepsilon\\ &\leq 2\varepsilon(\varepsilon/c_{\mathbf{K}}+2) \end{aligned} \tag{32}$$

where we use (31) and the fact that $\lambda_{\min}(\mathbf{K})\geq c_{\mathbf{K}}$.

Putting together, this gives

$$\begin{aligned}&\left\|\mathbf{K}-\frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}-\left(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}-\frac{\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}}\right)\right\|\\ &\leq\|\mathbf{K}-\hat{\mathbf{K}}\|+\left\|\frac{\hat{\mathbf{K}}\mathbf{y}\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}}{\gamma+\mathbf{y}^\mathsf{T}\hat{\mathbf{K}}\mathbf{y}}-\frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}\right\|\\ &\leq\varepsilon+2\varepsilon+\varepsilon(\varepsilon/c_{\mathbf{K}}+2)=\frac{\varepsilon^2}{c_{\mathbf{K}}}+4\varepsilon, \end{aligned} \tag{33}$$

so that in particular, for $\|\mathbf{K}-\hat{\mathbf{K}}\|\leq\varepsilon$ with $0<\varepsilon\leq\min(c_{\mathbf{K}},1)$, one gets

$$\left\|\mathbf{K}-\frac{\mathbf{K}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{K}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{K}\mathbf{y}}-\left(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}-\frac{\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}}{\gamma+\mathbf{y}^\mathsf{T}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{y}}\right)\right\|\leq 5\varepsilon, \tag{34}$$

which, together with Theorem 1, concludes the proof of Corollary 1.

C Additional numerical experiments

In this section, we provide additional experimental results. All experiments are performed on a machine with Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz and 3090Ti GPU. In Section C.1 we provide additional numerical results on the original TCA proposed in [21], by showing the difference between TCA and kernel PCA approaches, as well as the effect of regularization parameter γ (as discussed in, e.g., Remark 1). Section C.2 compares the running time of the original and the

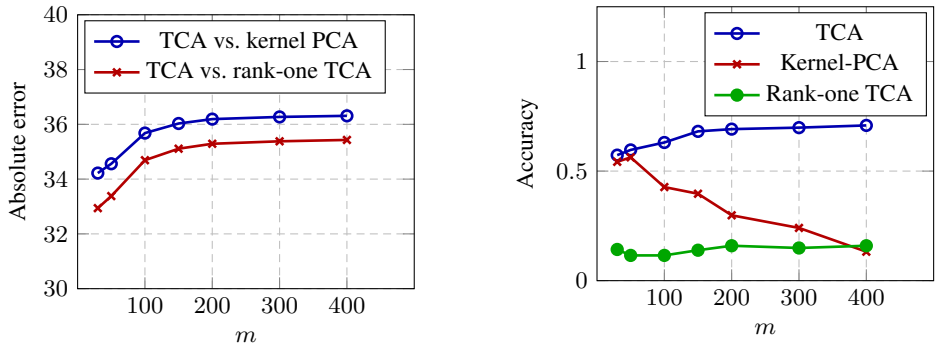
proposed random features-based TCA (RF-TCA) approaches. Section C.3 discusses the impact of pre-processing applied on the input data $\mathbf{X}$ and on the "transferred" feature $\mathbf{F} \equiv \mathbf{W}^\top \mathbf{K}$ in the case of TCA. Section C.4 illustrates the effects of different classifiers including fully-connected neural network (FCNN), support vector machine (SVM), and k-nearest neighbors (kNN) methods. Section C.5 contains the detailed experimental results to produce the average performance in Figure 3.

Below are the experimental settings in this section:

- **Classifiers:** We use 3 types of classifiers in this section: fully-connected neural network (FCNN), support vector machine (SVM) and k-nearest neighbor (kNN). FCNN is a fully-connected neural network with two hidden layers (having 100 neurons per layer). SVM uses the (Gaussian) RBF kernel and follows the same hyperparameter searching protocol as below. And the parameter k of kNN is 1.
- **Hyperparameters:** The choices of hyperparameter are given in the caption (of the corresponding figure or table), where we denote N the number of random features, m the dimension of latent space, γ the regularization parameter for TCA or RF-TCA, and σ the Gaussian (width) kernel parameter as in (6). We perform hyperparameter search in the range $\gamma \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3\}$ and $\sigma \in \{5, 6, \cdots, 14, 15\}$ and report the best performance.

C.1 Additional experiments and discussions on vanilla TCA

Following the discussion in Section 4 on the close connection between the TCA and kernel PCA (as numerically illustrated in Figure 2), we compare, in Figure 4, the absolute error and the classification accuracy of the transferred features obtained from (vanilla) TCA, kernel PCA (by using the top m eigenvectors of the centered kernel matrix $\mathbf{HKH}$), as well as the rank-one TCA approach using the rank-one matrix $\frac{\mathbf{HK}^2\mathbf{yy}^\top\mathbf{K}^2\mathbf{H}}{\gamma+\mathbf{y}^\top\mathbf{K}^2\mathbf{y}}$. Different from the case of Figure 2, Figure 4 shows that vanilla TCA *significantly* outperforms kernel PCA with a large m, while the performance of rank-one TCA remains the same for $m \geq 1$, as a consequence of the fact that the transferred feature is of rank one.



(a) Absolute error between features

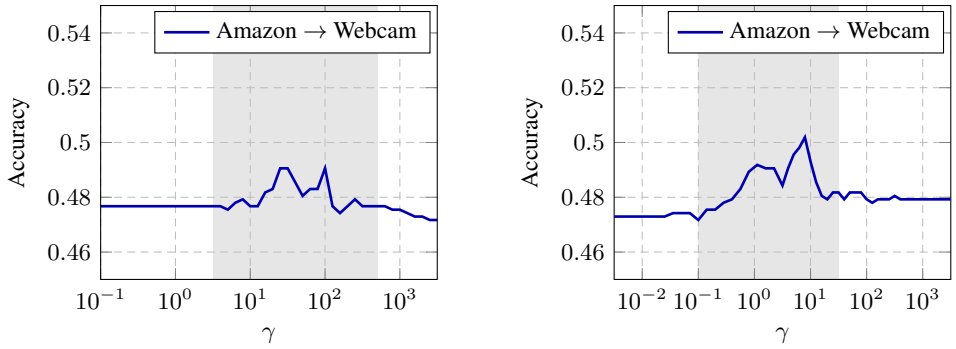
(b) Accuracy of different approaches

Figure 4: Absolute spectral norm error (**left**) and accuracy (**right**) between different transferred features, as a function of the latent space size m, on Decaf6 features of Amazon to Webcam of Office-Caltech [9] with a SVM classifier.

Figure 5 provides numerical evidence for Remark 1 on knowledge transfer from Amazon to Webcam of Office-31 data, with $n_S = 2817$ and $n_T = 795$. In accordance with the discussions in Remark 1, the classification accuracy varies as a function of the regularization parameter γ *only* when γ belongs to a specific interval (marked in gray in Figure 5). And the same holds for R-TCA. The largest eigenvalue of $\mathbf{K}$ is given by $C_{\mathbf{K}} = 81.70$ in this setting, which is consistent with the upper bound estimate in (12).

C.2 Running time of TCA versus RF-TCA

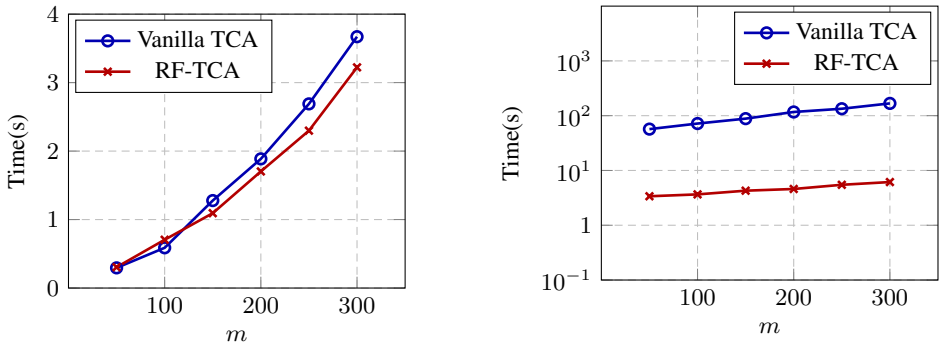
Using the power iteration approach to retrieve the top eigenvectors, the running time of vanilla TCA increases with the number of top eigenvectors m, while the runtime of RF-TCA increases with both



(a) Vanilla TCA (b) R-TCA

Figure 5: Classification accuracy versus the regularization parameter γ on Decaf6 features of Office-31 for $n_S = 2817$ (Amazon), $n_T = 795$ (Webcam) with SVM classifier. In this case, $\mathbf{y}^\top \mathbf{K}^2 \mathbf{y} = 10^{-0.793} = 0.1607$ and $\mathbf{y}^\top \mathbf{K} \mathbf{y} = 10^{-1.874} = 0.01336$.

m and N, the number of random features, as in Figure 3. Figure 6 depicts the running time with different m using vanilla TCA and RF-TCA: When the feature dimension m and the number of data n are both small, the running time of the two approaches are similar. However, the proposed RF-TCA takes much less time in large-scale problem for n large, as observed in Figure 6b.



(a) $n = 2\,081$ (b) $n = 10\,000$

Figure 6: Running time versus the dimension of latent space m, for $n = 2\,081$ (**left**) and $n = 10\,000$ (**right**), with vanilla TCA and RF-TCA ($N = 500$).

C.3 Impact of normalization

We observe in the numerical experiments that, perhaps surprisingly, the pre-processing of the input data plays a significant role in the performance of TCA. To have a more quantitative assessment of the impact of data pre-processing, we compare, in Table 1 the classification accuracy using raw $\mathbf{X} \in \mathbb{R}^{p \times n}$ and "normalized" data $\tilde{\mathbf{X}}$, as well as the transferred features $\mathbf{F} \equiv \mathbf{W}^\top \mathbf{K} \in \mathbb{R}^{m \times n}$ and normalized $\tilde{\mathbf{F}}$. Precisely, we consider the following normalization strategy on $\mathbf{X} \in \mathbb{R}^{p \times n}$ and/or $\mathbf{F} \in \mathbb{R}^{m \times n}$:

$$[\tilde{\mathbf{X}}]_{ij} = \frac{[\mathbf{X}]_{ij}}{\sqrt{\sum_{i=1}^{p} [\mathbf{X}]_{ij}^2}}, \quad [\tilde{\mathbf{F}}]_{ij} = \frac{[\mathbf{F}]_{ij}}{\sqrt{\sum_{i=1}^{m} [\mathbf{F}]_{ij}^2}}, \tag{35}$$

so that the normalized data or feature vectors are of *unit* (Euclidean) norm.

Table 1 shows the ablation study of how the above normalization affects the performance of vanilla TCA, where we observe, by pre-processing the input data vectors to have unit norm, TCA significantly performs better, and by a large margin, than on raw data. This observation is *consistent* over three different classifiers.

Table 1: Ablation study of normalization and its impact on classification accuracy.

Task		Pre-processing		Accuracy		
Source	Target	$\tilde{\mathbf{X}}$	$\tilde{\mathbf{F}}$	FCNN	SVM	KNN
Amazon	Caltech			0.1228	0.1228	0.1050
		✔		**0.8397**	0.7764	0.7373
			✔	0.1219	0.1121	0.0908
		✔	✔	0.8272	0.8219	0.7569
Amazon	Caltech			0.1524	0.1471	0.1137
		✔		**0.9363**	0.9008	0.8914
			✔	0.1440	0.0814	0.1012
		✔	✔	0.8945	0.8987	0.8935

C.4 Impact of different classifiers

In this section, we quantitatively evaluate, in Table 2, the classification performance of TCA on Decaf6 features of Office-Caltech data using three different classifiers: FCNN, SVM and kNN. As expected, TCA achieves optimal performance with FCNN, while other classifiers show somewhat comparable results.

Table 2: Different classifiers and their corresponding classification accuracy.

Classifier	A→C	C→A	A→D	D→A	Average
FCNN	**0.8272**	0.9112	**0.9112**	**0.9112**	**0.8902**
SVM	0.8219	**0.9217**	0.8598	0.7776	0.8452
kNN	0.7569	0.8977	0.8535	0.8924	0.8501

C.5 Numerical results

In this section, we display the experimental results to produce the averaged behaviors in Figure 3 in this paper. To illustrate the computational efficiency of the explicit formulation of TCA proposed in Lemma 1, we use TCA for the solution obtained by solving (2) via an ordinary or generalized eigenvalue problem, and vanilla TCA the solution obtained by using Lemma 2: the two solutions are numerically close, but have significantly different time complexity. These tables are more informative than Figure 3, and show, for example, that knowledge transferring between different domains is *directed* and leads to a significant gap in accuracy, e.g., between A→D and D→A in Table 5 for *all* methods, as in line with the observations made in precious efforts [18, 17]. Also, on more involved classification tasks (31-class classification in Office-31 versus ten-class classification in Office-Caltech), TCA seems to yield (relatively) better performance.

Table 3: Performance of different transfer learning approaches on Office-Caltech data with Decaf6 features.

Methods	TCA	RF1000	RF500	Vanilla TCA	JDA	CORAL	GFK	DaNN
A→C	0.8103	0.7649	0.7604	0.8076	0.8112	**0.8432**	0.8257	0.8392
A→D	0.8343	0.8343	0.8025	**0.8789**	0.8407	0.8471	0.8535	0.8300
A→W	0.7457	0.6610	0.6813	0.7355	0.6576	0.7389	**0.7593**	0.7567
C→A	0.8789	0.9102	0.9144	0.8778	0.8799	0.9237	**0.9242**	0.9030
C→D	0.6369	**0.9044**	0.8917	0.5859	0.5732	0.8853	0.8938	0.9000
C→W	0.6271	0.7932	0.7728	0.6101	0.5864	0.7932	**0.8350**	0.7967
D→A	0.8215	0.7453	0.6993	0.8131	0.7964	**0.8507**	0.7991	0.7960
D→C	0.5289	0.5975	0.5850	0.5075	0.4888	0.7684	**0.7857**	0.7392
D→W	0.8000	0.9084	0.9050	0.8000	0.8000	0.9796	**0.9910**	0.9667
W→A	0.7755	0.7286	0.6993	**0.7797**	0.7682	0.7766	0.7521	0.7400
W→C	0.6242	0.5805	0.5503	0.5983	0.6349	0.7079	**0.7262**	0.6850
W→D	0.9108	**1.0000**	**1.0000**	0.9171	0.8662	**1.0000**	**1.0000**	**1.0000**
Ave	0.7495	0.7857	0.7718	0.7426	0.7252	0.8429	**0.8454**	0.8324

Table 4: Running time of different transfer learning approaches on Office-Caltech data with Decaf6 features.

Methods	TCA	RF1000	RF500	Vanilla TCA	JDA	CORAL	GFK	DaNN
A→C	38.76	2.812	**0.6355**	3.072	604.7	92.91	117.6	983.6
A→D	5.231	2.349	**0.4795**	0.5285	180.9	90.27	191.3	973.8
A→W	7.981	2.438	**0.4993**	0.7088	203.3	87.09	191.5	957.5
C→A	38.63	2.892	**0.6337**	3.067	588.1	89.51	124.9	1004
C→D	10.92	2.569	**0.5157**	0.7563	249.0	90.99	117.9	955.6
C→W	10.75	2.538	**0.5471**	0.9705	251.8	88.10	119.8	947.4
D→A	5.356	2.373	**0.4776**	0.5294	183.9	89.27	185.1	924.4
D→C	10.99	2.475	**0.4985**	0.7536	249.1	90.44	116.3	947.4
D→W	0.2790	0.4151	0.4116	**0.07043**	81.59	87.74	187.6	908.7
W→A	7.774	2.459	**0.4993**	0.7024	210.4	86.76	184.7	959.9
W→C	10.66	2.527	**0.5304**	0.9671	245.5	87.18	123.4	965.5
W→D	0.2802	2.387	0.4131	**0.07328**	82.09	87.92	186.6	975.3
Ave	12.30	2.353	**0.5118**	1.016	260.9	89.01	153.9	961.8

Table 5: Performance of different transfer learning approaches on Office-31 data with Decaf6 features.

Methods	TCA	RF1000	RF500	Vanilla TCA	JDA	CORAL	GFK	DaNN
A→D	**0.6024**	0.5602	0.5562	0.5983	0.5742	0.5481	0.5582	0.4980
A→W	0.4855	0.5157	0.5106	0.4729	0.4553	0.4993	**0.5241**	0.5063
D→A	**0.4075**	0.3130	0.2942	0.3961	0.3943	0.3588	0.3463	0.3576
D→W	**0.9308**	0.8767	0.8792	0.9295	0.9106	0.8981	0.8704	0.8725
W→A	0.3890	0.3013	0.3258	**0.3933**	0.3706	0.3365	0.3395	0.3676
W→D	**0.9899**	0.9497	0.9357	0.9779	0.9759	0.9136	0.9016	0.9060
Ave	**0.6342**	0.5861	0.5836	0.6280	0.6134	0.5924	0.5900	0.5847

Table 6: Running time of different transfer learning approaches on Office-31 data with Decaf6 features.

Methods	TCA	RF1000	RF500	Vanilla TCA	JDA	CORAL	GFK	DaNN
A→D	206.2	3.257	**0.8764**	13.48	2207	96.64	125.9	6078
A→W	265.4	3.459	**0.9498**	17.28	2911	93.35	133.2	6066
D→A	208.0	3.415	**0.8695**	13.49	2211	93.06	124.4	5783
D→W	11.16	2.365	**0.4760**	0.8867	243.8	86.94	116.5	5545
W→A	266.9	3.428	**0.9389**	17.32	2779	93.38	127.3	5941
W→D	11.13	2.343	**0.4953**	0.8869	251.1	88.64	128.8	5620
Ave	161.5	3.045	**0.7676**	10.55	1767	92.00	126.0	5839