Domain and Modality Adaptation Using Multi-Kernel Matching

Tamir Baruch Yampolsky and Ronen Talmon
Viterbi Faculty of Electrical and Computer Engineering
Technion – Israel Institute of Technology
Haifa, Israel
{stamirya@campus, ronen@ee}.technion.ac.il

Ofir Lindenbaum
Faculty of Engineering
Bar-Ilan University
Ramat Gan, Israel
ofir.lindenbaum@biu.ac.il

***Abstract*—In this paper, we propose a new method for domain and modality adaptation using multi-kernel matching. Our method is based on the representation of the source and target sets with several local kernels centered at a small number of apriori known corresponding samples. We propose to match the local kernels and, in turn, aggregate the local matches and find a mapping between the source and target sets. We showcase the applicability of our method on simulations and real-world data sets that include EEG recordings for mental arithmetic identification and single-cell multi-omics. In these applications, we demonstrate the advantages of our method over recent competing schemes.**

***Index Terms*—Domain Adaptation, Modality Adaptation, Kernel Matching**

I. Introduction

In a broad range of machine learning applications, the training and test sets are assumed to reside in the same domain. However, often in practice, this assumption does not hold, for example, when the data are collected from different sensors, subjects, environments, etc. Such domain differences or shifts typically lead to significant performance degradation.

Many domain adaptation (DA) methods applied prior to or together with the learning procedure were developed as a remedy. Notable works include [1]–[4], to name but a few. Most DA methods assume that the train and test sets have different distributions but reside in the same feature space. This limits their applicability because they cannot be directly applied to data obtained from multiple modalities.

In this paper, in contrast to classical DA methods, we consider distinct domains residing in different diffeomorphic feature spaces, facilitating both domain and modality adaptation. Specifically, we assume some latent bijective map exists between these spaces and propose a domain and modality adaptation method based on multi-kernel matching (MKM). Our method receives as input two sets of points sampled from two domains in two possibly different spaces and a small reference set consisting of pairs of bijective points from the two domains. The proposed method has three stages. First, the two sets are divided into corresponding pairs of overlapping neighborhoods centered at the reference points. Then, the correspondence between the neighborhoods of each pair is found using kernel matching [5]. Finally, the multiple correspondences between the neighborhoods are integrated into a single function that maps one set to another.

We remark that the problem we consider is significantly different from the classical correspondence problem typically considered in shape analysis and graph matching [6] since we do not assume that for each point in one set, a corresponding point exists in the other set.

We test our method on simulations and real-world data sets that include Electroencephalography (EEG) recordings and single-cell multi-omics and demonstrate superior or on-par results compared with the state-of-the-art.

II. Related Work

Domain adaptation is a well-explored problem that has led to the development of many algorithms. Here, we mention several related geometric DA methods that serve as baselines for comparison to our method.

Scatter Component Analysis (SCA) [7] uses a simple geometric measure called *scatter* to construct a linear transformation that attenuates unimportant factors and enhances the distinction between classes. Multi-domain Discriminant Analysis (MDA) [8] learns a domain-invariant feature transformation that maximizes class separability using average class discrepancy. We remark that both methods do not support modality adaptation. DA methods based on Optimal Transport (OT) [9] map the source set to the target by solving the earth movers optimization problem. Manifold Alignment with Procrustes Analysis (MA-PA) [10] and Semi-Supervised Manifold Alignment (SSMA) [11] are kernel-based methods. MA-PA uses kernels to estimate the similarities between points in each set and then applies Procrustes analysis to the known corresponding pairs to retrieve a mapping plan. SSMA uses the Laplacian eigenmaps loss function with additional cost that accounts for mismatches of the known correspondences. Harmonic Alignment (HA) [12] and Diffusion Transport Alignment (DTA) [13] use a combination of kernel and geometric approaches to find the mapping plan. HA and DTA use kernel functions applied to the given sets and then attempt to minimize the geometric differences between the transformed sets. A significant difference between DTA and HA is that DTA uses a reference set, whereas HA is an unsupervised method.

III. Background

A. Kernel Matching

We briefly describe a method based on kernel matching originally presented for shape correspondence [5]. Here, we present it in the context of dataset alignment. Let k be a symmetric positive kernel function measuring the relation between two points in some set $\mathcal{F}$, i.e., $k : \mathcal{F} \times \mathcal{F} \to \mathbb{R}^+$. Assume $\mathcal{F}$ is a finite set consisting of N points. Then, the kernel of the set can be represented by a symmetric matrix $M_{\mathcal{F}} \in \mathbb{R}^{N\times N}$, whose (i,j)th element is given by

$$[M_{\mathcal{F}}]_{i,j} = k(x_i, x_j),$$

where x_i and x_j are the ith and jth samples in the set $\mathcal{F}$.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets consisting of N points, each with corresponding kernel matrices $M_{\mathcal{X}}$ and $M_{\mathcal{Y}}$, respectively. Accordingly, the sets alignment problem can be formulated as the following optimization problem

$$\begin{aligned}\Gamma_{\text{opt}} &= \text{argmin}_{\Gamma\in\mathcal{P}_N} \|\Gamma M_{\mathcal{X}} - M_{\mathcal{Y}}\Gamma\|^2 \\ &= \text{argmax}_{\Gamma\in\mathcal{P}_N} \langle \Gamma, M_{\mathcal{Y}}\Gamma M_{\mathcal{X}}\rangle,\end{aligned} \tag{1}$$

where $\mathcal{P}_N$ is the set of permutation matrices of size $N \times N$, and the inner product is $\langle A, B\rangle = \text{tr}(A^T B)$. This optimization problem is known as *kernel matching* (KM) (or *graph matching* (GM) in a more general context). The optimization in (1) is a quadratic assignment problem that is NP-hard. One can use a relaxation of the problem and solve

$$\Gamma' = \text{argmax}_{\Gamma\in\mathcal{B}_N} \langle \Gamma, M_{\mathcal{Y}}\Gamma M_{\mathcal{X}}\rangle, \tag{2}$$

where $\mathcal{B}_N$ is the set bi-stochastic matrices of size $N \times N$. The optimization problem in (2) can be solved by iteratively applying a linear programming algorithm such as the simplex method [14]. Then, given the bi-stochastic matrix Γ', we find a permutation according to

$$\Gamma_{\text{opt}} = \text{argmin}_{\Gamma\in\mathcal{P}_N} \|\Gamma - \Gamma'\|.$$

For more details on the kernel matching algorithm, see [5].

B. Discrete Optimal Transport for Domain Adaptation

OT aims to minimize the overall effort required to transport elements from a source set to a target set. Broadly, the discrete OT problem formulation for domain adaptation is as follows [9]. Consider two sets of samples from some space $\mathbb{S}$: $\mathcal{X}$ consisting of N_s samples and $\mathcal{Y}$ consisting of N_t samples. Let $\mathbf{f}_s \in \mathbb{R}^{N_s}$ be a vector representing a discrete uniform density on the samples in $\mathcal{X}$, such that $\mathbf{f}_s[i] = 1/N_s$ for $i = 1, \ldots, N_s$. Similarly, let $\mathbf{f}_t \in \mathbb{R}^{N_t}$ be a vector representing a discrete uniform density on $\mathcal{Y}$, such that $\mathbf{f}_t[j] = 1/N_t$ for $j = 1, \ldots, N_t$. In addition, let $C \in \mathbb{R}^{N_s\times N_t}$ be a cost matrix, whose (i,j)th element encodes the "work" one needs to invest in order to move sample $x_i \in \mathcal{X}$ to $y_j \in \mathcal{Y}$.

The discrete OT is the following optimization problem

$$\min_{\Gamma\in\mathcal{B}} \langle \Gamma, C\rangle, \tag{3}$$

where $\mathcal{B} = \{\Gamma \in \mathbb{R}^{N_s\times N_t} | \Gamma \mathbf{1}_{N_t} = \mathbf{f}_s, \Gamma^T \mathbf{1}_{N_s} = \mathbf{f}_t\}$ and $\mathbf{1}_{N_t} \in \mathbb{R}^{N_t}$ and $\mathbf{1}_{N_s} \in \mathbb{R}^{N_s}$ are all one vectors.

The OT problem in (3) can be solved using linear programming or more efficiently using the Sinkhorn OT [15]. The resulting Γ represents the mapping plan from $\mathcal{X}$ to $\mathcal{Y}$.

IV. Problem formulation

Consider two different diffeomorphic feature spaces, denoted by $\mathbb{S}_s$ and $\mathbb{S}_t$, representing the source and target spaces, respectively. Let $\gamma : \mathbb{S}_s \to \mathbb{S}_t$ denote a diffeomorphism, i.e., a smooth and bijective map between the spaces, whose inverse is smooth.

We assume we have access to three sets. The first set, referred to as the *source set*, is given by $\mathcal{X} = \{x_i\}_{i=1}^{N_s}$, consisting of N_s samples from the source space, i.e., $x_i \in \mathbb{S}_s$. The second set, referred to as the *target set*, is given by $\mathcal{Y} = \{y_i\}_{i=1}^{N_t}$, consisting of N_t samples from the target space, i.e., $y_i \in \mathbb{S}_t$. The third set, referred to as the *reference set*, is a set of n pairs $\mathcal{R} = \{(\bar{x}_i, \bar{y}_i)\}_{i=1}^{n}$, where $\bar{x}_i \in \mathbb{S}_s$, $\bar{y}_i \in \mathbb{S}_t$, $\bar{y}_i = \gamma(\bar{x}_i)$, and $n \ll \min(N_s, N_t)$.

Our goal is to find a map $\Gamma : \mathcal{X} \to \mathbb{S}_t$ that assigns for each sample in the source set $x_i \in \mathcal{X}$ its corresponding sample $\gamma(x_i)$ in the target space, with access only to the three sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{R}$, and without knowing γ. Using for example the L_2 loss, our goal could be formulated as

$$\min_{\Gamma} \sum_{i=1}^{N_s} \|\gamma(x_i) - \Gamma(x_i)\|_2^2.$$

However, the true map γ is unknown. To learn Γ, one could use the reference set and solve

$$\Gamma^* = \arg\min_{\Gamma} \sum_{i=1}^{n} \|\bar{y}_i - \Gamma(\bar{x}_i)\|_2^2$$

using some regression model for Γ. Yet, due to the small size of n, the available reference pairs $(\bar{x}_i, \bar{y}_i)$ are not enough to learn the mapping of the entire source set. Therefore, such direct minimization is not possible.

To evaluate the obtained map, we use two quantitative measures. The first measure is computed in cases, such as simulations and particular applications, where the hidden diffeomorphism γ is known, and one could directly compute

$$L_1 = \frac{\sum_{i=1}^{N_s} d_t(\Gamma(x_i), \gamma(x_i))^2}{\sum_{i=1}^{N_s} \|\gamma(x_i)\|^2} \tag{4}$$

in order to evaluate Γ, where d_t is the distance measure of $\mathbb{S}_t$. The second measure considers a downstream task. Suppose we have the labels of the target set $\mathcal{Y}$ denoted by $l_{\mathcal{Y}} = \{l_{y_i}\}_{i=1}^{N_t}$. To assess the obtained map Γ, we learn a labeling function $l_{\tilde{\mathcal{X}}} : \mathbb{S}_t \to \mathbb{R}$ from $\tilde{\mathcal{X}} = \Gamma(\mathcal{X})$ and compute

$$L_2 = \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(l_{\tilde{\mathcal{X}}}(y_i), l_{y_i})), \tag{5}$$

where $\ell : \mathbb{R}\times\mathbb{R}$ is some loss function measuring the agreement between labels.

Algorithm 1: MKM($\mathcal{X}, \mathcal{Y}, \mathcal{R}$)

Input: $\mathcal{X} = \{x_i\}_{i=1}^{N_s}, \mathcal{Y} = \{y_i\}_{i=1}^{N_t}, \mathcal{R} = \{(\bar{x}_i, \bar{y}_i)\}_{i=1}^{n}$.
Output: $\tilde{\mathcal{X}} = \{\tilde{x}_i\}_{i=1}^{N_s}$ the mapping of $\mathcal{X}$ to $\mathbb{S}_t$

1 **set** $\mathcal{X}_i = \{\bar{x}_i\},\ \mathcal{Y}_i = \{\bar{y}_i\}\ \forall\ i \leq n$
2 **while** $\bigcup_{i=1}^{n} \mathcal{X}_i \neq \mathcal{X}$ **do**
3 **for** $i \leftarrow 0$ **to** n **do**
4 $\mathcal{X}_i = \mathcal{X}_i \cup \min_{x \in \mathcal{X} \setminus \mathcal{X}_i} d_s(x_i, x)$
5 $\mathcal{Y}_i = \mathcal{Y}_i \cup \min_{y \in \mathcal{Y} \setminus \mathcal{Y}_i} d_t(y_i, y)$
6 **set** $S = |\hat{\mathcal{X}}_0|$
7 **Init** $\Gamma = 0_{\mathbb{R}^{N_s \times N_t}}$
8 **for** $i \leftarrow 0$ **to** n **do**
9 Calculate K_i, Q_i kernel matrices for $\mathcal{X}_i, \mathcal{Y}_i$
 $\Pi_i = \text{argmax}_{\Pi \in \mathcal{P}_S} \langle \Pi, Q_i \Pi K_i \rangle$
10 $\{\Pi\}_{i=1}^{n} \rightarrow \Gamma$ [see Sec. V-C]
11 **for** $m \leftarrow 0$ **to** N_s **do**
12 $\tilde{x}_m = \sum_{k=1}^{N_t} \Gamma(m,k) y_k$

V. The Proposed Algorithm

The proposed algorithm consists of three main steps. The first step divides the source and target sets into pairs of equal-size neighborhoods around the reference points. The second step finds a correspondence between each pair of neighborhoods. The third step incorporates all the correspondences into a single mapping plan of the source set to the target space.

The entire algorithm is presented in Algorithm 1, and each of its steps is described in more detail below.

A. Neighborhoods Construction

For each pair of reference points $(\bar{x}_i, \bar{y}_i),\ i = 1, \ldots, n$, we build two neighborhoods of equal size, denoted by $\mathcal{X}_i \subset \mathcal{X}$ and $\mathcal{Y}_i \subset \mathcal{Y}$. The construction procedure is iterative. We start by initializing the subsets with the reference points, i.e., $\mathcal{X}_i = \{\bar{x}_i\}$ and $\mathcal{Y}_i = \{\bar{y}_i\}$ for every $i = 1, \ldots, n$. Then, in an iterative manner, for each $i = 1, \ldots, n$, we add the nearest neighbor of $\bar{x}_i$ from $\mathcal{X}$ to $\mathcal{X}_i$ according to $\arg\min_{x \in \mathcal{X} \setminus \mathcal{X}_i} d_s(\bar{x}_i, x)$, where d_s is a distance in $\mathbb{S}_s$. Similarly, we add the nearest neighbor of $\bar{y}_i$ from $\mathcal{Y}$ to $\mathcal{Y}_i$ using d_y, a distance in $\mathbb{S}_t$. The procedure stops once we cover the entire source set, i.e., $\bigcup_{i=1}^{n} \mathcal{X}_i = \mathcal{X}$.

B. Neighborhood Matching

Since $\mathcal{X}_i$ and $\mathcal{Y}_i$ represent small neighborhoods around corresponding points between $\mathbb{S}_s$ and $\mathbb{S}_t$, we expect that the mapping of the points in $\mathcal{X}_i$ to the target space $\mathbb{S}_t$ will be close to the points in $\mathcal{Y}_i$. Hence, we compute a point correspondence between $\mathcal{X}_i$ and $\mathcal{Y}_i$ using kernel matching as follows.

For each pair of neighborhoods $i = 1, \ldots, n$, we build two kernels, K_i and Q_i, consisting of pairwise affinities. The two kernels are symmetric matrices of size $S \times S$, where $S = |\mathcal{X}_i|$ is the cardinality of the neighborhoods, and their (j,l)th elements are given by

$$K_i(j,l) = \exp\left(-d_s^2\left(x_{i_j}, x_{i_l}\right)/(2\epsilon_{\mathcal{X}_i}^2)\right),$$
$$Q_i(j,l) = \exp\left(-d_t^2\left(y_{i'_j}, y_{i'_l}\right)/(2\epsilon_{\mathcal{Y}_i}^2)\right),$$

where $\epsilon_{\mathcal{F}}$ is a hyper-parameter of the kernel calculated as the median of all distances in the subset $\mathcal{F}$, i_j is the index of the jth point in $\mathcal{X}_i$, and i'_j is the index of the jth point in $\mathcal{Y}_i$.

For each pair of corresponding kernel matrices (K_i, Q_i), we solve (1), using the relaxation proposed in Sec. III-A, to get a mapping plan $\Pi_i \in \mathbb{R}^{S \times S}$ between $\mathcal{X}_i$ and $\mathcal{Y}_i$:

$$\Pi_i = \text{argmax}_{\Pi \in \mathcal{P}_S} \langle \Pi, Q_i \Pi K_i \rangle.$$

Then, we extended the mapping plan Π_i to a full plan $\Gamma_i \in \mathbb{R}^{N_s \times N_t}$ by setting $\Gamma_i(i_j, i'_l) = \Pi_i(j,l)$ for $j, l = 1, \ldots, S$ and 0 otherwise. Overall, this step results in n mapping plans, $\{\Gamma_i \in \mathbb{R}^{N_s \times N_t} | i = 1, \ldots, n\}$.

C. Neighborhood Aggregation

In this step, we build a single mapping plan $\Gamma \in \mathbb{R}^{N_s \times N_t}$ based on the n neighborhood mapping plans Γ_i as follows. Consider a source sample $x_m \in \mathcal{X}$. If x_m resides only in one subset $\mathcal{X}_i$, we map x_m according to Γ_i. If x_m resides in more than one subset, we map x_m to the centroid of the respective mappings. Formally, the aggregated plan $\Gamma \in \mathbb{R}^{N_s \times N_t}$ is given by $\boldsymbol{\Upsilon}(m,k) = \sum_{i=1,\ldots,n} \Gamma_i(m,k)$, with $\Gamma(m,k) = \frac{1}{\sum_{k'=1,\ldots,N_t} \boldsymbol{\Upsilon}(m,k')} \boldsymbol{\Upsilon}(m,k)$, for $m = 1, \ldots, N_s$ and $k = 1, \ldots, N_t$.

Using Γ, we map each point x_m in the source set to the target space as follows $\tilde{x}_m = \sum_{k=1}^{N_t} \Gamma(m,k) y_k$. We denote the mapped set as $\tilde{\mathcal{X}} = \{\tilde{x}_i\}_{m=1}^{N_s}$. Since our algorithm involves matching multiple pairs of kernels, we term it Multi-Kernel Matching (MKM).

Regarding computational complexity, we remark that graph (kernel) matching is an NP-hard problem generally unsolvable in polynomial time. There exist algorithms that relax the problem, leading to the computational complexity of $\mathcal{O}(N_s^{\alpha})$, where $\alpha \geq 4$ [16]. The computational complexity of MKM mainly depends on the complexity of the neighborhood matching (Step 10 in Algorithm 1), which relies on graph matching. There, we solve n matching problems. Assuming that the size of the subsets is $S = \frac{N_s}{n}$, the computational complexity of this step is of $\mathcal{O}(\frac{N_s^{\alpha}}{n^{\alpha-1}})$.

VI. Experimental results

We test MKM on a simulation and two datasets of EEG measurements and multi-omics data.

A. Toy Problem

This toy problem has two purposes: the first is visualization, and the second is to show the advantage of MKM over OT-based methods. OT is a commonly used method for DA. But as shown in [17], it does not accommodate volume-preserving discrepancies such as rotations.

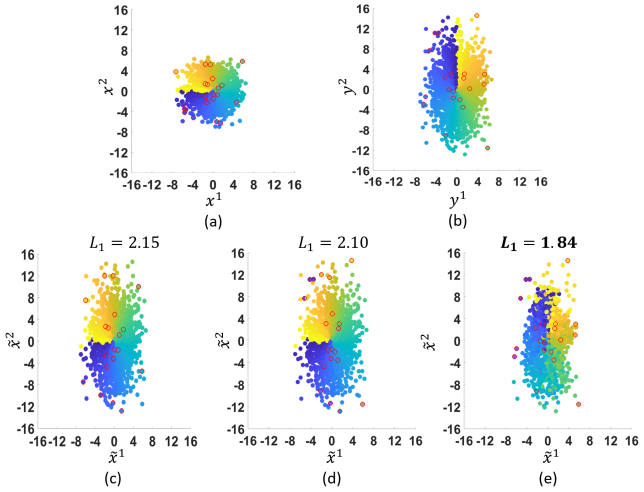

Fig. 1. Adaptation results on the toy problem. (a) The source set. (b) The target set. (c)-(e) The adaptation results of the source set to the target domain of OT, MOT, and MKM, respectively, with L_1 evaluation scheme (4) result. The reference pairs are marked in red.

Therefore, we compare our method to OT as presented in Sec. III-B with $c(x_i, y_j) := \|x_i - y_j\|_2$. Because OT does not use the reference set, we also compare MKM to the following modified OT for a fair comparison. We add the elements from the reference set to the source and target sets $\hat{\mathcal{X}} = \mathcal{X} \cup \{\bar{x}_i\}_{i=1}^n, \hat{\mathcal{Y}} = \mathcal{Y} \cup \{\bar{y}_i\}_{i=1}^n$. Then, we solve the OT optimization problem (3) between $\hat{\mathcal{X}} = \{\hat{x}\}_{i=1}^{N_s+n}$ and $\hat{\mathcal{Y}} = \{\hat{y}\}_{i=1}^{N_t+n}$ with the following transportation cost matrix between the two sets

$$C(i,j) := \begin{cases} 0 & (\hat{x}_i, \hat{y}_j) \in \mathcal{R} \\ \|\hat{x}_i - \hat{y}_j\|_2 & \text{otherwise} \end{cases}. \tag{6}$$

We call this method Modified OT (MOT).

In this toy problem, we consider the spaces $\mathbb{S}_s = \mathbb{S}_t = \mathbb{R}^2$. The diffeomorphism between the spaces $\gamma : \mathbb{S}_s \to \mathbb{S}_t$ is

$$\begin{pmatrix} y^1 \\ y^2 \end{pmatrix} = \gamma \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} \cos(\frac{\pi}{2}) & -\sin(\frac{\pi}{2}) \\ \sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix},$$

i.e., γ applies rotation and scaling of the vertical axis. The source set $\mathcal{X}$ consists of $N_s = 1800$ points sampled from $\mathbb{P}_s \sim \mathcal{N}(\mathbf{0}, 5I_2)$, where $\mathbf{0}$ is an all-zero vector and I_2 is the 2×2 identity matrix. The target set $\mathcal{Y}$ consists of $N_t = 2000$ points obtained by sampling N_t points from $\mathbb{P}_s$ and applying γ. The reference set consists of $n = 20$ corresponding pairs.

We applied MKM to $(\mathcal{X}, \mathcal{Y}, \mathcal{R})$ and compared it to OT and MOT. Fig. 1(a) shows the source set, where every sample is colored by its angle with respect to the origin. Additionally, the reference samples are marked by red circles. Fig. 1(b) shows the target set. Every sample is colored by the corresponding angle in the source space. Figs. 1(c-e) show the result of applying OT, MOT, and MKM, respectively, where each sample maintains its color.

In Fig. 1(c), we see that OT stretched the source set vertically but did not retrieve the rotation applied by γ. This behavior could be expected due to the limitations of OT [17]. In Fig. 1(d), we can see that MOT did not show a significant advantage over OT. MOT mapped known pairs appropriately, but, the remaining samples were mapped similarly to the standard OT without accommodating the rotation. This is because the cost function of MOT gives a zero cost for mapping known pairs, but there is no constraint for neighboring samples. In contrast to OT and MOT, we see in Fig. 1(e) that MKM recovers the map γ and appropriately rotates the source set.

To compare the three algorithms, we use the measure L_1 (4) equipped with the Euclidean distance. The results appear above each plot in Fig. 1 and further support the visual trends.

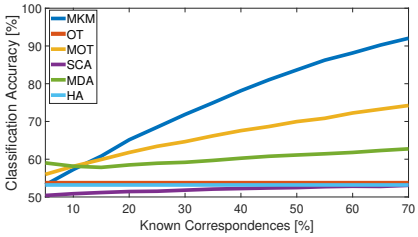

Fig. 2. Classification accuracy in mental arithmetic identification.

B. Mental Arithmetic Identification from EEG

We apply the proposed algorithm to a brain-computer interface (BCI) task. Specifically, we consider the task of inter-subject mental state identification based on EEG recordings. We use a publically available data set [18] of EEG recordings with 30 channels from 29 subjects. A total of $N = 60$ trials were recorded from each subject. At each trial, the subject was in one of two mental states: while solving an arithmetic assignment or resting. We postulate that the recordings of each subject live in a different domain and use MKM for adapting these domains to facilitate inter-subject identification. We remark that this dataset inherently does not include known correspondences between the domains that can be used as a reference set for MKM. Instead, we assume we have n EEG recordings with known mental state labels from each subject and designate these trials with the same label as corresponding.

We follow previous work [17], [19], [20] that showed that covariance matrices are useful features for various BCI tasks. Specifically, covariance matrices are Symmetric and Positive Definite (SPD) matrices. In [21], it was shown that the Riemannian geometry of SPD matrices based on the affine-invariant metric [22] provides a useful measure of matrix similarity. Accordingly, we compute the covariance matrix of the EEG recordings from each trial, resulting in N covariance matrices per subject, and use the following distance $d_{\text{SPD}}(T_1, T_2) = ||\log(T_2^{-\frac{1}{2}} T_1 T_2^{-\frac{1}{2}})||_F$, s.t. $T_1, T_2 \in \mathbb{R}^{30\times30}$, which is induced from the affine-invariant metric. In the remainder of this section, we use d_{SPD} whenever a distance calculation is required. It is worthwhile noting the capability of MKM to support various feature and data metric spaces.

Let $\mathcal{X}^j = \{x_i^j\}_{i=1}^N$ denote the set of covariance matrices of subject j, where i is the trial index. In addition, let $\mathcal{R}^{j,l} = \{(\bar{x}_i^j, \bar{x}_i^l)\}_{i=1}^n$ denote the reference set between $\mathcal{X}^j$ and $\mathcal{X}^l$. We apply MKM to every pair of sets $\mathcal{X}^j$ and $\mathcal{X}^l$ and their associate reference set $\mathcal{R}^{j,l}$ for every $1 \le j, l \le 29, l \ne m$ and obtain the adapted set $\widetilde{\mathcal{X}}^j$.

TABLE I
CLASSIFICATION ACCURACY IN SINGLE-CELL MODALITY MATCHING.

Downstream Task	Method				
	DTA	MA-PA	MAGAN	SSMA	MKM
1-NN	0.726	0.666	0.673	0.721	**0.7729**
10-NN	0.708	0.581	0.675	0.659	**0.733**

We compare MKM to several other DA algorithms: OT, MOT, SCA, MDA, and HA. To evaluate the adaptation, we train a linear SVM classifier on $\widetilde{\mathcal{X}}^j$, test it on $\mathcal{X}^l$, and compute L_2 (5) to assess the classification result.

We repeated this experiment for different numbers of corresponding pairs n. In Fig. 2, we show the average results over all possible pairs of subjects as a function of the relative number of known correspondences. We see that the performance of MKM increases as the number of known correspondences grows. For more than 12% known correspondences, MKM outperforms all the other algorithms by a large margin. However, we see that for a small number of known correspondences ($n \leq 7$), OT, MOT, and MDA outperform MKM because a small number of reference pairs could make MKM very sensitive to errors in one of the kernel matching problems.

This experiment highlights two fundamental advantages of MKM. First, MKM is easily adapted to different feature spaces, e.g., the space of SPD matrices. Second, MKM performs well even when accurate sample-wise correspondences between the domains are unavailable.

C. Single-Cell Modality Matching

We showcase an application of MKM to modality adaptation. For this purpose, we consider publicly available data from a recent competition [23]. The dataset contains two sets from different modalities with a known sample correspondence. The first set consists of measurements of gene expressions (RNA) of multiple cells. The second set consists of protein abundance (ADT) from the same cells.

We repeated the experiment in [13] with MKM as the modality matching algorithm. Our evaluation is equivalent to calculating L_2 with 1-NN or 10-NN as the labeling function. Table I shows the performance of MKM compared to several competing algorithms. We see that MKM outperforms all the other algorithms.

We tested the sensitivity of MKM to false sample correspondences by applying it with a reference set of size $N_s/10$, out of which some were not from the same cell but only of the same cell type. We repeated the previous experiment with different numbers of true (and false) correspondences. The results are in Fig. 3. The results indicate the robustness of MKM to the false correspondences. Specifically, even in the presence of false correspondences, MKM still outperforms the other algorithms in Table I.

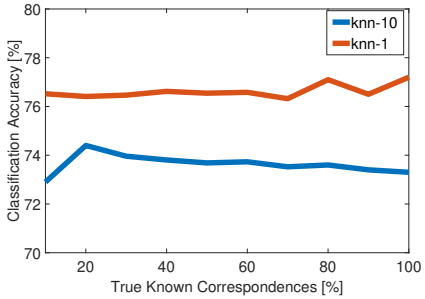

Fig. 3. Classification accuracy in single-cell data with false correspondences.

ACKNOWLEDGEMENT

TBY and RT were supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 802735-ERC-DIFFOP. RT also acknowledges the support of the Schmidt Career Advancement Chair.

REFERENCES

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *NIPS*, 2006.

[2] Daumé III, Hal, "Frustratingly easy domain adaptation," *preprint arXiv:0907.1815*, 2009.

[3] S. Ben-David, J. Blitzer, and K. Crammer, et al., "A theory of learning from different domains", *Machine learning*, vol. 79, pp. 151–175, 2010.

[4] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," *AAAI*, 2016.

[5] M. Vestner, Z. Lahner, and A. Boyarski, et al., "Efficient deformable shape correspondence via kernel matching," *3DV*, pages 517-526, 2017.

[6] O. V. Kaick, H. Zhang, and G. Hamarneh, et al., "A survey on shape correspondence," *Comput. graphics forum*, Vol. 30, No. 6, pp. 1681–1707, 2011.

[7] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Jul. 2017.

[8] S. Hu, K. Zhang, Z. Chen, and L. Chan, "Domain generalization via multidomain discriminant analysis," *UAI*, PMLR, 2020.

[9] N. Courty, R. Flamary, and D. Tuia , et al., "Optimal transport for Domain Adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[10] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," *In Proceedings of the 25th ICML*, pages 1120–1127, 2008.

[11] J. Ham, D. Lee, and L. Saul, "Semisupervised alignment of manifolds", *AIStat*, pages 120–127. PMLR, 2005.

[12] J. S. Stanley III, S. Gigante, G. Wolf, and S. Krishnaswamy, "Harmonic alignment," *SDM*, pages 316–324., 2020.

[13] A. F. Duque, G. Wolf, and K. R. Moon, K. R. Moon, "Diffusion Transport Alignment," *arXiv preprint arXiv:2206.07305*, 2022.

[14] R. H. Bartels and G. H. Golub, "The simplex method of linear programming using LU decomposition," *Commun. ACM*, 12, 5, pp. 266–268, 1969.

[15] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, 26, 2013.

[16] R. E. Burkard, "Quadratic assignment problems," *Cent. Eur. Oper. Res.*, Volume 15, Issue 3, Pages 283-289, 1984.

[17] O. Yair, F. Dietrich, R. Talmon, and I. G. Kevrekidis, "Domain Adaptation with Optimal Transport on the Manifold of SPD matrices," *arXiv e-prints*, 2019.

[18] J. Shin, A. V. Lühmann, and B. Blankertz, et al., "Open access dataset for EEG+ NIRS single-trial classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1735– 1745, Oct. 2017.

[19] O. Yair, M. Ben-Chen and R. Talmon, "Parallel transport on the cone manifold of SPD matrices for Domain Adaptation," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1797-1811, 2019.

[20] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, 2013.

[21] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain-computer interface classification by Riemannian geometry," *IEEE Trans. Biomed. Eng.*, Vol. 59, No. 4, pp. 920–928, 2011.

[22] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vision*, Vol. 66, pp. 41–66, 2006.

[23] M. D. Luecken, D. B. Burkhardt, and R. Cannoodt, et al., et al., "A sandbox for prediction and integration of dna, rna, and proteins in single cells," *35th conference on NeurIPS datasets and benchmarks track*, 2021.