Manifold Alignment with Label Information

Andres F. Duque Correa
Department of Mathematics and Statistics
Utah State University
Logan, Utah, USA
andres.duque@usu.edu

Myriam Lizotte
Department of Mathematics and Statistics
University of Montreal
Montreal, QC, Canada
myriam.lizotte@mila.quebec

Guy Wolf
Department of Mathematics and Statistics
University of Montreal
Montreal, QC, Canada
wolfguy@mila.quebec

Kevin R. Moon
Department of Mathematics and Statistics
Utah State University
Logan, Utah, USA
kevin.moon@usu.edu

***Abstract*—Multi-domain data is becoming increasingly common and presents both challenges and opportunities in the data science community. The integration of distinct data-views can be used for exploratory data analysis, and benefit downstream analysis including machine learning related tasks. With this in mind, we present a novel manifold alignment method called MALI (Manifold alignment with label information) that learns a correspondence between two distinct domains. MALI belongs to a middle ground between the more commonly addressed *semi-supervised* manifold alignment, where some known correspondences between the two domains are assumed to be known beforehand, and the purely *unsupervised* case, where no information linking both domains is available. To do this, MALI learns the manifold structure in both domains via a diffusion process and then leverages discrete class labels to guide the alignment. MALI recovers a pairing and a common representation that reveals related samples in both domains. We show that MALI outperforms the current state-of-the-art manifold alignment methods across multiple datasets.**

***Index Terms*—Manifold alignment, manifold learning, semi-supervised learning**

I. Introduction

The data collection process of a given phenomena may be affected by different sources of variability, creating seemingly distinct domains. For instance, natural images with different illumination, contrast or noise, may affect the classification performance of a machine learning model previously trained on a different domain. In biology, the modern study of single-cell dynamics is conducted via different instruments, conditions and modalities, raising different challenges and opportunities [1], [2]. In many cases, the relationships between the different domains are unknown. Hence, the fusion and integration of multi-domain data has been extensively studied in the data science community for supervised learning as well as data mining and exploratory data analysis. One of the earliest methods to do this is Canonical Correlation Analysis (CCA), which finds a linear projection that maximizes the correlation between the two domains [3].

This research was supported in part by the NSF under Grant 2212325 [K.M].

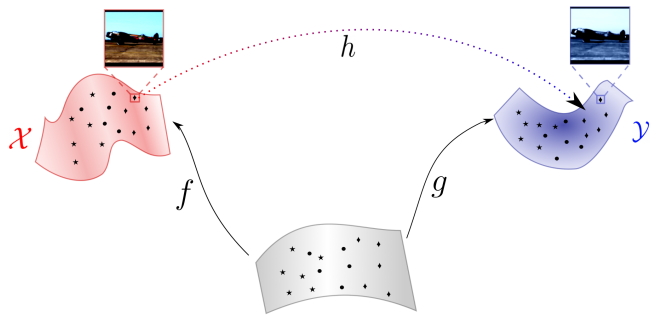

Fig. 1. **Manifold alignment**. Two different datasets of the same underlying phenomena are captured in different conditions, instruments, experimental designs, etc. Manifold alignment assumes a common latent space (grey) from which the observations are mapped by functions f and g to the different ambient spaces. We seek to find the underlying relationship h between observations living in different spaces $\mathcal{X}$ and $\mathcal{Y}$ without assuming any pairing known *a priori*. Instead we assume there are labeled observations for different classes (different shapes).

In many applications, a reasonable assumption to make is that the data collected in different domains is controlled by a set of shared underlying modes of variation or latent variables. The manifold assumption is also often applicable in this case, in which the data measured in the different domains lie on low-dimensional manifolds embedded in the high-dimensional ambient spaces, being the result of smooth mappings of the latent variables (see Fig. 1). With this in mind, manifold alignment (MA) has become a common technique for data integration. Some applications of MA include handling different face poses and protein structure alignment ([4], [5]), medical images for Alzheimer's disease classification ([6], [7]), multi-modal sensing images [8], graph-matching [9], and integrating single-cell multi-omics data [10] .

Multiple MA methods have been proposed under different prior knowledge assumptions that relate the two domains. Methods such as CCA or multi-view diffusion maps [11] can be categorized as *supervised* MA, since the data is assumed

to come in a paired fashion. More challenging scenarios arise when partial or null *a priori* pairing knowledge is considered. The purely *unsupervised* algorithms are designed for scenarios where neither pairings between domains nor any other side-information is available. As a consequence they rely solely on the particular topology of each domain to infer inter-domain similarities (e.g. [10], [12]–[14]).

The group of methods that leverage some sort of additional information are often categorized as *semi-supervised* MA. As a special widely studied case, several methods consider partial correspondence information, where a few one-to-one matching samples work as anchor points to find a consistent alignment for the rest of the data. Some papers leverage the graph structure of the data [15]–[18] and are closely related to Laplacian eigenmaps [19]. Others resort to neural networks such as the GAN-based MAGAN [20] or the autoencoder presented in [21].

However, even partial correspondences can be expensive or impossible to acquire. This is the case in many biological applications where the measurement process destroys the cells, making it impossible to measure other qualities of the exact same cells. But, even if there are no known correspondences between domains, we do not have to resort to unsupervised MA if we have access to side information about the datasets from both domains. A common source of side information, is a set of discrete class labels [22]–[24]. For instance, cell measurements can be labeled by experts by cell type in the different domains. In this paper, we propose a new semi-supervised MA algorithm called MALI (Manifold Alignment with Label Information) that leverages the manifold structure of the data in both domains combined with the discrete label information to align the datasets. MALI is built upon the widely-used manifold learning method Diffusion Maps [25] and does not require any known corresponding points in the different domains.

II. Problem Description

Assume we have two datasets $\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \in \mathrm{R}^{n\times p}$ and $\mathcal{Y} = \{y_1, y_2, \ldots, y_m\} \in \mathrm{R}^{m\times q}$. We assume that all of the points in $\mathcal{X}$ are labeled with discrete (i.e. class) labels $\mathcal{L}_x = \{\ell_1^x, \ldots, \ell_n^x\}$ while the points in $\mathcal{Y}$, called the target domain, may be partially or fully labeled with discrete labels $\mathcal{L}_y = \{\ell_1^y, \ldots, \ell_r^y\}$, with $r \leq m$. The problem consists of learning an alignment between both data manifolds by leveraging their respective geometric structures as well as the label knowledge available from both domains. There are several possible ways to represent such an alignment using MA algorithms. One way is to directly learn correspondences between points in $\mathcal{X}$ and $\mathcal{Y}$. A second way to represent the alignment is to learn a shared embedding space. A third way is to learn cross-domain similarities between points. This information can then be leveraged to either learn direct correspondences between the domains or to learn a shared embedding space. The data may also be visualized using the learned cross-domain similarities for exploratory data analysis. We focus primarily on learning the cross-domain similarities as they can be used to obtain either a shared embedding space or direct correspondences.

III. Manifold alignment with label information (MALI)

The basic idea behind MALI resides in finding an inter-domain distance between x_i and y_j that is leveraged to recover cross-domain relations. We start by building a similarity graph from the data on each domain, where the weights of the edges connecting the nodes are computed via an α-decay kernel [26]:

$$W_{\mathcal{X}}(x_i, x_j) = \frac{1}{2}\exp\left(-\frac{||x_i - x_j||^\alpha}{\sigma_k^\alpha(x_i)}\right) + \frac{1}{2}\exp\left(-\frac{||x_i - x_j||^\alpha}{\sigma_k^\alpha(x_j)}\right), \quad (1)$$

where $\alpha > 0$ and $\sigma_k(x_i)$ is the k-nearest neighbor distance of x_i in $\mathcal{X}$. The kernel $W_{\mathcal{Y}}$ is computed in a similar manner. The hyperparameters α and k control the connectivity and local geometry preservation in the graph, although most methods are typically robust to these hyperparameters when using this kernel [26]. For the particular experiments presented in this work we set $\alpha = 10$ and $k = 10$.

By row-normalizing W_* ($* \in \{\mathcal{X}, \mathcal{Y}\}$) using their corresponding degree matrices D_*, we obtain the respective diffusion operators $P_{\mathcal{X}} = D_{\mathcal{X}}^{-1} W_{\mathcal{X}} \in \mathrm{R}^{n\times n}$ and $P_{\mathcal{Y}} = D_{\mathcal{Y}}^{-1} W_{\mathcal{Y}} \in \mathrm{R}^{m\times m}$. The diffusion operator thus represents a probability transition matrix for the graph represented with the kernel matrix W_*. Typically, the diffusion operator is exponentiated P_*^t to describe the transition probabilities among observations after t steps in a random walk [25], [26]. Instead, we build a time-independent similarity matrix by aggregating the transition probabilities from every possible t-step random walk as follows:

$$M_* = \sum_{t=1}^{\infty}(P_* - \mathbb{1}\phi_0^T)^t = (I - (P_* - \mathbb{1}\phi_0^T))^{-1} - I, \quad (2)$$

where $\mathbb{1}$ is a vector of ones and ϕ_0 is the stationary distribution of the Markov chain, represented by the first left eigenvector of P_*. Subtracting $\mathbb{1}\phi_0^T$ from P_* enables the series to converge.

This construction was previously developed in [27], and constitutes the key quantity for the calculation of diffusion pseudotime (DPT). The advantages of working with M instead of P^t are twofold. First, we do not need to select the hyper-parameter t, which may be dataset dependent. Given the nature of the problem, a cross-validation-based approach for hyper-parameter tuning is not possible. Second, the matrix M builds a more extensive connection across the data as it considers the relationships between points at different scales of random walks. This does have the effect of smoothing the local relationships and the fine granular similarities are lost. However, this is not a problem as MALI focuses more on coarse similarities as we describe next.

To find inter-domain similarities, we need a new feature representation shared by the two domains. For this purpose, we use the label information, since it is the only available cross-domain information we possess a priori. Therefore, we

aggregate the similarities of each observation grouped by each of the labels as follows:

$$M_*^l(i,c) = \frac{1}{p_c^*} \sum_{j \in \mathcal{I}_c^*} M_*(i,j), \tag{3}$$

where $\mathcal{I}_c^*$ is the set of indices in domain $*$ labeled with class $c \in \mathcal{C}$, with $\mathcal{C}$ denoting the set of all labels present in both domains, and p_c^* is the estimated prior class probability (e.g. $p_c^{\mathcal{X}} = \frac{1}{n}\sum_{i=1}^n \mathbf{1}\{\ell_i^x = c\}$ where $\mathbf{1}\{\cdot\}$ is the indicator function). Normalizing by the priors p_c^* accounts for class unbalance. The matrix $M_*^l \in \mathrm{R}^{\cdot \times |\mathcal{C}|}$ encodes a coarse similarity between samples and labels. Notice that we now have a common similarity between $\mathcal{X}$ and $\mathcal{Y}$ and the labels, even though the datasets may have come from different spaces with different dimensionality. This allows us to compute inter-domain cosine distances between points $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$:

$$D_{ij} = \left(1 - \frac{\langle M_{\mathcal{X}}^l(i,:), M_{\mathcal{Y}}^l(j,:)\rangle}{||M_{\mathcal{X}}^l(i,:)||||M_{\mathcal{Y}}^l(j,:)||}\right). \tag{4}$$

The matrix D contains the information we need for recovering a matching among samples from both domains. One way to do this is to construct an inter-domain similarity matrix T with entries $T_{ij} = 1$ if y_j is the nearest neighbor of x_i among all observations in $\mathcal{Y}$ with respect to the distances given by (4). A reasonable alternative is to construct a soft assignment using k-nearest neighbors. However, in our experimental results we found a slight edge in performance by solving an entropic optimal transport problem [28] instead, for which D acts as the cost matrix:

$$T = \underset{T \in \mathcal{U}}{\arg\min} \langle T, D\rangle_F + \epsilon\Omega(T). \tag{5}$$

T is sometimes referred to as a coupling matrix and belongs to the set $\mathcal{U}(\boldsymbol{a},\boldsymbol{b}) = \{T \in \mathbb{R}_+^{n\times m} : T\mathbb{1}_m = \boldsymbol{a}, T^T\mathbb{1}_n = \boldsymbol{b}\}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors containing a user-defined "mass" for each observation. Matrix T is the main quantity of interest as it contains the coupling information among all the samples. The entropic regularization imposed by $\Omega(T) = \sum_{ij} T_{ij}\log(T_{ij})$ drives the solution towards less sparse solutions.

The final step of MALI projects the data into a common representation. One approach to do this is to project directly into the ambient space using the barycentric projection $x_i \mapsto \sum_j T_{ij} y_j$. This mapping is represented by h in Figure 1. Alternatively, a joint latent representation can be obtained after computing a spectral embedding [19] on the cross-domain similarity:

$$W = \begin{bmatrix} \mu W_{\mathcal{X}} & (1-\mu)W_{\mathcal{X}\mathcal{Y}} \\ (1-\mu)W_{\mathcal{X}\mathcal{Y}}^T & \mu W_{\mathcal{Y}} \end{bmatrix}, \tag{6}$$

where μ controls the preservation of the domain specific topology, and the off-diagonal blocks in W reproduce the intra-domain similarities according to the assignments in T; i.e. $W_{\mathcal{X}\mathcal{Y}} = (W_{\mathcal{X}}T + TW_{\mathcal{Y}})$. We argue this construction of W is more beneficial than keeping only the T matrix in the off-diagonal blocks, as was formulated in [16]. We present evidence for this claim in the next section.

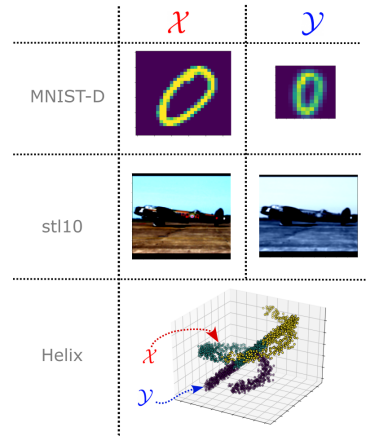

Fig. 2. **Depiction of datasets**. Three of the employed datasets highlighting the differences between the domains $\mathcal{X}$ and $\mathcal{Y}$. For the MNIST-D dataset, $\mathcal{X}$ contains the original images while $\mathcal{Y}$ contains transformed versions of the images. This also applies to the stl10 dataset. The helix dataset contains a helix ($\mathcal{X}$) and a straight line ($\mathcal{X}$).

MALI differs from methods such as KEMA [23] and the approach in [22], which directly find a latent joint representation. In contrast, MALI finds a coupling first, and then builds a joint similarity matrix, which if needed, can be used to find a latent joint representation.

IV. Experimental Results

A. Experimental setting

We now compare the performance of MALI with KEMArbf [23] and [22] which we refer to as KEMAlin since it produces similar results to those from KEMA with a linear kernel. We use the code provided by the authors[1]. We perform the comparisons for the three datasets described next and displayed in Fig. 2. **Helix**: two one-dimensional manifolds embedded in a 3D space plus noise, where one domain is a helix and the other a straight line. **MNIST-D**: one domain consists of the original MNIST digits, while the other is generated by applying multiple transformations including rotation, downscaling, and Gaussian blurring (see Fig. 2). **stl10**: a popular dataset for computer vision, where the first domain contains the original images, and we generated the second by applying brightness, gray scaling, and Gaussian blurring. We performed feature extraction using the 512 outputs after the last convolution layer in ResNet-18, a smaller version of the ResNet neural network [29].

For all of these datasets, we have access to the correspondences between points, although this information is not provided to any of the algorithms. A good alignment should map matching observations close to each other, as well as be useful for classifying unlabeled target observations employing the labeled source samples. The following two metrics meet these requirements, and have been previously employed in [10], [30], [14].

[1]https://github.com/dtuia/KEMA

1) Fraction of samples closer to the true match (FOSCTTM): Given the ground truth one-to-one correspondence knowledge, this metric computes how many samples are closer to the true match after alignment, normalized by the total size of the data. Then the average error for all the samples is computed, yielding a final score. A perfect alignment would produce a score equal to zero.
2) Label Transfer: Using the labels of the source domain, a 1-NN classifier is built after alignment and tested on the target domain. The final score corresponds to the percentage of correctly predicted labels on the target domain.

B. Metric performance

To test the performance of the methods, it is important to analyze their behavior for various levels of labeled data in the target domain. In what follows, we include the scores for two variations of MALI. MALI-S10 is obtained when the alignment representation corresponds to the 10-dimensional spectral embedding of the joint similarities W, while MALI-AS is the ambient space alignment after the barycentric projection. Since the FOSCTTM metric relies on the computation of Euclidean distances, MALI-AS scores are more likely to suffer from the curse of dimensionality.

For now, we are restricting ourselves to a particular case where $\epsilon = 0$, $a_i = 1, \forall x_i \in \mathcal{X}$ and $b_j = 1, \forall y_j \in \mathcal{Y}$. Given that for the experiments in this section both domains have the same size $n = m$, and we want to find a one-to-one coupling. Thus, in this particular setting, matrix T is a zero-one matrix.

TABLE I
FOSCTTM AVERAGE SCORES OVER 10 RUNS UNDER VARIOUS PERCENTAGES OF LABELED DATA ON THE TARGET DOMAIN.

		FOSCTTM				
Dataset	Model	100%	50%	5%	2%	1%
Helix	KEMAlin	0.298 (4)	0.308 (4)	0.251 (4)	0.241 (4)	0.242 (4)
	KEMArbf	0.137 (3)	0.130 (3)	0.131 (3)	0.116 (3)	0.130 (3)
	MALI-S10	**0.033** (1)	**0.033** (1)	**0.033** (1)	**0.034** (1)	**0.033** (1)
	MALI-AS	**0.042** (2)	**0.042** (2)	**0.042** (2)	**0.043** (2)	**0.042** (2)
MNIST-D	KEMAlin	0.334 (4)	0.333 (4)	0.352 (4)	0.378 (4)	0.330 (4)
	KEMArbf	**0.027** (2)	**0.019** (2)	**0.067** (2)	**0.071** (2)	**0.063** (2)
	MALI-S10	**0.005** (1)	**0.006** (1)	**0.018** (1)	**0.040** (1)	**0.056** (1)
	MALI-AS	0.045 (3)	0.049 (3)	0.098 (3)	0.136 (3)	0.161 (3)
stl10	KEMAlin	0.123 (4)	0.119 (4)	0.147 (4)	0.129 (3)	0.142 (3)
	KEMArbf	**0.049** (1)	**0.056** (1)	**0.087** (2)	**0.087** (1)	**0.096** (1)
	MALI-S10	**0.054** (2)	**0.060** (2)	**0.077** (1)	**0.091** (2)	**0.121** (2)
	MALI-AS	0.117 (3)	0.117 (3)	0.147 (3)	0.175 (4)	0.195 (4)

The FOSCTTM scores are reported in Table I. Here we see that MALI-S10 greatly outperforms the KEMA methods on the Helix and MNIST-D datasets across all label percentages. For the stl10 dataset, MALI-S10 is 1st when 5% of the data is labeled and 2nd for all other percentages, although it is not far behind KEMArbf in most of these instances.

Table II presents the label transfer scores. In general and in contrast with the FOSCTTM scores, we observe little discrepancy between MALI-S10 and MALI-AS. This indicates that indeed the curse of dimensionality affects MALI-AS in the FOSCTTM metric, which also explains why both MALI variants achieve closer FOSCTTM results for the 3D Helix than in the other high-dimensional datasets.

TABLE II
LABEL TRANSFER ACCURACY AVERAGED OVER 10 RUNS UNDER VARIOUS LEVELS OF LABELED DATA ON THE TARGET DOMAIN.

		Label transfer 1-NN				
Dataset	Model	100%	50%	5%	2%	1%
Helix	KEMAlin	0.915 (4)	0.873 (4)	0.811 (4)	0.828 (4)	0.845 (4)
	KEMArbf	**0.982** (1)	**0.975** (2)	0.933 (3)	0.960 (3)	0.928 (3)
	MALI-S10	**0.976** (2)	**0.976** (1)	**0.976** (1)	**0.975** (1)	**0.976** (1)
	MALI-AS	0.971 (3)	0.971 (3)	**0.971** (2)	**0.972** (2)	**0.973** (2)
MNIST-D	KEMAlin	0.243 (4)	0.216 (4)	0.188 (4)	0.202 (4)	0.243 (4)
	KEMArbf	0.812 (3)	0.835 (3)	0.635 (3)	0.632 (3)	0.654 (3)
	MALI-S10	**0.918** (2)	**0.913** (2)	**0.844** (1)	**0.781** (1)	**0.714** (1)
	MALI-AS	**0.922** (1)	**0.920** (1)	**0.836** (2)	**0.756** (2)	**0.682** (2)
RNA-ATAC	KEMAlin	0.411 (4)	0.677 (4)	0.586 (4)	0.613 (4)	0.589 (4)
	KEMArbf	0.530 (3)	0.702 (3)	0.630 (3)	0.623 (3)	0.624 (3)
	MALI-S10	**0.755** (2)	**0.743** (2)	**0.734** (2)	**0.702** (2)	**0.698** (1)
	MALI-AS	**0.780** (1)	**0.771** (1)	**0.736** (1)	**0.711** (1)	**0.695** (2)
stl10	KEMAlin	0.571 (4)	0.584 (4)	0.546 (4)	0.564 (4)	0.539 (4)
	KEMArbf	0.684 (3)	0.673 (3)	0.613 (3)	0.603 (3)	0.586 (3)
	MALI-S10	**0.879** (1)	**0.864** (1)	**0.822** (1)	**0.778** (1)	**0.717** (1)
	MALI-AS	**0.858** (2)	**0.848** (2)	**0.766** (2)	**0.690** (2)	**0.636** (2)

As an example of how an embedding is produced using MALI, we show in Figure 3 a UMAP 2-dimensional embedding of the stl10 dataset using the matrix W.

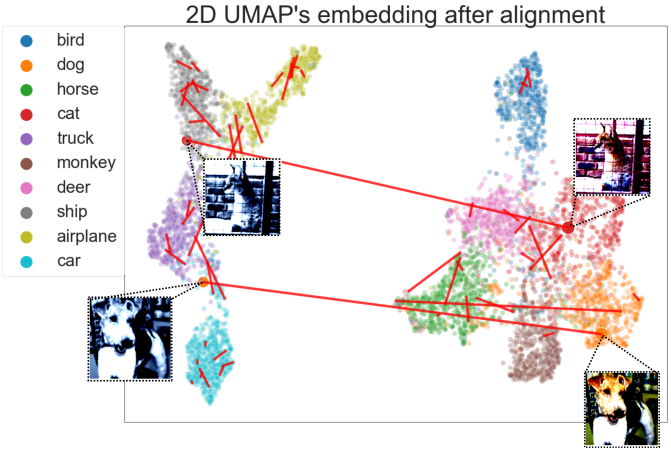

Fig. 3. **UMAP embedding of stl10 after MALI alignment**. Samples are colored by their class label and red lines connect the ground truth paired images for a subset of the data. Shorter lines are indicative of a better alignment. We also display the two images with the worst alignment score.

C. Soft assignments with entropic regularized OT

MALI is not restricted to producing one-to-one matchings as we obtained in the previous section. In many cases, we might be interested in finding a soft matching between domains. For such a task, imposing an entropic regularization by setting $\epsilon > 0$, is a natural alternative. It forces the transport plan T to find a dense solution instead of a sparse one. Thus, we can interpret the learned entries of T as a soft matching between samples in both domains.

Figure 4 exemplifies how including the entropic regularization affects the solutions. Here we show on the MNIST-D dataset how each point can be matched with multiple points with high similarity. Using the dense T matrix to construct the joint similarity matrix W results in a good embedding with relatively few large errors. This is corroborated by our metrics which show an improvement when including the regularization. This is likely because the soft assignments created with the entropy regularization are more robust to local changes in the neighborhood of a given data point.

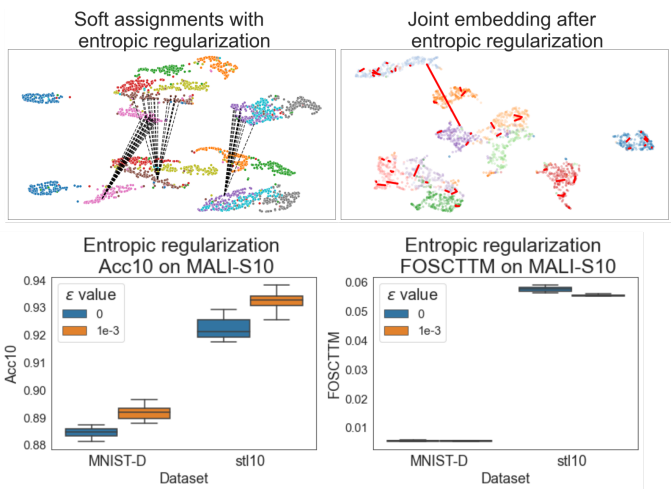

Fig. 4. **Soft matchings with entropic regularization**. The **top-left** subplot shows the assignments from individual points in $\mathcal{X}$ to $\mathcal{Y}$ in the MNIST-D dataset. Instead of matching in a one-to-one fashion, each point finds a collection of matches with high similarity. **Top-right**, we computed a UMAP embedding using the joint similarity matrix W built with a dense matrix T. To represent the goodness of alignment, we highlight randomly selected red lines connecting ground truth matches. Most of the connections are short, with longer connections being an artifact of the UMAP algorithm. Overall, the alignment not only looks qualitatively accurate, but as displayed in the bottom panels, the performance metrics score better with $\epsilon = 0.001$ than for the unregularized case ($\epsilon = 0$). Acc10 is the label transfer accuracy using a 10-NN classifier.

D. Unbalanced number of observations

So far, we have assumed that we have the same number of observations for both domains, i.e., $n = m$. The extension to the case $n \neq m$ can be handled in many ways. Here, we present two relatively straight-forward solutions. One is to rebalance the masses. Without loss of generality, we can set $b = \frac{a \times n}{m}$, where b and a are the individual masses assigned uniformly to all samples in domains $\mathcal{Y}$ and $\mathcal{X}$, respectively. This enables a soft assignment for some or all of the observations in either of the domains. Figure 5 shows an example of this.

We note that the masses can also be modified when $n = m$, resulting in soft assignments for some or all the observations. The specific problem will determine the best approach for modifying the masses. For instance, if the data density is lower for a given data region in one domain compared to its counterpart in the other domain, one may consider increasing the masses of a set of samples belonging to the low density region.

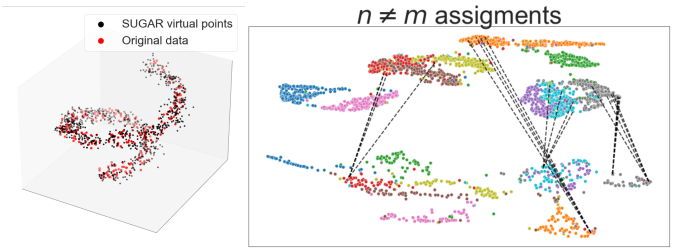

Fig. 5. **Left**, Oversampling and density equalization via SUGAR. This strategy allows to avoid rebalancing the masses, since we can synthetically reproduce the $n = m$ case. **Right**, assignments for a low-sampled $\mathcal{X}$ domain. After rebalancing, each sample from $\mathcal{X}$ is assigned to multiple counterparts in $\mathcal{Y}$ due to their higher mass.

Another simple but powerful solution to the imbalanced problem is to oversample with density equalization via a method such as SUGAR [31]. In this case, both domains will be balanced with the same amount of samples and uniform densities. And the problem is narrowed down to the $n = m$ case. The generated points can then be eliminated after performing the alignment and embedding. Figure 5 shows an example of this.

V. Conclusion

We presented MALI, a manifold alignment method capable of finding a common meaningful representation for two distinct but related domains. MALI only requires side coarse information to perform the alignment, such as discrete labels in both domains. MALI combines the diffusion geometry of the data alongside the labels to find inter-domain distances, which are then used to couple the datasets via optimal transport.

References

[1] T. Stuart and R. Satija, "Integrative single-cell analysis," *Nature reviews genetics*, vol. 20, no. 5, pp. 257–272, 2019.

[2] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck III, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija, "Comprehensive integration of single-cell data," *Cell*, vol. 177, no. 7, pp. 1888–1902, 2019.

[3] B. Thompson, *Canonical correlation analysis: Uses and interpretation*. Sage, 1984, no. 47.

[4] D. Zhai, H. Chang, S. Shan, X. Chen, and W. Gao, "Manifold alignment via corresponding projections," in *BMVC*, 2010.

[5] T. A. Abeo, X.-J. Shen, E. D. Ganaa, Q. Zhu, B.-K. Bao, and Z.-J. Zha, "Manifold alignment via global and local structures preserving pca framework," *IEEE Access*, vol. 7, pp. 38 123–38 134, 2019.

[6] C. F. Baumgartner, A. Gomez, L. M. Koch, J. R. Housden, C. Kolbitsch, J. R. McClelland, D. Rueckert, and A. P. King, "Self-aligning manifolds for matching disparate medical image datasets," in *IPMI*. Springer, 2015, pp. 363–374.

[7] R. Guerrero, C. Ledig, and D. Rueckert, "Manifold alignment and transfer learning for classification of alzheimer's disease," in *MLMI*. Springer, 2014, pp. 77–84.

[8] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls, "Semisupervised manifold alignment of multimodal remote sensing images," *IEEE Trans Geosci Remote Sens*, vol. 52, no. 12, pp. 7708–7720, 2014.

[9] F. Escolano, E. Hancock, and M. Lozano, "Graph matching through entropic manifold alignment," in *CVPR 2011*. IEEE, 2011, pp. 2417–2424.

[10] K. Cao, X. Bai, Y. Hong, and L. Wan, "Unsupervised topological alignment for single-cell multi-omics integration," *Bioinformatics*, vol. 36, no. Supplement_1, pp. i48–i56, 2020.

[11] O. Lindenbaum, A. Yeredor, M. Salhov, and A. Averbuch, "Multi-view diffusion maps," *Information Fusion*, vol. 55, pp. 127–149, 2020.

[12] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[13] Z. Cui, H. Chang, S. Shan, and X. Chen, "Generalized unsupervised manifold alignment," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[14] P. Demetci, R. Santorella, B. Sandstede, W. S. Noble, and R. Singh, "Scot: Single-cell multi-omics alignment with optimal transport," *Journal of Computational Biology*, vol. 29, no. 1, pp. 3–18, 2022.

[15] J. H. Ham, D. D. Lee, and L. K. Saul, "Learning high dimensional correspondences from low dimensional manifolds," in *ICML*, 2003.

[16] J. Ham, D. Lee, and L. Saul, "Semisupervised alignment of manifolds," in *AIStat*, 2005, pp. 120–127.

[17] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *ICML*, 2008, pp. 1120–1127.

[18] A. F. Duque, G. Wolf, and K. R. Moon, "Diffusion transport alignment," in *International Symposium on Intelligent Data Analysis*, 2023, pp. 116–129.

[19] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[20] M. Amodio and S. Krishnaswamy, "MAGAN: Aligning biological manifolds," in *ICML*, 2018, pp. 215–223.

[21] F. Aziz, A. S. Wong, and S. Chalup, "Semi-supervised manifold alignment using parallel deep autoencoders," *Algorithms*, vol. 12, no. 9, p. 186, 2019.

[22] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Twenty-second international joint conference on artificial intelligence*, 2011.

[23] D. Tuia and G. Camps-Valls, "Kernel manifold alignment for domain adaptation," *PloS one*, vol. 11, no. 2, p. e0148655, 2016.

[24] J. Wang, X. Li, and J. Du, "Label space embedding of manifold alignment for domain adaption," *Neural Processing Letters*, vol. 49, no. 1, pp. 375–391, 2019.

[25] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl Comput Harmon Anal*, vol. 21, no. 1, pp. 5–30, 2006.

[26] K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, A. van den Elzen, M. J. Hirn, R. R. Coifman *et al.*, "Visualizing structure and transitions in high-dimensional biological data," *Nature Biotechnology*, vol. 37, no. 12, pp. 1482–1492, 2019.

[27] L. Haghverdi, M. Büttner, F. A. Wolf, F. Buettner, and F. J. Theis, "Diffusion pseudotime robustly reconstructs lineage branching," *Nature methods*, vol. 13, no. 10, pp. 845–848, 2016.

[28] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *ICVPR*, 2016, pp. 770–778.

[30] K. Cao, Y. Hong, and L. Wan, "Manifold alignment for heterogeneous single-cell multi-omics data integration using pamona," *Bioinformatics*, vol. 38, no. 1, pp. 211–219, 2022.

[31] O. Lindenbaum, J. Stanley, G. Wolf, and S. Krishnaswamy, "Geometry based data generation," *NeurIPS*, vol. 31, 2018.