Cluster-Aware Matching via Laplacian Optimal Transport

Gabriel Samberg, *Department of Applied Mathematics, Tel Aviv University*
YoonHaeng Hur*, *Department of Statistics, Columbia University*
Yuehaw Khoo, *Department of Statistics, University of Chicago*
Nir Sharon, *Department of Applied Mathematics, Tel Aviv University*

Abstract

In many applications of matching, the point clouds to be matched are not merely unstructured sets of points but rather samples from distributions with an intrinsic cluster structure. In such cases, as individual points are often interchangeable within a coherent region, finding a robust region-to-region alignment is more desirable than establishing a precise point-to-point correspondence. To this end, we propose a novel approach for cluster-aware matching based on Laplacian Optimal Transport (LapOT). The key idea is to regularize the optimal transport problem with quadratic Laplacian terms constructed from similarity graphs of the point clouds, which encourages the optimal coupling to respect the cluster structure of both point sets. We also introduce Refined Simultaneous Clustering (RSC), a method that leverages the cluster-aware coupling obtained from LapOT to produce consistent partitions across the point sets, which can overcome the limitations of independent clustering and yield more stable and interpretable results. We demonstrate the effectiveness of our approach through theoretical analysis and empirical experiments, showing that LapOT indeed produces cluster-aware matching that leads to more consistent and meaningful alignments between point clouds.

1 Introduction

Matching is a fundamental problem of establishing meaningful correspondences or alignment [46, 37] between two or more sets of points drawn from potentially distinct distributions or geometric spaces. Traditionally rooted in computer vision and pattern recognition for rigid and non-rigid shape alignment [4, 29], the scope of point matching has expanded significantly with the advent of high-dimensional statistics and machine learning. Importantly, along with the rise of closely related optimal transport theory [47] and computation [32], establishing accurate point-to-point correspondences is increasingly critical in many applications, such as computational biology [40, 5], biological imaging [35, 43, 52], word translation [1, 16], to name a few.

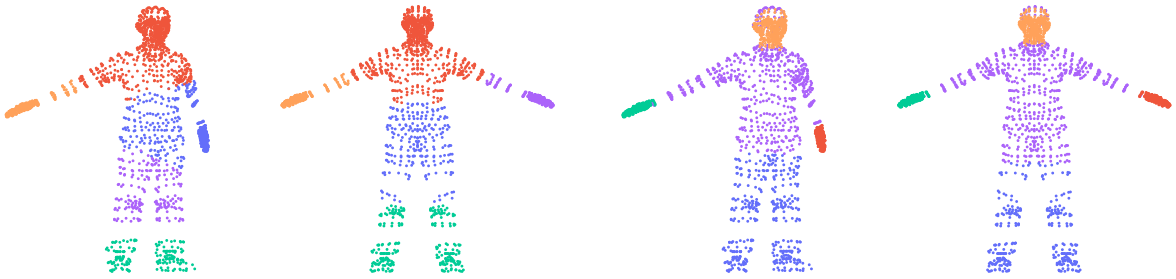
In many applications, the point clouds to be matched are not merely unstructured sets of points but rather samples from distributions with an intrinsic cluster structure. These clusters often represent meaningful regions or components of the underlying objects, such as the limbs of a human figure or functional groups of proteins. In such cases, as individual points are often interchangeable within a coherent region, finding a robust region-to-region alignment is more desirable than establishing a precise point-to-point correspondence. A natural yet naive approach to achieve such a region-level alignment would be to first cluster each point cloud independently and then match the resulting clusters. However, this two-stage approach can be extremely problematic due to the instability of clustering algorithms [3, 34, 49], leading to inconsistent partitions across the point sets, which in turn severely degrades the quality of the subsequent matching. This instability is illustrated in Figure 1(a), where independent clustering fails to produce consistent partitions for two similar shapes. This motivates the need for a cluster-aware matching framework that integrates information about the cluster structure directly into the matching process. By treating matching and clustering as coupled problems, we can leverage the shared structure of the point clouds to obtain more robust and meaningful correspondences.

*Corresponding author: yoonhaeng.hur@columbia.edu

To this end, we propose a novel approach for cluster-aware matching based on Laplacian Optimal Transport (LapOT). The key idea is to regularize the optimal transport problem with quadratic Laplacian terms that encourage the optimal coupling to respect the cluster structure of both point sets. This is achieved by constructing similarity graphs for each point cloud, where the edges encode the likelihood of points belonging to the same cluster. The resulting Laplacian regularization terms promote similar rows and columns in the optimal coupling for points that are close in their respective similarity graphs, effectively encouraging a cluster-aware matching. We demonstrate the effectiveness of our approach through theoretical analysis and empirical experiments, showing that LapOT indeed produces cluster-aware couplings that lead to more consistent and meaningful alignments between point clouds.

We also introduce a Refined Simultaneous Clustering (RSC) method that leverages the cluster-aware coupling obtained from LapOT to produce consistent partitions across the point sets. By conditioning the clustering process on the information provided by the matching, RSC can overcome the limitations of independent clustering and yield more stable and interpretable results. Figure 1(b) illustrates the improved clustering obtained by RSC, which produces consistent and meaningful clusters across both shapes.

The rest of the paper is organized as follows. In Section 2, after laying out the relevant background, we introduce the LapOT framework. Section 3 establishes theoretical guarantees for the cluster-aware properties of the optimal coupling obtained from LapOT. In Section 4, we present the RSC method and demonstrate its effectiveness through experiments on 3D shapes, followed by further applications in Section 5.



(a) Independent Clustering (b) Our Method (RSC)

Figure 1: Clustering of two 3D human shapes from the CAPOD dataset [31]. (a) is obtained by applying spectral clustering independently to the two shapes with $k = 5$ for each shape, which yields inconsistent partitions. (b) is the result of our proposed Refined Simultaneous Clustering (RSC) method, which produces consistent and meaningful clusters across both shapes learned from the cluster-aware matching obtained via Laplacian optimal transport. See Section 4 for details.

Notation. Let $\mathrm{tr}(A)$ denote the trace of a square matrix A. For matrices A, B of the same dimension, let $\langle A, B\rangle \triangleq \mathrm{tr}(A^\top B)$ denote their Frobenius inner product, with the associated Frobenius norm $\|A\|_{\mathrm{F}} \triangleq \sqrt{\langle A, A\rangle}$. For a matrix $A \in \mathbb{R}^{n\times m}$, let $\|A\|_1 \triangleq \sum_{i,j} |A_{ij}|$ and $\|A\|_\infty = \max_{i,j} |A_{ij}|$. For a vector $v \in \mathbb{R}^n$ and a symmetric positive semi-definite matrix $L \in \mathbb{R}^{n\times n}$, let $\|v\|_L \triangleq \sqrt{v^\top L v}$. Let $\mathbb{R}_+ = [0, \infty)$. For any $n \in \mathbb{N}$, let $\mathcal{S}_n$ be the set of all permutations of $[n] = \{1, \dots, n\}$, let $1_n = (1, \dots, 1) \in \mathbb{R}^n$ denote the all-ones vector, and let $\Delta_n \triangleq \{a \in \mathbb{R}^n_+ : \sum_{i=1}^n a_i = 1\}$ be the probability simplex. For $a \in \Delta_n$ and $b \in \Delta_m$, we denote by $\Pi_{a,b} \triangleq \{P \in \mathbb{R}^{n\times m}_+ : P1_m = a \text{ and } P^\top 1_n = b\}$ the set of couplings between a and b. For any $P \in \mathbb{R}^{n\times m}_+$, let $H(P) \triangleq -\sum_{i,j} P_{ij}(\log(P_{ij}) - 1)$ denote the Shannon entropy; we also use $H(v) \triangleq -\sum_i v_i(\log(v_i) - 1)$ to denote the Shannon entropy of a vector $v \in \mathbb{R}^n_+$. For two vectors $u, v \in \mathbb{R}^n$, let u/v denote the element-wise division of u by v. For a vector $v \in \mathbb{R}^n$, let $\mathrm{diag}(v)$ denote the diagonal matrix with v on the diagonal. For any $z \in \mathbb{R}^d$, let δ_z denote the Dirac measure at z.

2 Laplacian Optimal Transport

Point matching concerns finding a correspondence between two sets of points $\{X_1, \dots, X_n\}$ and $\{Y_1, \dots, Y_m\}$ defined on some sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. This section introduces Laplacian Optimal Transport (LapOT),

a novel approach for point matching that incorporates the cluster structure of the point sets. We first review relevant background on point matching and optimal transport, followed by a detailed description of LapOT and its low-rank extension.

2.1 Preliminaries: Matching, Optimal Transport, and Clustering

Quadratic Matching. Quadratic matching is one standard approach for point matching, which seeks a suitable alignment between two sets of points based on pairwise relationships. To this end, we first define two pairwise relationship matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$, which capture the relationships between the points in $\{X_1, \dots, X_n\}$ and $\{Y_1, \dots, Y_m\}$, respectively. The choice of A and B depends on the specific application and the nature of the data. For instance, when $\mathcal{X}$ is a metric space with a metric $d_{\mathcal{X}}$, we may set $A_{ii'} = d_{\mathcal{X}}(X_i, X_{i'})$; when the points are from a graph, we may use the adjacency matrix as A. Once suitable A, B are chosen, quadratic matching seeks the best alignment between them. Particularly, when $n = m$, this can be formulated as follows:

$$\min_{\sigma\in\mathcal{S}_n} \sum_{i,i'=1}^{n} |A_{ii'} - B_{\sigma(i)\sigma(i')}|^p, \tag{1}$$

where $p > 0$ is a suitable exponent. In words, (1) finds the permutation that minimizes the total discrepancy between A and B after alignment. When $n \neq m$, we may seek the best alignment in terms of couplings, leading to the following formulation:

$$\min_{\pi\in\Pi_{a,b}} \sum_{i,i'=1}^{n}\sum_{j,j'=1}^{m} |A_{ii'} - B_{jj'}|^p \pi_{ij}\pi_{i'j'}, \tag{2}$$

where $a \in \Delta_n$ and $b \in \Delta_m$ are user-specified weights to emphasize the importance of the points. In plain language, this formulation seeks a coupling that aligns A, B with the smallest total discrepancy. When $\mathcal{X}, \mathcal{Y}$ are metric spaces with A, B being the pairwise distances of the points, (2) is the Gromov-Wasserstein distance [27] between two empirical measures $\sum_{i=1}^n a_i\delta_{X_i}$ and $\sum_{j=1}^m b_j\delta_{Y_j}$. Despite its rich theoretical properties [28, 12] and practical success across various domains [33, 45, 1, 53, 7, 19], quadratic matching is NP-hard: notice that (1) is a quadratic assignment problem [24, 26], and (2) is a non-convex quadratic program.

Profile-Based Matching and Optimal Transport. Another approach to point matching is to assign suitable profiles to the points and match them if the profiles are similar, which has been considered by [11] and [20] for graph and object matching, respectively. Concretely, we assign to X_i the weighted distribution $\mu_i \triangleq \sum_{i'=1}^n a_{i'}\delta_{A_{ii'}}$ as its profile, called a degree profile in [11] when A is the adjacency matrix and a distance profile in [20] when A consists of the pairwise distances. Similarly, Y_j's profile is defined as $\nu_j \triangleq \sum_{j'=1}^m b_{j'}\delta_{B_{jj'}}$. Here, a_i's and b_j's are user-specified weights to emphasize the importance of the points. Now that the profiles are one-dimensional measures, we construct a matrix $C \in \mathbb{R}^{n\times m}$ whose entries measure the similarity between the profiles based on a suitable discrepancy, for instance, $C_{ij} = W_1(\mu_i, \nu_j)$ when using the Wasserstein distance. Then, the point matching problem can be reduced to the following optimization problem:

$$\min_{\pi\in\Pi_{a,b}} \langle \pi, C\rangle, \tag{3}$$

which is a discrete optimal transport (OT) problem between $\sum_{i=1}^n a_i\delta_{X_i}$ and $\sum_{j=1}^m b_j\delta_{Y_j}$, with C serving as a transport cost matrix. Unlike quadratic matching, (3) is a linear program that can be solved by polynomial-time algorithms. Of course, solving (3) can still be computationally expensive for large-scale problems. To tackle this, the following entropic regularized optimal transport problem [51, 10, 32] has been proposed: given any $a \in \Delta_n$, $b \in \Delta_m$, and $C \in \mathbb{R}^{n\times m}$,

$$\mathsf{OT}_\varepsilon(a, b, C) \triangleq \min_{\pi\in\Pi_{a,b}} \langle \pi, C\rangle - \varepsilon H(\pi). \tag{4}$$

Adding the entropic regularization term to (3) leads to a strongly convex problem. Crucially, (4) can be solved efficiently via the Sinkhorn algorithm that iteratively scales the kernel matrix $e^{-C/\varepsilon}$ so that its row and column sums become a and b, respectively.

Clustering via Graph Cuts. One common approach to clustering is to view the data as a graph and partition the graph into clusters by minimizing the graph cuts. To be specific, consider an undirected graph with n vertices with an adjacency matrix $K \in \mathbb{R}^{n\times n}$. Here, the idea is to cluster the vertices of the graph by minimizing the total weight of edges that are cut by the partition. Given a partition $\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_k = [n]$ of the vertices, each partition $\mathcal{I}_j$ represents a cluster, and the total weight of edges that are cut by $\mathcal{I}_j$ is given by $\sum_{i\in\mathcal{I}_j, i'\notin\mathcal{I}_j} K_{ii'}$. Hence, the sum of these weights for all partitions, weighted by the size of the partitions, can be written as

$$\sum_{j=1}^{k} \frac{\sum_{i\in\mathcal{I}_j, i'\notin\mathcal{I}_j} K_{ii'}}{|\mathcal{I}_j|},$$

which is called the ratio cut of the partition. This can be rewritten as $\langle P, LP\rangle$, where $L = \operatorname{diag}(K1_n) - K$ is the unnormalized graph Laplacian and $P \in \mathbb{R}^{n\times k}$ is the partition matrix defined by $P_{ij} = \frac{1}{\sqrt{|\mathcal{I}_j|}}$ if vertex i belongs to partition $\mathcal{I}_j$ and $P_{ij} = 0$ otherwise. Accordingly, minimizing the ratio cut can be formulated as the following optimization problem:

$$\min_{P\in\mathbb{R}^{n\times k}} \langle P, LP\rangle, \quad \text{subject to } P_{ij} = \begin{cases} \frac{1}{\sqrt{|\mathcal{I}_j|}} & \text{if vertex } i \text{ belongs to partition } \mathcal{I}_j, \\ 0 & \text{otherwise.} \end{cases}$$

This, however, is a combinatorial optimization problem that is NP-hard. A common relaxation is to relax the constraint on P to be $P^\top P = I_k$, which leads to the following optimization problem:

$$\min_{P\in\mathbb{R}^{n\times k}} \langle P, LP\rangle, \quad \text{subject to } P^\top P = I_k,$$

which boils down to finding the k eigenvectors of L corresponding to the k smallest eigenvalues. This is the backbone of spectral clustering [2, 8, 48], which is a widely used clustering algorithm in practice. Instability of clustering [3, 34, 49] can lead to unreliable scientific findings and downstream decisions filled with uncertainty [42, 44], which has motivated a huge body of work on uncertainty quantification for clustering [23, 17, 25, 18, 50, 22, 30]. However, these methods are not designed to produce consistent partitions across multiple point sets, which is crucial for cluster-aware matching. This motivates the need for a new approach that can leverage the shared structure of the point clouds to obtain more robust and meaningful correspondences.

2.2 Laplacian Optimal Transport (LapOT)

We introduce Laplacian Optimal Transport (LapOT), a novel approach for point matching that incorporates the cluster structure of the point sets. The key idea is to regularize the optimal transport problem (4) with quadratic Laplacian regularization terms that encourage the optimal coupling to carry the cluster structure of both point sets.

To this end, we require the user to specify the cluster structure of the point sets in the form of similarity graphs, represented by the symmetric matrices $K_X \in \mathbb{R}^{n\times n}$ and $K_Y \in \mathbb{R}^{m\times m}$, for the point sets $\{X_1, \ldots, X_n\}$ and $\{Y_1, \ldots, Y_m\}$, respectively. The entry $(K_X)_{ii'}$ (resp. $(K_Y)_{jj'}$) represents the similarity between X_i and $X_{i'}$ (resp. Y_j and $Y_{j'}$), where the larger value indicates higher similarity and thus being more likely to belong to the same cluster. The choice of K_X and K_Y depends on the specific application and the nature of the data. For instance, when $\mathcal{X}$ is a metric space with a metric $d_{\mathcal{X}}$, we may set $(K_X)_{ii'} = e^{-d_{\mathcal{X}}(X_i, X_{i'})/\sigma}$ for some $\sigma > 0$; when the points are from a graph, we may use the usual binary adjacency matrix as K_X. The same applies to K_Y for the point set $\{Y_1, \ldots, Y_m\}$.

Now, we want the optimal coupling to reflect the cluster structure encoded by K_X and K_Y. Concretely, when $(K_X)_{ii'}$ (resp. $(K_Y)_{jj'}$) is large, we want to encourage the i-th and i'-th rows (resp. j-th and j'-th columns) of the optimal coupling to be similar to each other. In other words, if X_i and $X_{i'}$ are similar to each other, we want them to be matched to similar distributions over $\{Y_1, \ldots, Y_m\}$; the same applies to Y_j's. To do so, we define $L_X = \operatorname{diag}(K_X 1_n) - K_X$ and $L_Y = \operatorname{diag}(K_Y 1_m) - K_Y$, which are the unnormalized graph Laplacians of the similarity graphs defined by K_X and K_Y, respectively. Then, for a coupling $\pi \in \Pi_{a,b}$,

consider the following terms:

$$\begin{aligned}\langle \pi, L_X \pi \rangle &= \mathrm{tr}(\pi^\top L_X \pi) = \frac{1}{2} \sum_{i,i'=1}^{n} (K_X)_{ii'} \|\pi_i - \pi_{i'}\|_2^2, \\ \langle \pi, \pi L_Y \rangle &= \mathrm{tr}(\pi L_Y \pi^\top) = \frac{1}{2} \sum_{j,j'=1}^{m} (K_Y)_{jj'} \|\pi_j^\top - \pi_{j'}^\top\|_2^2.\end{aligned} \tag{5}$$

Here, π_i and $\pi_j^\top$ are the i-th row and j-th column of the coupling π, respectively. By adding these terms to (4), we encourage the optimal coupling to have similar rows for similar X_i's and similar columns for similar Y_j's. Accordingly, the optimal coupling is incentivized to carry the cluster structure of both point sets defined by K_X and K_Y.

In summary, given any $a \in \Delta_n$, $b \in \Delta_m$, $C \in \mathbb{R}^{n \times m}$, and L_X, L_Y defined as above, we define LapOT as follows:

$$\mathsf{LapOT}_{\lambda_x, \lambda_y, \lambda}(a, b, C, L_X, L_Y) \triangleq \min_{\pi \in \Pi_{a,b}} \langle \pi, C \rangle + \lambda_x \langle \pi, L_X \pi \rangle + \lambda_y \langle \pi, \pi L_Y \rangle - \lambda H(\pi), \tag{6}$$

where $\lambda_x, \lambda_y, \lambda > 0$ are hyperparameters that control the strength of the regularization. We can view LapOT as a generalization of entropic OT: when $\lambda_x = \lambda_y = 0$, LapOT reduces to entropic OT. By tuning λ_x, λ_y, we can control the extent to which the optimal coupling is encouraged to carry the cluster structure of the point sets, which can be beneficial for downstream tasks such as clustering and alignment.

Optimization of LapOT. Note that the objective function of (6) consists of the convex quadratic terms—as the graph Laplacian is positive semidefinite—and the strongly convex entropic regularization term. Hence, (6) is a strongly convex optimization problem that can be solved by general-purpose convex optimization solvers. However, such solvers, which usually rely on interior point methods, can be inefficient for large-scale problems. [21] proposed specialized algorithms for smooth objective functions with entropic regularization over couplings, which encompasses (6) as a special case. Their algorithms call the Sinkhorn algorithm as a subroutine, and the number of Sinkhorn calls is independent of the coupling size nm if λ is sufficiently large compared to the norm of $\lambda_x L_X + \lambda_y L_Y$ or depends logarithmically on nm if λ is sufficiently small (including the case $\lambda = 0$). Therefore, the algorithms are far more efficient than general-purpose convex optimization solvers, especially for large-scale problems, and we use them to solve (6) in our experiments. We refer the readers to [21] for more details on the algorithms and their theoretical guarantees.

Remark 1. The Laplacian regularization terms in (5) were also considered in [14] and [9] in the context of domain adaptation. However, they do not study the cluster-awareness property of this regularization, which is the main focus of this paper.

2.3 Low-Rank Extension of Laplacian Optimal Transport

As LapOT encourages the optimal coupling to carry the cluster structure of the point sets, it is natural to expect that the optimal coupling of LapOT enjoys a low-rank structure. While we will formally justify this in Section 3, we propose a method to directly impose a low-rank constraint on the optimal coupling of LapOT, which can be beneficial for large-scale problems.

In the optimal transport literature, low-rank structure of couplings has been proposed and studied by [15, 39, 38] based on the following notion of non-negative rank:

$$\mathrm{rk}_+(M) := \min \left\{ r \in \mathbb{N} : \exists R_1, \ldots, R_r \in \mathbb{R}_+^{n \times m} \text{ s.t. } \mathrm{rk}(R_1) = \cdots = \mathrm{rk}(R_r) = 1 \text{ and } M = \sum_{i=1}^{r} R_i \right\}, \tag{7}$$

where rk is the usual rank of a matrix. The idea of low-rank optimal transport is to solve the optimal transport problem by imposing a non-negative rank constraint on the coupling, namely, the set $\Pi_{a,b}$ of all couplings is replaced by the set of couplings with a non-negative rank of at most r defined as

$$\Pi_{a,b}(r) \triangleq \{P \in \Pi_{a,b} : \mathrm{rk}_+(P) \leq r\}.$$

The feasible set becomes non-convex, and the resulting optimization problem is no longer convex. In [39], instead of directly optimizing over $\Pi_{a,b}(r)$, the following observation is made:

$$\Pi_{a,b}(r) = \bigcup_{g \in \Delta_r^+} \Pi_{a,g,b} \quad \text{where} \quad \Pi_{a,g,b} = \left\{ U \mathrm{diag}(1_r/g) V^\top : U \in \Pi_{a,g} \text{ and } V \in \Pi_{b,g} \right\}, \tag{8}$$

which leads to the following reformulation of the low-rank optimal transport problem:

$$\min_{\pi \in \Pi_{a,b}(r)} \langle \pi, C \rangle = \min_{(U,V,g) \in \mathcal{C}(a,b,r)} \langle U \mathrm{diag}(1_r/g) V^\top, C \rangle,$$

where

$$\mathcal{C}(a,b,r) = \{(U,V,g) \in \mathbb{R}_+^{n \times r} \times \mathbb{R}_+^{m \times r} \times \mathbb{R}_+^r : U 1_r = a, V 1_r = b, U^\top 1_n = V^\top 1_m = g\}. \tag{9}$$

While the right-hand side of (8) is non-convex as the objective function is not jointly convex in U, V, g, one can efficiently utilize Dykstra's algorithm [13] to solve the problem as a subroutine of a mirror descent algorithm with KL divergence. [39] also considers the entropic regularized version of the low-rank optimal transport problem (8) by adding individual entropic regularization terms for U, V, g to the objective function, which leads to the following optimization problem:

$$\min_{(U,V,g) \in \mathcal{C}(a,b,r)} \langle U \mathrm{diag}(1_r/g) V^\top, C \rangle - \lambda H(U) - \lambda H(V) - \lambda H(g).$$

Accordingly, a natural extension of LapOT to the low-rank setup can be formulated as follows:

$$\min_{(U,V,g) \in \mathcal{C}(a,b,r)} q(U \mathrm{diag}(1_r/g) V^\top) - \lambda H(U) - \lambda H(V) - \lambda H(g), \tag{10}$$

where $q \colon \mathbb{R}_+^{n \times m} \to \mathbb{R}$ is the quadratic objective function of LapOT defined as

$$q(\pi) = \langle \pi, C \rangle + \lambda_x \langle \pi, L_X \pi \rangle + \lambda_y \langle \pi, \pi L_Y \rangle.$$

As in [39], we can efficiently solve (10) by utilizing Dykstra's algorithm as a subroutine of a mirror descent algorithm with KL divergence. We defer the details of the optimization scheme to Section A in the appendix.

3 Theory

This section studies the theoretical properties of Laplacian optimal transport. As noted earlier, the Laplacian regularization terms in (5) encourage the optimal coupling to be smooth with respect to the graph structures of L_X and L_Y. Therefore, if L_X and L_Y are induced by graphs with multiple connected components, then the optimal coupling is expected to be block-constant with respect to the partitions induced by the connected components, which leads to a low-rank structure of the optimal coupling.

Our main result, Theorem 1, provides a non-asymptotic bound on the deviation of the optimal coupling from its block-averaged version with respect to the partitions induced by the connected components of L_X and L_Y. This result shows that the optimal coupling is close to a block-constant matrix when (1) the cost matrix is close to a block-constant matrix, (2) the regularization parameters λ_x and λ_y are large, or (3) the spectral gaps of L_X and L_Y are large.

Theorem 1. *Let $\pi^\star$ be the solution to $\mathsf{LapOT}_{\lambda_x,\lambda_y,\lambda}(a,b,C,L_X,L_Y)$ with $\lambda_x, \lambda_y > 0$. Assume the following.*

(i) L_X and L_Y are induced by graphs with r and s connected components, where $\{A_\alpha\}_{\alpha=1}^r$ and $\{B_\beta\}_{\beta=1}^s$ are the corresponding partitions of the index sets $[n]$ and $[m]$, respectively.

(ii) a and b are block-constant with respect to the partitions $\{A_\alpha\}_{\alpha=1}^r$ and $\{B_\beta\}_{\beta=1}^s$, respectively.

Let $L_X = \Phi_X \Lambda_X \Phi_X^\top$ and $L_Y = \Phi_Y \Lambda_Y \Phi_Y^\top$ be their orthogonal decompositions, where Λ_X and Λ_Y are diagonal matrices consisting of the eigenvalues $\mu_1^X \le \cdots \le \mu_n^X$ and $\mu_1^Y \le \cdots \le \mu_m^Y$ of L_X and L_Y, respectively. For any $\ell \in [n]$ and $h \in [m]$, define the projection matrices as $P_{X,\ell} \triangleq \Phi_{X,\ell} \Phi_{X,\ell}^\top$ and $P_{Y,h} \triangleq \Phi_{Y,h} \Phi_{Y,h}^\top$, where $\Phi_{X,\ell}$ and $\Phi_{Y,h}$ are the submatrices consisting of the first ℓ and h columns of Φ_X and Φ_Y, respectively. Then,

$P_{X,r}\pi^\star$ and $\pi^\star P_{Y,s}$ are obtained by block-averaging the rows and columns of $\pi^\star$ with respect to the partitions $\{A_\alpha\}_{\alpha=1}^r$ and $\{B_\beta\}_{\beta=1}^s$, respectively. Moreover, we have

$$\|\pi^\star - P_{X,r}\pi^\star\|_{\mathrm{F}}^2 \le \frac{\|P_{X,r}C - C\|_\infty}{\lambda_x \mu_{r+1}^X} \quad \textit{and} \quad \|\pi^\star - \pi^\star P_{Y,s}\|_{\mathrm{F}}^2 \le \frac{\|CP_{Y,s} - C\|_\infty}{\lambda_y \mu_{s+1}^Y}. \tag{11}$$

Proof. From spectral graph theory, e.g., Proposition 2 of [48], notice that (i) implies that the eigenspaces of L_X and L_Y corresponding to the zero eigenvalues ($\mu_1^X = \cdots = \mu_r^X = 0$ and $\mu_1^Y = \cdots = \mu_s^Y = 0$) are spanned by the indicator vectors of the connected components of the graphs. Accordingly, we have $L_X\Phi_{X,r} = 0$ and $L_Y\Phi_{Y,s} = 0$. Also, we deduce that left multiplication by $P_{X,r}$ and right multiplication by $P_{Y,s}$ are the operators that average matrices with respect to the partitions $\{A_\alpha\}_{\alpha=1}^r$ and $\{B_\beta\}_{\beta=1}^s$, respectively.

Now, we compare the objective values of $\pi^\star$ and $P_{X,r}\pi^\star$ in $\mathsf{LapOT}_{\lambda_x,\lambda_y,\lambda}(a,b,C,L_X,L_Y)$. Notice that $P_{X,r}\pi^\star \in \Pi_{a,b}$. To see this, observe that (ii) implies $P_{X,r}a = a$ and $P_{Y,s}b = b$. Hence,

$$P_{X,r}\pi^\star 1_m = P_{X,r}a = a \quad \text{and} \quad (P_{X,r}\pi^\star)^\top 1_n = (\pi^\star)^\top P_{X,r}1_n = (\pi^\star)^\top 1_n = b.$$

As $L_X P_{X,r} = 0$, we have $\langle P_{X,r}\pi^\star, L_X P_{X,r}\pi^\star\rangle = 0$. By the optimality of $\pi^\star$, we have

$$\langle \pi^\star, C\rangle + \lambda_x\langle \pi^\star, L_X\pi^\star\rangle + \lambda_y\langle \pi^\star, \pi^\star L_Y\rangle - \lambda H(\pi^\star) \le \langle P_{X,r}\pi^\star, C\rangle + \lambda_y\langle P_{X,r}\pi^\star, P_{X,r}\pi^\star L_Y\rangle - \lambda H(P_{X,r}\pi^\star).$$

As $\langle P_{X,r}\pi^\star, C\rangle = \langle \pi^\star, P_{X,r}C\rangle$, we have

$$\lambda(H(P_{X,r}\pi^\star) - H(\pi^\star)) + \lambda_x\langle \pi^\star, L_X\pi^\star\rangle \le \langle \pi^\star, P_{X,r}C - C\rangle + \lambda_y\langle P_{X,r}\pi^\star, P_{X,r}\pi^\star L_Y\rangle - \lambda_y\langle \pi^\star, \pi^\star L_Y\rangle.$$

Since $P_{X,r}\pi^\star$ is obtained by block-averaging the rows of $\pi^\star$ with respect to the partition $\{A_\alpha\}_{\alpha=1}^r$, we have $H(P_{X,r}\pi^\star) \ge H(\pi^\star)$ due to the concavity of H. Also, we observe $\langle P_{X,r}\pi^\star, P_{X,r}\pi^\star L_Y\rangle \le \langle \pi^\star, \pi^\star L_Y\rangle$. To see this, let ϕ_i be the i-th column of Φ_X. Then,

$$\begin{aligned}
\langle \pi^\star, \pi^\star L_Y\rangle - \langle P_{X,r}\pi^\star, P_{X,r}\pi^\star L_Y\rangle &= \mathrm{tr}(\pi^\star L_Y(\pi^\star)^\top) - \mathrm{tr}(\Phi_{X,r}^\top \pi^\star L_Y(\pi^\star)^\top \Phi_{X,r}) \\
&= \sum_{i=r+1}^n \phi_i^\top \pi^\star L_Y(\pi^\star)^\top \phi_i \\
&\ge 0,
\end{aligned}$$

where the inequality holds because $\pi^\star L_Y(\pi^\star)^\top$ is positive semi-definite. Meanwhile,

$$\langle \pi^\star, P_{X,r}C - C\rangle \le \|\pi^\star\|_1 \|P_{X,r}C - C\|_\infty = \|P_{X,r}C - C\|_\infty.$$

Hence,

$$\lambda_x\langle \pi^\star, L_X\pi^\star\rangle \le \|P_{X,r}C - C\|_\infty. \tag{12}$$

Now, we derive a lower bound on $\lambda_x\langle \pi^\star, L_X\pi^\star\rangle$. Let $\tilde\pi = \Phi_X^\top\pi^\star$ be the representation of $\pi^\star$ in the eigenbasis of L_X. Then, we have

$$\langle \pi^\star, L_X\pi^\star\rangle = \mathrm{tr}((\pi^\star)^\top L_X\pi^\star) = \mathrm{tr}(\tilde\pi^\top\Lambda_X\tilde\pi) = \sum_{i=1}^n\sum_{j=1}^m \mu_i^X\tilde\pi_{ij}^2 \ge \mu_{r+1}^X \sum_{i=r+1}^n\sum_{j=1}^m \tilde\pi_{ij}^2.$$

Meanwhile,

$$P_{X,r}\pi^\star = \Phi_{X,r}\Phi_{X,r}^\top\pi^\star = \Phi_X \begin{bmatrix}\Phi_{X,r}^\top \\ 0\end{bmatrix}\pi^\star = \Phi_X\begin{bmatrix}\tilde\pi_{1:r} \\ 0\end{bmatrix},$$

where $\tilde\pi_{1:r}$ is the submatrix of $\tilde\pi$ containing the first r rows. Hence, from $\pi^\star = \Phi_X\tilde\pi$, we have

$$\|\pi^\star - P_{X,r}\pi^\star\|_{\mathrm{F}}^2 = \sum_{i=r+1}^n\sum_{j=1}^m \tilde\pi_{ij}^2.$$

Hence, $\mu_{r+1}^X\|\pi^\star - P_{X,r}\pi^\star\|_{\mathrm{F}}^2 \le \langle \pi^\star, L_X\pi^\star\rangle$, which, together with (12), implies the first inequality in (11). The second inequality in (11) can be proved similarly by comparing the objective values of $\pi^\star$ and $\pi^\star P_{Y,s}$. □

From the bounds in (11), we immediately deduce that $\pi^\star = P_{X,r}\pi^\star$ and $\pi^\star = \pi^\star P_{Y,s}$ if $P_{X,r}C = C$ and $CP_{Y,s} = C$, respectively. In other words, if the matching cost is determined at the cluster level, then the optimal coupling indeed captures the cluster-level matching due to the Laplacian regularization terms and thus admits a low-rank structure with non-negative rank, defined in (7), bounded by the number of clusters. This is summarized in the following corollary.

Corollary 1. *In Theorem 1, we deduce the following.*

(i) If C is row block-constant with respect to the partition $\{A_\alpha\}_{\alpha=1}^r$, namely, $P_{X,r}C = C$, then $\pi^\star$ is row block-constant with respect to the partition $\{A_\alpha\}_{\alpha=1}^r$, namely, $\pi^\star = P_{X,r}\pi^\star$, and $\mathrm{rk}_+(\pi^\star) \le r$.

(ii) If C is column block-constant with respect to the partition $\{B_\beta\}_{\beta=1}^s$, namely, $CP_{Y,s} = C$, then $\pi^\star$ is column block-constant with respect to the partition $\{B_\beta\}_{\beta=1}^s$, namely, $\pi^\star = \pi^\star P_{Y,s}$, and $\mathrm{rk}_+(\pi^\star) \le s$.

(iii) If both (i) and (ii) hold, or equivalently, $P_{X,r}CP_{Y,s} = C$, then $\pi^\star$ is block-constant with respect to the product partition $\{A_\alpha \times B_\beta\}_{\alpha\in[r],\beta\in[s]}$, namely, $\pi^\star = P_{X,r}\pi^\star P_{Y,s}$, and $\mathrm{rk}_+(\pi^\star) \le \min\{r, s\}$.

Meanwhile, from the bounds in (11), we deduce that $\pi^\star$ converges to its block-averaged version as λ_x and λ_y increase, showing that stronger Laplacian regularization leads to a more pronounced low-rank structure of the optimal coupling. Indeed, at the limit of $\lambda_x \to \infty$ or $\lambda_y \to \infty$, the optimal coupling converges to its block-averaged version. In this case, we can further characterize the limit of the optimal coupling as follows.

Proposition 1. *In Theorem 1, we deduce the following.*

(i) If $\lambda_x \to \infty$, then $\pi^\star$ converges as follows:

$$\lim_{\lambda_x\to\infty} \pi^\star = \operatorname*{arg\,min}_{\pi\in\Pi_{a,b}} \langle \pi, P_{X,r}C\rangle + \lambda_y\langle \pi, \pi L_Y\rangle - \lambda H(\pi).$$

(ii) If $\lambda_y \to \infty$, then $\pi^\star$ converges as follows:

$$\lim_{\lambda_y\to\infty} \pi^\star = \operatorname*{arg\,min}_{\pi\in\Pi_{a,b}} \langle \pi, CP_{Y,s}\rangle + \lambda_x\langle \pi, L_X\pi\rangle - \lambda H(\pi).$$

(iii) If $\lambda_x, \lambda_y \to \infty$, then $\pi^\star$ converges as follows:

$$\lim_{\lambda_x,\lambda_y\to\infty} \pi^\star = \operatorname*{arg\,min}_{\pi\in\Pi_{a,b}} \langle \pi, P_{X,r}CP_{Y,s}\rangle - \lambda H(\pi).$$

Proof. To show (i), let $\pi^\star_{\lambda_x}$ denote the optimal coupling for a given λ_x. Now, consider a sequence $\{\lambda_x^{(k)}\}_{k\in\mathbb{N}}$ such that $\lambda_x^{(k)} \to \infty$ as $k \to \infty$. Since $\Pi_{a,b}$ is compact, there exists a convergent subsequence of $\{\pi^\star_{\lambda_x^{(k)}}\}_{k\in\mathbb{N}}$; by swapping the original sequence with its convergent subsequence, we may assume that $\{\pi^\star_{\lambda_x^{(k)}}\}_{k\in\mathbb{N}}$ converges to some $\tilde\pi \in \Pi_{a,b}$. We now claim

$$\tilde\pi = \operatorname*{arg\,min}_{\pi\in\Pi_{a,b}} \langle \pi, P_{X,r}C\rangle + \lambda_y\langle \pi, \pi L_Y\rangle - \lambda H(\pi) =: \pi^\star_\infty.$$

We first note that $\pi^\star_\infty = P_{X,r}\pi^\star_\infty$. To see this, as in the proof of Theorem 1, observe that $P_{X,r}\pi^\star_\infty \in \Pi_{a,b}$, $H(P_{X,r}\pi^\star_\infty) \ge H(\pi^\star_\infty)$, and $\langle P_{X,r}\pi^\star_\infty, P_{X,r}\pi^\star_\infty L_Y\rangle \le \langle \pi^\star_\infty, \pi^\star_\infty L_Y\rangle$. Hence,

$$\langle P_{X,r}\pi^\star_\infty, P_{X,r}C\rangle + \lambda_y\langle P_{X,r}\pi^\star_\infty, P_{X,r}\pi^\star_\infty L_Y\rangle - \lambda H(P_{X,r}\pi^\star_\infty) \le \langle \pi^\star_\infty, P_{X,r}C\rangle + \lambda_y\langle \pi^\star_\infty, \pi^\star_\infty L_Y\rangle - \lambda H(\pi^\star_\infty),$$

where we also use $\langle P_{X,r}\pi^\star_\infty, P_{X,r}C\rangle = \langle \pi^\star_\infty, P_{X,r}C\rangle$. By the uniqueness of the minimizer $\pi^\star_\infty$, we conclude that $\pi^\star_\infty = P_{X,r}\pi^\star_\infty$.

For notational convenience, let $\pi^\star_{\lambda_x^{(k)}} = \pi^\star_k$ for $k \in \mathbb{N}$. Then, by the optimality of $\pi^\star_k$, we have

$$\langle \pi^\star_k, C\rangle + \lambda_x^{(k)}\langle \pi^\star_k, L_X\pi^\star_k\rangle + \lambda_y\langle \pi^\star_k, \pi^\star_k L_Y\rangle - \lambda H(\pi^\star_k) \le \langle \pi^\star_\infty, C\rangle + \lambda_y\langle \pi^\star_\infty, \pi^\star_\infty L_Y\rangle - \lambda H(\pi^\star_\infty), \tag{13}$$

where we use $\langle \pi^\star_\infty, L_X \pi^\star_\infty \rangle = 0$ from $\pi^\star_\infty = P_{X,r}\pi^\star_\infty$. Now, we claim

$$\lim_{k\to\infty} \left(\langle \pi^\star_k, C\rangle + \lambda_x^{(k)} \langle \pi^\star_k, L_X \pi^\star_k\rangle + \lambda_y \langle \pi^\star_k, \pi^\star_k L_Y\rangle - \lambda H(\pi^\star_k) \right) = \langle \tilde{\pi}, P_{X,r} C\rangle + \lambda_y \langle \tilde{\pi}, \tilde{\pi} L_Y\rangle - \lambda H(\tilde{\pi}). \tag{14}$$

To see this, use (11) to deduce that

$$\lim_{k\to\infty} P_{X,r}\pi^\star_k = \lim_{k\to\infty} \pi^\star_k = \tilde{\pi}.$$

Hence,

$$\lim_{k\to\infty} \langle \pi^\star_k, C\rangle = \lim_{k\to\infty} \langle P_{X,r}\pi^\star_k, C\rangle = \lim_{k\to\infty} \langle \pi^\star_k, P_{X,r} C\rangle = \langle \tilde{\pi}, P_{X,r} C\rangle \tag{15}$$

and

$$\lim_{k\to\infty} \left(\lambda_y \langle \pi^\star_k, \pi^\star_k L_Y\rangle - \lambda H(\pi^\star_k) \right) = \lambda_y \langle \tilde{\pi}, \tilde{\pi} L_Y\rangle - \lambda H(\tilde{\pi}). \tag{16}$$

Meanwhile, as in the proof of Theorem 1, we have the following from the optimality of $\pi^\star_k$:

$$\lambda_x^{(k)} \langle \pi^\star_k, L_X \pi^\star_k\rangle \le \langle P_{X,r}\pi^\star_k, C\rangle + \lambda_y \langle P_{X,r}\pi^\star_k, P_{X,r}\pi^\star_k L_Y\rangle - \lambda H(P_{X,r}\pi^\star_k) - \langle \pi^\star_k, C\rangle - \lambda_y \langle \pi^\star_k, \pi^\star_k L_Y\rangle + \lambda H(\pi^\star_k),$$

where the right-hand side converges to 0 as $k \to \infty$ by the previous two limits. Hence,

$$\lim_{k\to\infty} \lambda_x^{(k)} \langle \pi^\star_k, L_X \pi^\star_k\rangle = 0. \tag{17}$$

Combining (15), (16), and (17), we obtain (14). Then, combining (13) and (14), we have

$$\langle \tilde{\pi}, P_{X,r} C\rangle + \lambda_y \langle \tilde{\pi}, \tilde{\pi} L_Y\rangle - \lambda H(\tilde{\pi}) \le \langle \pi^\star_\infty, P_{X,r} C\rangle + \lambda_y \langle \pi^\star_\infty, \pi^\star_\infty L_Y\rangle - \lambda H(\pi^\star_\infty),$$

where we use $\langle \pi^\star_\infty, C\rangle = \langle \pi^\star_\infty, P_{X,r} C\rangle$ from $\pi^\star_\infty = P_{X,r}\pi^\star_\infty$. By the uniqueness of the minimizer $\pi^\star_\infty$, we conclude that $\tilde{\pi} = \pi^\star_\infty$.

Hence, we have shown that for any sequence $\{\lambda_x^{(k)}\}_{k\in\mathbb{N}}$ such that $\lambda_x^{(k)} \to \infty$, the corresponding sequence of optimal couplings $\{\pi^\star_{\lambda_x^{(k)}}\}_{k\in\mathbb{N}}$ has a convergent subsequence that converges to $\pi^\star_\infty$. As the limit is independent of the choice of the sequence $\{\lambda_x^{(k)}\}_{k\in\mathbb{N}}$, we conclude that $\pi^\star_{\lambda_x} \to \pi^\star_\infty$ as $\lambda_x \to \infty$. This proves (i). The proofs of (ii) and (iii) follow similarly. □

The above results describe an idealized setting in which clusters are represented as connected components of the similarity graphs. In the numerical experiments below, however, the similarity graphs are constructed from RBF kernels and are typically connected, while the degree-based marginals need not be block-constant. In this setting, the appropriate analog of block-constancy is low-frequency smoothness with respect to the graph Laplacians. The following result makes this connection precise: for arbitrary graph Laplacians, the LapOT solution is close to its projection onto the low-frequency eigenspaces of the two graphs, with an error controlled by the regularization parameters and the corresponding spectral gaps. This provides a theoretical explanation for why the coupling obtained by LapOT can be used in RSC to reveal approximate cluster structure even when the input graphs are connected.

Proposition 2. *Let $\pi^\star$ be the solution of $\mathsf{LapOT}_{\lambda_x,\lambda_y,\lambda}(a, b, C, L_X, L_Y)$, where L_X and L_Y are arbitrary graph Laplacians having the eigenvalues $\mu_1^X \le \cdots \le \mu_n^X$ and $\mu_1^Y \le \cdots \le \mu_m^Y$ of L_X and L_Y, respectively. Let $P_{X,\ell}$ and $P_{Y,h}$ denote the orthogonal projections onto the first ℓ and h eigenvectors of L_X and L_Y, respectively, as in Theorem 1. Define*

$$\tau_\lambda(C) = \min_{\pi\in\Pi_{a,b}} \{\langle \pi, C\rangle - \lambda H(\pi)\}.$$

Then, for any $\ell < n$ and $h < m$, we have

$$\|\pi^\star - P_{X,\ell}\pi^\star\|_{\mathrm{F}}^2 \le \frac{F(\pi^\star) - \tau_\lambda(C)}{\lambda_x \mu^X_{\ell+1}} \quad \text{and} \quad \|\pi^\star - \pi^\star P_{Y,h}\|_{\mathrm{F}}^2 \le \frac{F(\pi^\star) - \tau_\lambda(C)}{\lambda_y \mu^Y_{h+1}},$$

where $F(\pi) = \langle \pi, C\rangle + \lambda_x \langle \pi, L_X \pi\rangle + \lambda_y \langle \pi, \pi L_Y\rangle - \lambda H(\pi)$.

Proof. By optimality, $F(\pi^\star) \le F(\pi)$ for any $\pi \in \Pi_{a,b}$. Since $\langle \pi^\star, C\rangle - \lambda H(\pi^\star) \ge \tau_\lambda(C)$, we obtain

$$\lambda_x \langle \pi^\star, L_X \pi^\star\rangle + \lambda_y \langle \pi^\star, \pi^\star L_Y\rangle \le F(\pi) - \tau_\lambda(C).$$

As this is true for any $\pi \in \Pi_{a,b}$, we have

$$\lambda_x \langle \pi^\star, L_X \pi^\star\rangle + \lambda_y \langle \pi^\star, \pi^\star L_Y\rangle \le F(\pi^\star) - \tau_\lambda(C).$$

Now, recall the orthogonal decomposition $L_X = \Phi_X \Lambda_X \Phi_X^\top$ in Theorem 1. Then,

$$\langle \pi^\star, L_X \pi^\star\rangle = \sum_{i=1}^{n} \mu_i^X \|e_i^\top \Phi_X^\top \pi^\star\|_2^2 \ge \mu_{\ell+1}^X \sum_{i=\ell+1}^{n} \|e_i^\top \Phi_X^\top \pi^\star\|_2^2 = \mu_{\ell+1}^X \|\pi^\star - P_{X,\ell}\pi^\star\|_{\mathrm{F}}^2.$$

Hence, the first bound follows, and the second bound follows analogously. □

Another conclusion from Proposition 2 is

$$\|\pi^\star - P_{X,\ell}\pi^\star P_{Y,h}\|_{\mathrm{F}} \le \sqrt{\frac{F(\pi^\star) - \tau_\lambda(C)}{\lambda_x \mu_{\ell+1}^X}} + \sqrt{\frac{F(\pi^\star) - \tau_\lambda(C)}{\lambda_y \mu_{h+1}^Y}},$$

which follows from

$$\pi^\star - P_{X,\ell}\pi^\star P_{Y,h} = (I - P_{X,\ell})\pi^\star + P_{X,\ell}\pi^\star(I - P_{Y,h})$$

and the triangle inequality.

The quantity $F(\pi^\star) - \tau_\lambda(C) \ge 0$ is the gap between the LapOT and entropic-OT optimal values, both computed by the solver. Hence, Proposition 2 thus provides an *a posteriori* certificate: after solving LapOT, one may evaluate the right-hand side directly to verify that $\pi^\star$ concentrates on the low-frequency eigenspaces of L_X and L_Y. The bound is non-vacuous whenever this value gap is small relative to $\lambda_x \mu_{\ell+1}^X$ and $\lambda_y \mu_{h+1}^Y$. Accordingly, Proposition 2 states that when the similarity graphs have a pronounced eigengap after ℓ and h eigenvectors, the optimal coupling is close to a matrix whose rows and columns vary primarily along the corresponding approximate cluster indicators. This provides the theoretical mechanism behind the use of LapOT inside RSC, even when the graphs are connected and the marginals are not block-constant.

To conclude, Theorem 1 should be viewed as the exact disconnected-graph limit, while Proposition 2 explains the connected-graph regime used in the experiments in the following sections. In Section 5.2, we indeed report the upper bound on the right-hand side of Proposition 2 for the stock market data, numerically confirming the decay of the projection error as the dimension of the low-frequency eigenspaces increases.

4 Refined Simultaneous Clustering

This section introduces a new approach for clustering two point clouds simultaneously, which we call Refined Simultaneous Clustering (RSC). We have seen in the previous sections that the optimal coupling of LapOT can capture the cluster structure of the two point clouds. RSC is a practical method that distills the cluster structure from the optimal coupling of LapOT and uses it to perform simultaneous clustering of the two point clouds with the goal of making the clustering of both clouds more aligned with each other.

The full procedure of RSC is summarized in Algorithm 1. After obtaining the optimal coupling $\pi^\star$ from LapOT, we apply k-means to the rows and columns of $\pi^\star$ and define the switch matrices $\pi_{\mathsf{switch}}^X \in \{0,1\}^{n\times n}$ and $\pi_{\mathsf{switch}}^Y \in \{0,1\}^{m\times m}$ as follows:

$$(\pi_{\mathsf{switch}}^X)_{ij} = \begin{cases} 1 & \text{if row } i \text{ and row } j \text{ belong to the same cluster,} \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

$$(\pi_{\mathsf{switch}}^Y)_{ij} = \begin{cases} 1 & \text{if column } i \text{ and column } j \text{ belong to the same cluster,} \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

By applying k-means to the rows (resp. columns) of $\pi^\star$, we cluster together points in X (resp. Y) that have a similar fuzzy matching with Y (resp. X). Then, in Step 4 of Algorithm 1, we refine the original input

Algorithm 1 Refined Simultaneous Clustering

Input: matching cost $C \in \mathbb{R}^{n\times m}$, marginal weights $a \in \Delta_n$ and $b \in \Delta_m$.

Input: similarity matrices $K_X \in \mathbb{R}^{n\times n}$ and $K_Y \in \mathbb{R}^{m\times m}$.

Input: hyperparameters $\lambda_x, \lambda_y, \lambda$ for regularization and $k, k' \in \mathbb{N}$ for clustering.

1: Define $L_X = \mathrm{diag}(K_X 1_n) - K_X$ and $L_Y = \mathrm{diag}(K_Y 1_m) - K_Y$.

2: Find the solution $\pi^\star$ of $\mathsf{LapOT}_{\lambda_x,\lambda_y,\lambda}(a, b, C, L_X, L_Y)$.

3: Apply k-means with k' clusters to the rows, columns of $\pi^\star$ to define π^X_{switch}, π^Y_{switch} per (18), (19).

4: $\tilde{K}_X \leftarrow K_X \odot \pi^X_{\mathsf{switch}}$ and $\tilde{K}_Y \leftarrow K_Y \odot \pi^Y_{\mathsf{switch}}$

5: Update the graph Laplacians: $\tilde{L}_X \leftarrow \mathrm{diag}(\tilde{K}_X 1_n) - \tilde{K}_X$ and $\tilde{L}_Y \leftarrow \mathrm{diag}(\tilde{K}_Y 1_m) - \tilde{K}_Y$

6: Apply spectral clustering with k clusters to $\tilde{L}_X$ and $\tilde{L}_Y$ to obtain the final clusters of X and Y.

Output: Final clusters of X and Y.

similarity matrices K_X and K_Y by multiplying them element-wise with the switch matrices π^X_{switch} and π^Y_{switch}, respectively. Then, we recalculate the refined graph Laplacians $\tilde{L}_X$ and $\tilde{L}_Y$ based on the refined similarity matrices $\tilde{K}_X$ and $\tilde{K}_Y$, followed by the usual spectral clustering procedure to obtain the final clusters of X and Y. Here, the idea is to condition the final clustering procedure on the similarity matrices refined by the switch matrices, expecting the cluster formation in X and Y to reflect the matching relations between the clouds, and hence, to be more aligned with each other.

It is worth noting the similarity between RSC and the standard spectral clustering procedure. Spectral clustering is a two-step procedure that first obtains suitable spectral embeddings via various methods like Laplacian eigenmaps [2] or diffusion maps [8], and then applies k-means to the spectral embeddings to obtain the final clusters of a single point cloud. In RSC, we first obtain the optimal coupling of LapOT, which encodes the cluster-aware matching information between the two point clouds, allowing us to leverage the shared information between the two point clouds in the subsequent clustering step.

Figure 1(b) earlier in the paper shows an example of the RSC method applied to two point clouds X and Y taken from the CAPOD dataset [31]. Here, Algorithm 1 is applied with $k' = 3$ for the switch matrices in Step 3 and $k = 5$ for the final spectral clustering in Step 6. Unlike the independent clustering of X and Y shown in Figure 1(a), the RSC method correctly captures the matching cluster structure between the two clouds, resulting in aligned clusters. The input similarity matrices are computed using the RBF kernel, while the marginal weights are based on the degrees from the similarity matrices as explained below. The cost matrix is computed using the Wasserstein-1 distance between the distance profiles of points in X and Y as described in Section 2.1.

Switch Matrices. The role of the switch matrices π^X_{switch} and π^Y_{switch} is to capture the coarse cluster structure of the two point clouds based on the optimal coupling $\pi^\star$ before moving on to the final clustering step. Hence, the choice of the number of clusters k' for the switch matrices governs the rough partition of the point clouds, distilling the cluster structure of the match encoded by the optimal coupling of LapOT. In practice, we choose k' to be slightly more than half the number of clusters k for the final clustering step, so that the switch matrices capture the coarse cluster structure of the two clouds, while the subsequent clustering step can still produce clusters with enough granularity. Figure 2 visualizes the 3D human shapes in Figure 1 clustered according to the switch matrices π^X_{switch} and π^Y_{switch} with $k' = 3$. We can see that the rough cluster structure of the two clouds is captured by the switch matrices, providing a good starting point for the final clustering step.

Degree-Based Marginals. While it is common to set a, b to be uniform distributions in practice, we found that using degree-based marginals can improve the performance of RSC. The idea is to assign higher weights to points that are more significant in the similarity graph, which can help to capture the cluster structure of the point clouds more effectively. Precisely, we define the degree-based marginals as follows:

$$a_i = \frac{\deg(X_i)}{\sum_{i'=1}^n \deg(X_{i'})} \quad \text{and} \quad b_j = \frac{\deg(Y_j)}{\sum_{j'=1}^m \deg(Y_{j'})},$$

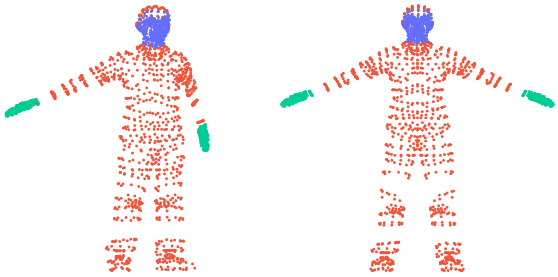
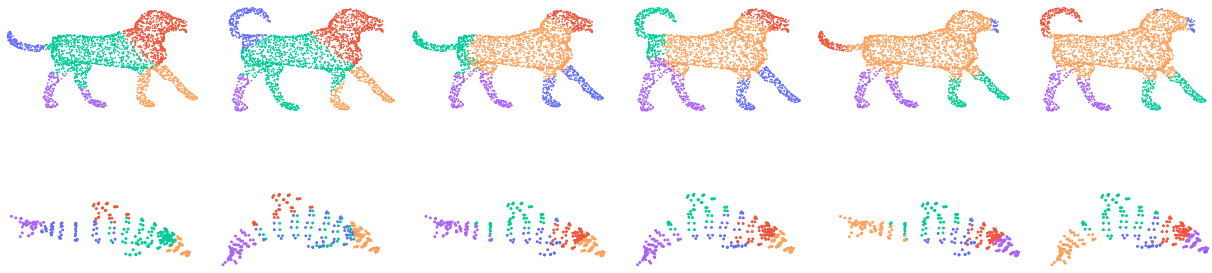


Figure 2: Clustering of two 3D human shapes according to the clustering structure encoded in the switch matrices π^X_{switch} and π^Y_{switch}.

where $\deg(X_i) = \sum_{i'=1}^{n}(K_X)_{ii'}$ and $\deg(Y_j) = \sum_{j'=1}^{m}(K_Y)_{jj'}$ are the degrees of points X_i and Y_j in the similarity graphs, respectively. In this case, the distance profiles to define the cost matrix C as described in Section 2.1 are also based on the degree-based marginals.

Discussion. Figure 3 shows additional examples of RSC applied to two point clouds from the CAPOD dataset. The first row shows two 3D dog shapes, while the second row shows two 3D dolphin shapes. In both cases, RSC produces more consistent clusters across the two point clouds compared to independent clustering. On top of these, we also experimented with the low-rank version of LapOT in Step 2 of Algorithm 1 with $r = 10$, which also yields consistent clusters across the two point clouds. Overall, these results demonstrate the effectiveness of RSC in capturing the cluster structure of two point clouds simultaneously, leveraging the shared information between them to produce more aligned clusters. Of course, like any clustering method, RSC is not guaranteed to produce perfectly consistent clusters across the two point clouds and depends on the choice of hyperparameters and similarity matrices. We leave a more detailed study of the hyperparameter selection for RSC to future work.



(a) Independent Clustering (b) RSC (c) RSC (Low-Rank)

Figure 3: Other examples of RSC applied to two point clouds from the CAPOD dataset. The first row shows two 3D dog shapes, while the second row shows two 3D dolphin shapes. (c) shows the results of RSC with the low-rank version of LapOT in Step 2 of Algorithm 1 with $r = 10$.

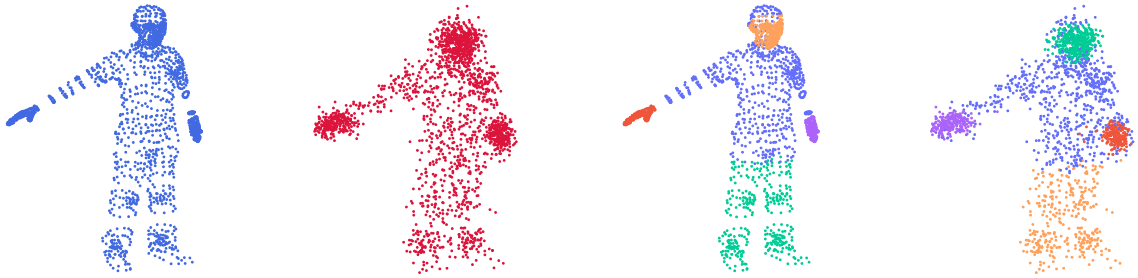
5 Applications

5.1 Alignment via Refined Simultaneous Clustering

Finding a rigid transformation that best aligns two point clouds is a fundamental problem in many applications, including computer vision, robotics, and 3D modeling. Once we establish a correspondence between

the two clouds, we can formulate an orthogonal Procrustes problem to estimate the underlying rigid transformation. In this section, we demonstrate how this can be done by applying RSC.

We pick a 3D human shape from the CAPOD dataset [31] as our point cloud X and generate a second point cloud Y by applying a random rotation to X and adding isotropic Gaussian noise $N(0, \sigma^2 I_3)$, which is shown in Figure 4(a). We begin by applying RSC, following the same hyperparameter selection procedure used to produce Figure 1(b) as described in Section 4; the result is shown in Figure 4(b). Then, we calculate the centroids of each cluster in X and Y and assign to each centroid a weight proportional to the sum of degrees of the points in its cluster. Now, we consider matching between the centroids of the clusters in X and Y following the usual distance profile matching (3) in Section 2.1. Then, for $k = 5$ matched cluster pairs, we perform the pointwise matching between the points in the matched clusters following the distance profile matching again, which can be done in parallel for each matched cluster pair. Finally, based on the established correspondence between the two clouds, we estimate the underlying rotation by solving the orthogonal Procrustes problem.



(a) Point Clouds (b) RSC

Figure 4: Point clouds and the result of RSC. (a) shows the point cloud of a human (left) and the same point cloud rotated and with noise added (right). (b) shows the result of RSC applied to the clouds in (a) with $k = 5$ clusters. For presentation purposes, the clouds are rotated to share the same viewing angle.

The main contribution here is to avoid global matching between the two clouds, which can be computationally expensive and fail to capture the local structure of the clouds. Instead, we simultaneously cluster the clouds in a consistent manner via RSC and establish a correspondence between the matched clusters, thereby breaking the global matching problem into smaller subproblems that can be solved simultaneously.

For a varying level of noise σ^2, we report the relative error in spectral norm between the estimated and true underlying rotation. We compare the proposed method with two alternative approaches: (i) an ICP [4] pipeline starting with fast point feature histograms [36] and RANSAC algorithms for a coarse global correspondence estimate, followed by the generalized ICP algorithm [41], and (ii) the global Distance Profile Matching (DPM) between clouds X and Y as formulated in [20], followed by the usual orthogonal Procrustes problem. Here, the noise level is converted to a signal-to-noise ratio (SNR) in decibels, defined as $\text{SNR}_{\text{db}} = 10 \log_{10}(\sigma_X^2/\sigma^2)$, where σ_X^2 is the variance of the point cloud X and σ^2 is the variance of the added noise. We repeat the comparison test over ten independent runs and report the mean relative error in Table 1. First, we note that for a low to moderate presence of noise, our approach and the global DPM approach perform remarkably well, while the ICP pipeline quickly deteriorates as the noise level increases. For higher noise levels, our method is more robust than the global DPM approach. When using global DPM in a high-noise setup, there are no restrictions on the match, and a point can potentially be matched with every other point in the opposite cloud, which can lead to a less accurate match. In contrast, our method restricts the match to clusters that are more likely to be similar, thereby yielding a more accurate match.

5.2 Non-Euclidean Application: RSC on High-Dimensional Stock Market Data

So far, we have focused on point clouds under the Euclidean metric to define the distance profiles. However, as profiles can be defined under any suitable similarity measure, we can also apply RSC beyond the usual Euclidean setting. In this section, we demonstrate the application of RSC to high-dimensional stock market data, where the similarity measure is based on the correlation between the returns of stocks.

Methods \ $\mathbf{SNR_{db}}$	**24.91**	**18.89**	**15.37**	**10.90**	**4.92**	**2.91**
ICP Pipeline	0.03285	0.37441	1.15744	1.67079	2.24804	2.51093
Global DPM	0.00288	0.00722	0.01460	0.02889	0.12395	0.25521
Our Method	0.00668	0.02421	0.03134	0.09283	0.10219	0.20242

Table 1: Mean spectral-norm relative error between the estimated and true underlying rotations over ten independent runs, for varying noise levels σ^2 expressed in $\mathrm{SNR}_{\mathrm{db}}$.

Here, our goal is to cluster the top 50 companies from S&P 500 (USA) and the top 50 companies from the Japanese stock market simultaneously, in order to analyze the similarity relations in the stock market between the two countries. For each company, we consider the daily closing price of its stock over a period of 5 years starting from the first day of 2020, which gives us a time series of length $T = 1256$ (the number of trading days in 5 years), say, $(P(1), \ldots, P(T))$. We normalize the time series so that $P(1) = 1$ for all stocks. We do that to neutralize scaling problems that can occur from differences in currencies (US dollars to Japanese yen) and different starting points in value. We then compute the daily returns of each stock as the percentage change in price from the previous day, namely, $R(t) = (P(t) - P(t-1))/P(t-1)$ for $t = 2, \ldots, T$ and $R(1) = 0$. We take $(R(1), R(2), \ldots, R(T)) \in \mathbb{R}^T$ as the feature vector for each stock, which gives us a point cloud of size $n = 50$ in $\mathbb{R}^T$ for both the S&P 500 and Japanese companies.

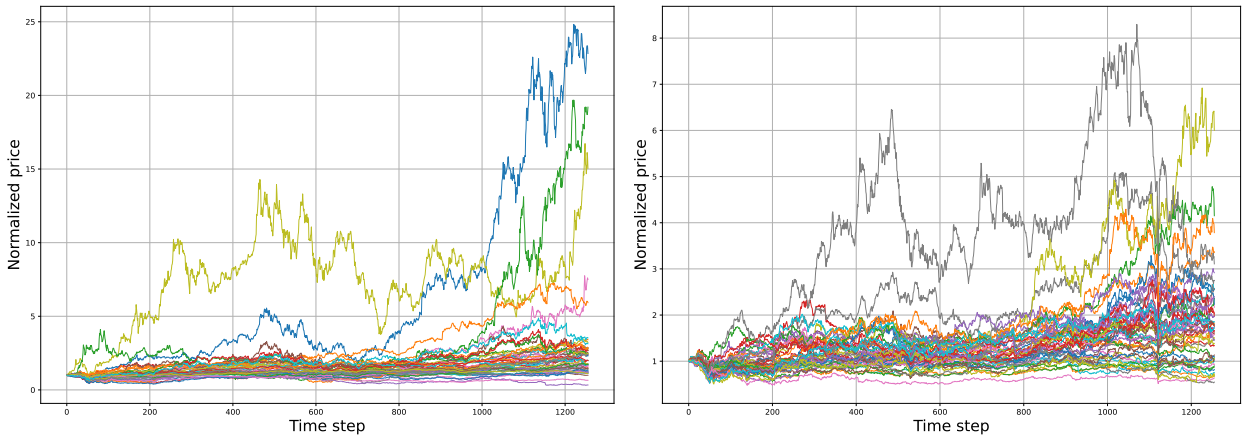


Figure 5: Normalized daily stock prices for the top 50 companies from the S&P 500 (left) and top 50 Japanese companies by market cap (right) over five years from 2020.

Let $R_i = (R_i(1), \ldots, R_i(T))$ for $i = 1, \ldots, n$ be the returns of the n companies in S&P 500. For $i, i' = 1, \ldots, n$, we define the covariance between the returns of companies i and i' as

$$\mathrm{Cov}(R_i, R_{i'}) = \frac{1}{T-1} \sum_{t=1}^{T} (R_i(t) - \bar{R}_i)(R_{i'}(t) - \bar{R}_{i'}),$$

where $\bar{R}_i = \frac{1}{T} \sum_{t=1}^{T} R_i(t)$ is the mean return of company i. Then, the correlation between the returns of companies i and i' is defined as $\rho_{ii'} := \frac{\mathrm{Cov}(R_i, R_{i'})}{\sqrt{\mathrm{Var}(R_i)\mathrm{Var}(R_{i'})}}$, where $\mathrm{Var}(R_i) = \mathrm{Cov}(R_i, R_i)$ is the variance of the returns of company i. Finally, we define the similarity $d_{ii'}$ between companies i and i' as $d_{ii'} = \sqrt{2(1 - \rho_{ii'})}$, which we treat as a distance measure between the two companies. Note that when R_i and $R_{i'}$ are positively correlated, $d_{ii'}$ is small. We define the similarity in the same way for the Japanese companies. This distance is then transformed into a similarity score using an RBF kernel as before, from which the similarity matrices K_X and K_Y are constructed.

Since the correlation-based RBF graphs in this example are connected, the exact block-constancy guarantee of Theorem 1 is best interpreted as an idealized limiting case. The relevant mechanism here is Propo-

sition 2: LapOT suppresses oscillatory components of the coupling over the two market similarity graphs, so broad sector-level structure can appear as an approximately low-rank or block-structured coupling.

Unlike the previous examples, we construct the cost matrix C for LapOT using different similarity profiles than the ones used to construct the similarity matrices K_X and K_Y. This is because we want to capture different financial characteristics of the stocks in the two countries. Specifically, we take the beta of each stock as a measure of its systematic risk, a measure of a stock's volatility in relation to the overall market. The beta of stock i is $\beta_i = \mathrm{Cov}(R_i, R_{\text{market}})/\mathrm{Var}(R_{\text{market}})$, where R_{market} is the return of the market index. The cost C_{ij} is then set to the Wasserstein-1 distance between the similarity profiles that are picked so that $W_1(\mu_i, \nu_j)$ is an upper bound on $|\beta_i - \beta_j|$. This choice of cost encourages matching stocks with similar systematic risk profiles. The formulation of the specific choice for μ_i and ν_j can be found in Section B in the appendix.

With this setup, we expect LapOT to produce a coupling that aligns companies with similar sector-based

Figure 6: The solution of LapOT applied to the US-Japan stock market data. We permuted the rows and columns of the optimal coupling to better visualize its low-rank structure.

behaviors (captured by L_X, L_Y) and similar market volatility profiles (captured by C). In Figure 6, we show the low-rank structure of the optimal coupling solution to (6), indicating broad cluster similarities between the two countries. To complement the low-rank structure observed in Figure 6, we numerically evaluate the upper bound in Proposition 2 for increasing values of ℓ and h. Figure 7 provides direct numerical support for the proposition: the theoretical bound uniformly controls the projection error and captures its decay as the dimensions of the retained low-frequency eigenspaces increase.

In the appendix, Figures 8 and 9 show the results of applying RSC to the companies from the USA and Japan, respectively, showing a consistent pattern in the results.

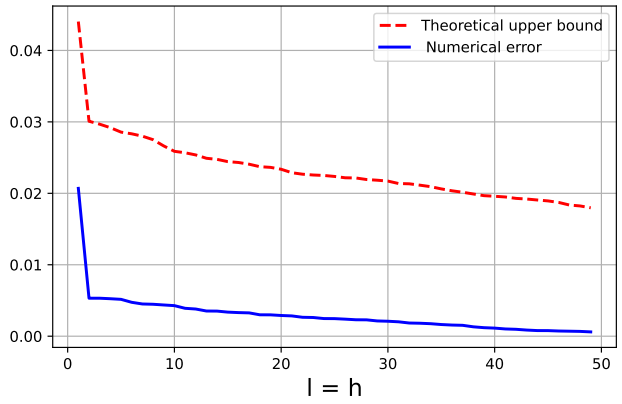


Figure 7: Comparison between the theoretical upper bound and the projection error $\|\pi^\star - P_{X,\ell}\pi^\star P_{Y,h}\|_{\mathrm{F}}$ of the optimal coupling $\pi^\star$, evaluated for $\ell = h = 1, \ldots, n-1$.

References

[1] David Alvarez-Melis and Tommi S Jaakkola. Gromov-Wasserstein alignment of word embedding spaces. *arXiv preprint arXiv:1809.00013*, 2018.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, 2001.

[3] Shai Ben-David, Ulrike Von Luxburg, and Dávid Pál. A sober look at clustering stability. In *Conference on Learning Theory*, 2006.

[4] Paul J Besl and Neil D McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] Andrew J Blumberg, Mathieu Carriere, Michael A Mandell, Raul Rabadan, and Soledad Villar. MREC: A fast and versatile framework for aligning and matching point clouds with applications to single cell molecular data. *arXiv preprint arXiv:2001.01666*, 2020.

[6] Richard A Brealey, Stewart C Myers, and Franklin Allen. *Principles of Corporate Finance*. McGraw Hill, 2011.

[7] Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning generative models across incomparable spaces. In *International Conference on Machine Learning*, 2019.

[8] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[9] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[10] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

[11] Jian Ding, Zongming Ma, Yihong Wu, and Jiaming Xu. Efficient random graph matching via degree profiles. *Probability Theory and Related Fields*, 179(1):29–115, 2021.

[12] Théo Dumont, Théo Lacombe, and François-Xavier Vialard. On the existence of Monge maps for the Gromov-Wasserstein problem. *Foundations of Computational Mathematics*, 25(2):463–510, 2025.

[13] Richard L Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

[14] Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

[15] Aden Forrow, Jan-Christian Hütter, Mor Nitzan, Philippe Rigollet, Geoffrey Schiebinger, and Jonathan Weed. Statistical optimal transport via factored couplings. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[16] Edouard Grave, Armand Joulin, and Quentin Berthet. Unsupervised alignment of embeddings with Wasserstein procrustes. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[17] Joeri Hofmans, Eva Ceulemans, Douglas Steinley, and Iven Van Mechelen. On the added value of bootstrap analysis for K-means clustering. *Journal of Classification*, 32(2):268–284, 2015.

[18] Hanwen Huang, Yufeng Liu, Ming Yuan, and JS Marron. Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, 24(4):975–993, 2015.

[19] YoonHaeng Hur, Wenxuan Guo, and Tengyuan Liang. Reversible Gromov–Monge sampler for simulation-based inference. *SIAM Journal on Mathematics of Data Science*, 6(2):283–310, 2024.

[20] YoonHaeng Hur and Yuehaw Khoo. Robust point matching with distance profiles. *Journal of Machine Learning Research*, 26(205):1–38, 2025.

[21] YoonHaeng Hur and Tengyuan Liang. A convexified matching approach to imputation and individualized inference. *arXiv preprint arXiv:2407.05372*, 2024.

[22] YoonHaeng Hur, Anirban Nath, and Genevera Allen. Inference for clustering: Conformal sets for cluster labels. *arXiv preprint arXiv:2604.03488*, 2026.

[23] M. K. Kerr and G. A. Churchill. Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *Proceedings of the National Academy of Sciences*, 98(16):8961–8965, 2001.

[24] Tjalling C Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica*, pages 53–76, 1957.

[25] Chuen-Ming Liu, Zhi-Ping Niu, and Kuan-Ting Liao. Mechanisms to improve clustering uncertain data with UKmeans. *Data & Knowledge Engineering*, 116:1–18, 2018.

[26] Eliane Maria Loiola, Nair Maria Maia De Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.

[27] Facundo Mémoli. Gromov-Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011.

[28] Facundo Mémoli and Tom Needham. Comparison results for Gromov-Wasserstein and Gromov-Monge distances. *ESAIM: Control, Optimisation and Calculus of Variations*, 30:78, 2024.

[29] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[30] Anirban Nath, YoonHaeng Hur, and Genevera I Allen. Weighted conformal clustering. *arXiv preprint arXiv:2606.00436*, 2026.

[31] Panagiotis Papadakis. The canonically posed 3d objects dataset. In *Eurographics Workshop on 3D Object Retrieval*, pages 33–36, 2014.

[32] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[33] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, 2016.

[34] Alexander Rakhlin and Andrea Caponnetto. Stability k-means clustering. In *Advances in Neural Information Processing Systems*, 2006.

[35] Aryan Tajmir Riahi, Geoffrey Woollard, Frédéric Poitevin, Anne Condon, and Khanh Dao Duc. AlignOT: An optimal transport based algorithm for fast 3d alignment with applications to cryogenic electron microscopy density maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(6):3842–3850, 2023.

[36] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3d registration. *IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[37] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020.

[38] Meyer Scetbon and Marco Cuturi. Low-rank optimal transport: Approximation, statistics and debiasing. In *Advances in Neural Information Processing Systems*, 2022.

[39] Meyer Scetbon, Marco Cuturi, and Gabriel Peyré. Low-rank Sinkhorn factorization. In *International Conference on Machine Learning*, 2021.

[40] Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

[41] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: Science and Systems*. The MIT Press, 2010.

[42] Yasin Senbabaoğlu, George Michailidis, and Jun Z Li. Critical limitations of consensus clustering in class discovery. *Scientific Reports*, 4(1):6207, 2014.

[43] Amit Singer and Ruiyi Yang. Alignment of density maps in Wasserstein distance. *Biological Imaging*, 4:e5, 2024.

[44] Kyle S Smith, Yiran Li, Parthiv Haldipur, Brian L Gudenas, Kathleen J Millen, Volker Hovestadt, and Paul A Northcott. Lack of evidence for the transitional cerebellar progenitor. *Nature*, 643(8071):E1–E8, 2025.

[45] Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics*, 35(4):1–13, 2016.

[46] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[47] Cédric Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2003.

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

[49] Ulrike Von Luxburg. Clustering stability: An overview. *Foundations and Trends® in Machine Learning*, 2(3):235–274, 2010.

[50] Sara Wade and Zoubin Ghahramani. Bayesian cluster analysis: Point estimation and credible balls (with discussion). *Bayesian Analysis*, 13(2):559–626, 2018.

[51] Alan Geoffrey Wilson. The use of entropy maximising models, in the theory of trip distribution, mode split and route split. *Journal of Transport Economics and Policy*, pages 108–126, 1969.

[52] Yang Xiao, Wang Lu, Jie Ji, Ruimeng Ye, Gen Li, Xiaolong Ma, and Bo Hui. Optimal transport for brain-image alignment: Unveiling redundancy and synergy in neural information processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20445–20455, 2025.

[53] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-Wasserstein learning for graph matching and node embedding. In *International Conference on Machine Learning*, 2019.

A Optimization of Low-Rank LapOT

This section discusses the optimization scheme for solving the low-rank version of the LapOT problem (10). As noticed in [39], the division by g in the objective function of (10) can lead to numerical instability. To mitigate this issue, [39] suggests restricting the search domain to

$$\mathcal{C}(a,b,r,\alpha) := \{(U,V,g) \in \mathcal{C}(a,b,r) : g \geq \alpha 1_r\},$$

where $\alpha > 0$ is a user-defined stability parameter. This restriction prevents any entry of g from being too close to zero. Hence, a stable formulation of the low-rank LapOT problem (10) is given by

$$\min_{(U,V,g)\in\mathcal{C}(a,b,r,\alpha)} f(U,V,g), \tag{20}$$

where $f(U,V,g)$ is the objective function in (10), namely,

$$\begin{aligned} f(U,V,g) = &\langle U\mathrm{diag}(1_r/g)V^\top, C\rangle \\ &+ \lambda_x\langle U\mathrm{diag}(1_r/g)V^\top, L_X U\mathrm{diag}(1_r/g)V^\top\rangle + \lambda_y\langle U\mathrm{diag}(1_r/g)V^\top, U\mathrm{diag}(1_r/g)V^\top L_Y\rangle \\ &- \lambda H(U) - \lambda H(V) - \lambda H(g). \end{aligned}$$

Algorithm 3 presented at the end of this section summarizes the optimization scheme for solving (20).

To understand Algorithm 3, let us start by following [39] to rewrite the feasible set $\mathcal{C}(a,b,r,\alpha)$ as the intersection of two sets $\mathcal{C}_1(a,b,r,\alpha)$ and $\mathcal{C}_2(r)$, where

$$\begin{aligned} \mathcal{C}_1(a,b,r,\alpha) &\triangleq \{(U,V,g) \in \mathbb{R}_+^{n\times r} \times \mathbb{R}_+^{m\times r} \times \mathbb{R}_+^r : U1_r = a, V1_r = b, g \geq \alpha 1_r\}, \\ \mathcal{C}_2(r) &\triangleq \{(U,V,g) \in \mathbb{R}_+^{n\times r} \times \mathbb{R}_+^{m\times r} \times \mathbb{R}_+^r : U^\top 1_n = V^\top 1_m = g\}, \end{aligned}$$

which allows us to apply Dykstra's algorithm for projection onto the intersection of convex sets. As in [39], we implement mirror descent under the KL divergence to solve (20). More precisely, for $x_k \triangleq (U_k, V_k, g_k)$ and step size $\gamma_k > 0$ at iteration k, we update x_{k+1} by solving the following optimization problem:

$$x_{k+1} = \operatorname*{arg\,min}_{x\in\mathcal{C}(a,b,r,\alpha)} \langle \nabla f(x_k), x\rangle + \frac{1}{\gamma_k}\mathsf{KL}(x \,\|\, x_k),$$

which is equivalent to

$$x_{k+1} = \operatorname*{arg\,min}_{x\in\mathcal{C}(a,b,r,\alpha)} \mathsf{KL}\left(x \,\|\, x_k \odot \exp(-\gamma_k \nabla f(x_k))\right), \tag{21}$$

where $\odot$ denotes the element-wise product, and the exponential is taken element-wise. Here, the gradient $\nabla f(U,V,g)$ is given by

$$\begin{aligned} \nabla_U f &= MV\mathrm{diag}(1_r/g) + \lambda\log(U), \\ \nabla_V f &= M^\top U\mathrm{diag}(1_r/g) + \lambda\log(V), \\ \nabla_g f &= -\frac{\mathrm{diag}(U^\top MV)}{g^2} + \lambda\log(g), \end{aligned}$$

where g^2 is the vector obtained by squaring each entry of g, and M is given by

$$M = 2\lambda_x L_X U \mathrm{diag}(1_r/g) V^\top + 2\lambda_y U \mathrm{diag}(1_r/g) V^\top L_Y + C.$$

The projection onto $\mathcal{C}(a,b,r,\alpha)$ under the KL divergence in (21) can be efficiently computed using Dykstra's algorithm, which alternates between projecting onto $\mathcal{C}_1(a,b,r,\alpha)$ and $\mathcal{C}_2(r)$. To see this, define

$$\begin{aligned}
\tilde{U}_k &\triangleq U_k \odot \exp(-\gamma_k \nabla_U f(U_k, V_k, g_k)) = U_k \odot \exp\left(-\gamma_k \left(M_k V_k \mathrm{diag}(1_r/g_k) + \lambda \log(U_k)\right)\right), \\
\tilde{V}_k &\triangleq V_k \odot \exp(-\gamma_k \nabla_V f(U_k, V_k, g_k)) = V_k \odot \exp\left(-\gamma_k \left(M_k^\top U_k \mathrm{diag}(1_r/g_k) + \lambda \log(V_k)\right)\right), \\
\tilde{g}_k &\triangleq g_k \odot \exp(-\gamma_k \nabla_g f(U_k, V_k, g_k)) = g_k \odot \exp\left(-\gamma_k \left(-\frac{\mathrm{diag}(U_k^\top M_k V_k)}{g_k^2} + \lambda \log(g_k)\right)\right),
\end{aligned}$$

where M_k is defined as

$$M_k = 2\lambda_x L_X U_k \mathrm{diag}(1_r/g_k) V_k^\top + 2\lambda_y U_k \mathrm{diag}(1_r/g_k) V_k^\top L_Y + C.$$

Hence, the mirror descent update in (21) can be rewritten as

$$(U_{k+1}, V_{k+1}, g_{k+1}) = \underset{(U,V,g)\in\mathcal{C}(a,b,r,\alpha)}{\arg\min} \mathrm{KL}\left((U,V,g) \,\|\, (\tilde{U}_k, \tilde{V}_k, \tilde{g}_k)\right). \tag{22}$$

Dykstra's Algorithm. Changing the variable names to $(\xi^{(1)}, \xi^{(2)}, \xi^{(3)}) \in \mathbb{R}_+^{n\times r} \times \mathbb{R}_+^{m\times r} \times \mathbb{R}_+^r$ in (22), we now derive the Dykstra's algorithm to solve the following:

$$\underset{(U,V,g)\in\mathcal{C}(a,b,r,\alpha)}{\arg\min} \mathrm{KL}\left((U,V,g) \,\|\, (\xi^{(1)}, \xi^{(2)}, \xi^{(3)})\right). \tag{23}$$

Algorithm 2 summarizes Dykstra's algorithm for solving (23), where $r(\cdot)$ and $c(\cdot)$ denote the row and column sum operators, respectively. To see this, recall that Dykstra's algorithm [13] iterates as follows:

$$\begin{aligned}
(\xi^{(1)}_{2k+1}, \xi^{(2)}_{2k+1}, \xi^{(3)}_{2k+1}) &= \underset{(U,V,g)\in\mathcal{C}_1(a,b,r,\alpha)}{\arg\min} \mathrm{KL}\left((U,V,g) \,\|\, (\xi^{(1)}_{2k} \odot q^{(1)}_{2k-1}, \xi^{(2)}_{2k} \odot q^{(2)}_{2k-1}, \xi^{(3)}_{2k} \odot q^{(3)}_{2k-1})\right), \\
(q^{(1)}_{2k+1}, q^{(2)}_{2k+1}, q^{(3)}_{2k+1}) &= \left(\frac{\xi^{(1)}_{2k} \odot q^{(1)}_{2k-1}}{\xi^{(1)}_{2k+1}}, \frac{\xi^{(2)}_{2k} \odot q^{(2)}_{2k-1}}{\xi^{(2)}_{2k+1}}, \frac{\xi^{(3)}_{2k} \odot q^{(3)}_{2k-1}}{\xi^{(3)}_{2k+1}}\right), \\
(\xi^{(1)}_{2k+2}, \xi^{(2)}_{2k+2}, \xi^{(3)}_{2k+2}) &= \underset{(U,V,g)\in\mathcal{C}_2(r)}{\arg\min} \mathrm{KL}\left((U,V,g) \,\|\, (\xi^{(1)}_{2k+1} \odot q^{(1)}_{2k}, \xi^{(2)}_{2k+1} \odot q^{(2)}_{2k}, \xi^{(3)}_{2k+1} \odot q^{(3)}_{2k})\right), \\
(q^{(1)}_{2k+2}, q^{(2)}_{2k+2}, q^{(3)}_{2k+2}) &= \left(\frac{\xi^{(1)}_{2k+1} \odot q^{(1)}_{2k}}{\xi^{(1)}_{2k+2}}, \frac{\xi^{(2)}_{2k+1} \odot q^{(2)}_{2k}}{\xi^{(2)}_{2k+2}}, \frac{\xi^{(3)}_{2k+1} \odot q^{(3)}_{2k}}{\xi^{(3)}_{2k+2}}\right).
\end{aligned}$$

Propositions 2 and 3 of [39] state the closed-form solutions for the projections onto $\mathcal{C}_1(a,b,r,\alpha)$ and $\mathcal{C}_2(r)$, where (24) and (25) provide the respective formulas. Algorithm 2 iterates these projections until the row sums of $\xi^{(1)}_{2k+2}$ and $\xi^{(2)}_{2k+2}$ are sufficiently close to a and b, respectively.

Adaptive Step Size. We use the adaptive step size scheme proposed by [38] to avoid overflowing in the exponent terms used as input for Dykstra's algorithm. Hence, when solving (21) at iteration k, we set the step size γ_k to be

$$\gamma_k = \frac{\gamma}{\|\nabla f(x_k)\|_\infty^2},$$

where $\gamma > 0$ is a user-defined parameter. Following the recommendation of [38], we set $\gamma \in [1, 10]$ for the initialization.

Algorithm 2 $\mathsf{Dykstra}(\xi^{(1)}, \xi^{(2)}, \xi^{(3)}, a, b, \alpha, \delta)$

Input: $\xi^{(1)} \in \mathbb{R}_+^{n\times r}$, $\xi^{(2)} \in \mathbb{R}_+^{m\times r}$, $\xi^{(3)} \in \mathbb{R}_+^{r}$ with positive entries, marginal weights $a \in \Delta_n$ and $b \in \Delta_m$.
Input: stability parameter $\alpha > 0$, desired accuracy $\delta > 0$.

Initialize $(\xi_0^{(1)}, \xi_0^{(2)}, \xi_0^{(3)}) = (\xi^{(1)}, \xi^{(2)}, \xi^{(3)})$ and $(q_{-1}^{(1)}, q_{-1}^{(2)}, q_{-1}^{(3)}) = (q_0^{(1)}, q_0^{(2)}, q_0^{(3)}) = (1_{n\times r}, 1_{m\times r}, 1_r)$.
Set $k = 0$.
while $||\xi_{2k}^{(1)} 1_r - a||_1 + ||\xi_{2k}^{(2)} 1_r - b||_1 \geq \delta$ **do**
    Update

$$\begin{aligned}
\xi_{2k+1}^{(1)} &= \operatorname{diag}\left(\frac{a}{r(\xi_{2k}^{(1)} \odot q_{2k-1}^{(1)})}\right)(\xi_{2k}^{(1)} \odot q_{2k-1}^{(1)}),\\
\xi_{2k+1}^{(2)} &= \operatorname{diag}\left(\frac{b}{r(\xi_{2k}^{(2)} \odot q_{2k-1}^{(2)})}\right)(\xi_{2k}^{(2)} \odot q_{2k-1}^{(2)}),\\
\xi_{2k+1}^{(3)} &= \max\left(\xi_{2k}^{(3)} \odot q_{2k-1}^{(3)}, \alpha 1_r\right).
\end{aligned} \tag{24}$$

    Update

$$\begin{aligned}
\xi_{2k+2}^{(3)} &= \left(\xi_{2k+1}^{(3)} \odot q_{2k}^{(3)} \odot c(\xi_{2k+1}^{(1)} \odot q_{2k}^{(1)}) \odot c(\xi_{2k+1}^{(2)} \odot q_{2k}^{(2)})\right)^{1/3},\\
\xi_{2k+2}^{(1)} &= (\xi_{2k+1}^{(1)} \odot q_{2k}^{(1)}) \odot \operatorname{diag}\left(\frac{\xi_{2k+2}^{(3)}}{c(\xi_{2k+1}^{(1)} \odot q_{2k}^{(1)})}\right),\\
\xi_{2k+2}^{(2)} &= (\xi_{2k+1}^{(2)} \odot q_{2k}^{(2)}) \odot \operatorname{diag}\left(\frac{\xi_{2k+2}^{(3)}}{c(\xi_{2k+1}^{(2)} \odot q_{2k}^{(2)})}\right).
\end{aligned} \tag{25}$$

    Update $k \leftarrow k + 1$.
end while

Output: $(\xi_{2k}^{(1)}, \xi_{2k}^{(2)}, \xi_{2k}^{(3)})$.

Algorithm 3 Low-Rank LapOT

Input: matching cost $C \in \mathbb{R}^{n\times m}$, marginal weights $a \in \Delta_n$ and $b \in \Delta_m$.
Input: Graph Laplacians $L_X \in \mathbb{R}^{n\times n}$ and $L_Y \in \mathbb{R}^{m\times m}$, hyperparameters $\lambda_x, \lambda_y, \lambda$.
Input: rank $r \in \mathbb{N}$, stability parameter $\alpha > 0$, desired accuracy $\delta > 0$ for Dykstra's algorithm.
Input: number of iterations $T \in \mathbb{N}$.

Initialize $(U_0, V_0, g_0) \in \mathbb{R}_+^{n\times r} \times \mathbb{R}_+^{m\times r} \times \mathbb{R}_+^{r}$ with positive entries.
for $k = 0, \ldots, T-1$ **do**
    Set the step size $\gamma_k = \frac{\gamma}{\|\nabla f(U_k, V_k, g_k)\|_\infty^2}$.
    Compute

$$M_k = 2\lambda_x L_X U_k \operatorname{diag}(1_r/g_k) V_k^\top + 2\lambda_y U_k \operatorname{diag}(1_r/g_k) V_k^\top L_Y + C.$$

    Compute

$$\begin{aligned}
\tilde{U}_k &= U_k \odot \exp\left(-\gamma_k \left(M_k V_k \operatorname{diag}(1_r/g_k) + \lambda \log(U_k)\right)\right),\\
\tilde{V}_k &= V_k \odot \exp\left(-\gamma_k \left(M_k^\top U_k \operatorname{diag}(1_r/g_k) + \lambda \log(V_k)\right)\right),\\
\tilde{g}_k &= g_k \odot \exp\left(-\gamma_k \left(-\frac{\operatorname{diag}(U_k^\top M_k V_k)}{g_k^2} + \lambda \log(g_k)\right)\right).
\end{aligned}$$

    Update $(U_{k+1}, V_{k+1}, g_{k+1}) = \mathsf{Dykstra}(\tilde{U}_k, \tilde{V}_k, \tilde{g}_k, a, b, \alpha, \delta)$.
end for
    return (U_T, V_T, g_T).

B Further Details of the Stock Market Application

Similarity Profiles. For the data introduced in Section 5.2, between every two stocks i and j, we set the similarity between them to be

$$\bar{\rho}_{ij} = \frac{\mathrm{Cov}(R_i, \frac{P_j}{P_{\text{market}}} R_j)}{\mathrm{Var}(R_{\text{market}})},$$

where P_{market} is the price of the market index, and R_{market} is the return of the market index. The distributions induced by these similarity profiles are then $\mu_i = \frac{1}{n}\sum_{k=1}^{n} \delta_{\bar{\rho}_{ik}}$ and $\nu_j = \frac{1}{n}\sum_{k=1}^{n} \delta_{\bar{\rho'}_{jk}}$. Now, let $C_{ij} = W_1(\mu_i, \nu_j)$, and assume $\bar{\rho}_{i1} \leq \bar{\rho}_{i2} \leq \cdots \leq \bar{\rho}_{in}$ and $\bar{\rho'}_{j1} \leq \bar{\rho'}_{j2} \leq \cdots \leq \bar{\rho'}_{jn}$, without loss of generality. Then, we have

$$\begin{aligned}
W_1(\mu_i, \nu_j) &= \frac{1}{n}\sum_{k=1}^{n} |\bar{\rho}_{ik} - \bar{\rho'}_{jk}| \\
&\geq \frac{1}{n} |\sum_{k=1}^{n} \bar{\rho}_{ik} - \bar{\rho'}_{jk}| \\
&= |\frac{1}{n}\sum_{k=1}^{n} \frac{\mathrm{Cov}(R_i, \frac{P_k}{P_{\text{market}}} R_k)}{\mathrm{Var}(R_{\text{market}})} - \frac{1}{n}\sum_{k=1}^{n} \frac{\mathrm{Cov}(R'_j, \frac{P'_k}{P'_{\text{market}}} R'_k)}{\mathrm{Var}(R'_{\text{market}})}| \\
&= |\frac{\mathrm{Cov}(R_i, R_{\text{market}})}{\mathrm{Var}(R_{\text{market}})} - \frac{\mathrm{Cov}(R'_j, R'_{\text{market}})}{\mathrm{Var}(R'_{\text{market}})}| \\
&= |\beta_i - \beta'_j|,
\end{aligned}$$

which, in practice, appears to be a pretty tight upper bound. This means that by choosing such C, we connect the cost of transportation between the American stock i and the Japanese stock j with the difference between their betas [6].

Additional Results of RSC Here, we provide visualizations of the results of using our method on financial data. In Figures 8 and 9, we show the cluster structure obtained by using RSC on the data introduced in Section 5.2.

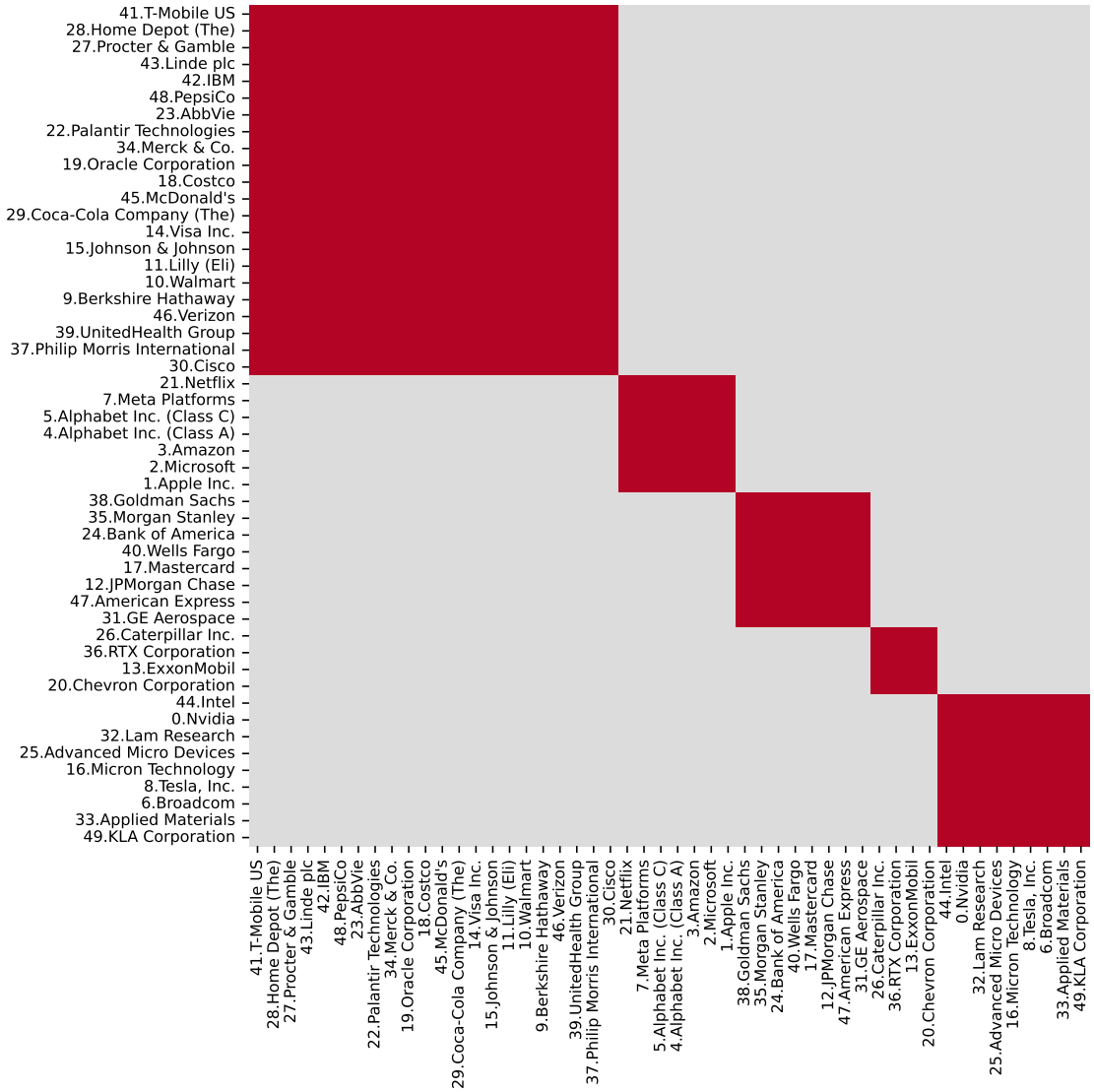


Figure 8: The cluster structure of the top 50 stocks from S&P 500 found by the RSC. We see the rough partition into {**big tech, semiconductor & deep tech infrastructure, finance, energy, pharma & retail & the rest**}.

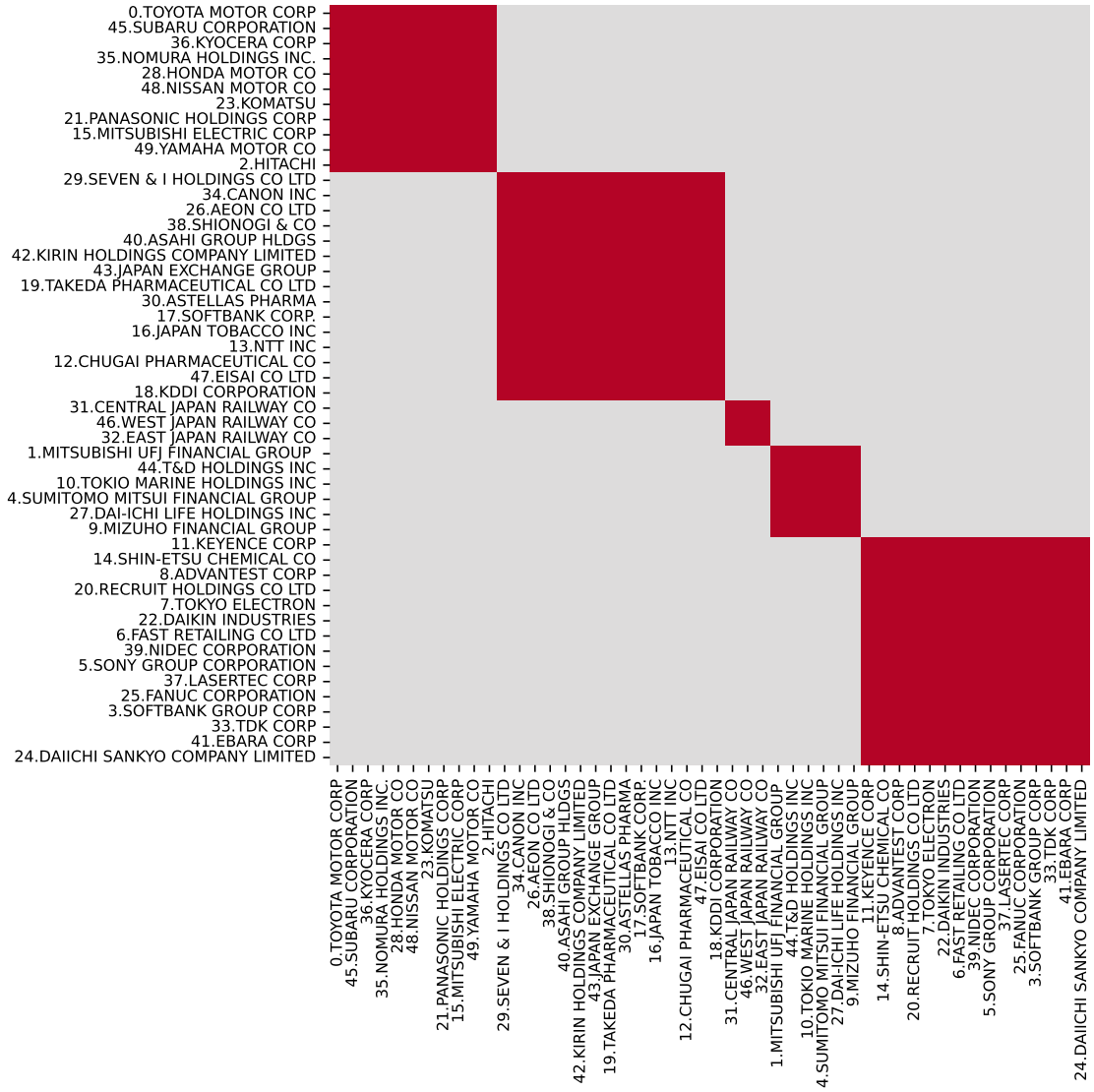


Figure 9: The cluster structure of the top 50 Japanese stocks found by the RSC. We see the rough partition into {**automotive & mobility, Japan Railway, finance, semiconductor & electronics manufacturing & technology, pharma & retail & the rest**}.