POINT LOCALIZATION AND DENSITY ESTIMATION FROM ORDINAL KNN GRAPHS USING SYNCHRONIZATION

MIHAI CUCURINGU* AND JOSEPH WOODWORTH*†

Abstract. We consider the problem of embedding unweighted, directed k-nearest neighbor graphs in low-dimensional Euclidean space. The k-nearest neighbors of each vertex provides ordinal information on the distances between points, but not the distances themselves. We use this ordinal information along with the low-dimensionality to recover the coordinates of the points up to arbitrary similarity transformations (rigid transformations and scaling). Furthermore, we also illustrate the possibility of robustly recovering the underlying density via the Total Variation Maximum Penalized Likelihood Estimation (TV-MPLE) method. We make existing approaches scalable by using an instance of a local-to-global algorithm based on group synchronization, recently proposed in the literature in the context of sensor network localization and structural biology, which we augment with a scaling synchronization step. We demonstrate the scalability of our approach on large graphs, and show how it compares to the Local Ordinal Embedding (LOE) algorithm, which was recently proposed for recovering the configuration of a cloud of points from pairwise ordinal comparisons between a sparse set of distances.

Key words. k-nearest-neighbor graphs, ordinal constraints, graph embeddings, density estimation, eigenvector synchronization, linear programming, divide-and-conquer.

1. Introduction. Embedding unweighted k-nearest neighbor (kNN) graphs is a special case of ordinal embedding, or non-metric embedding. Generally, one seeks a spatial embedding of n points $\{\vec{x}_i\}_{i=1}^n$ in $\mathbb{R}^d$ that satisfy

$$\forall (i,j,k,l) \in \mathcal{C}, \quad \|\vec{x}_i - \vec{x}_j\|_2 < \|\vec{x}_k - \vec{x}_l\|_2,$$

where $\mathcal{C}$ denotes the set of ordinal constraints. In the case of unweighted kNN graph embedding, $\mathcal{C} = \mathcal{C}(G) = \left\{(i,j,i,l) \middle| ij \in E(G), il \notin E(G)\right\}$, where $E(G)$ is the set of directed edges in the kNN graph G.

Graph-based methods are of utmost importance in several modern machine learning methods with applications such as clustering, dimensionality reduction, ranking and image processing. Many such methods rely on weighted graphs, with weights often based on similarity functions, i.e., $w_{ij} = S(x_i, x_j)$. From a practical standpoint, storing unweighted kNN graphs instead would allow for a very sparse representation of the data. If one could recover the source data $\{x_i\}_{i=1}^n$ from unweighted kNN graphs, such a computationally efficient sparser representation would incur no information loss. Because of the extreme sparsity of the representation, this is generally a hard problem. Just recently, however, a related method for recovering data distributions from unweighted kNN graphs was shown to converge to the correct solution [58], albeit in a suboptimal given regime.

The original work on this problem dates back to Shepard [47, 48] and Kruskal [33, 34], and lately has been studied intensively in the machine learning literature [42, 43, 1, 46, 38, 52, 2, 41, 39, 28, 29, 36, 60]. In this work, we compare against and extend the new Local Ordinal Embedding method presented in [53], which enjoys several favorable comparisons with other modern methods.

Another motivation for this problem comes from an instance of the popular sensor network localization problem, where each sensor is able to transmit only limited connectivity information to a central location, in the form of ID names of its k nearest neighbor sensors, but transmits neither the estimated distance measurements nor a complete list of all its neighbors within a given fixed radius. Note that either of these last two scenarios renders the localization problem (of estimating the sensor coordinates) easier to solve. Similar to the sensor network application, one could potentially apply this framework to cooperative control and sensing involving swarms of robot micro-vehicles with limited payloads communicating via radio with limited bandwidth [37, 23].

One key step presented here is an application of the As-Synchronized-As-Possible (ASAP) method [16, 17, 11], which makes existing embedding methods scalable via a divide-and-conquer, non-incremental, non-iterative local to global approach. This application reduces computational complexity, allows for massive

*Department of Mathematics, UCLA, Los Angeles, CA. {mihai, jwoodworth}@math.ucla.edu

†Corresponding author: jwoodworth@math.ucla.edu

parallelization of large problems, and increases robustness to noise. The ASAP algorithm introduced in [16], on which we rely in the present paper, renders our approach to reconstructing kNN graphs scalable to graphs with thousands or even tens of thousands of nodes, and is an example of a local-to-global approach that integrates local ordinal information into a global embedding calculation.

We detail in Section 4 the exact approach used to decompose the initial kNN graph into many overlapping subgraphs, that we shall refer to as patches throughout the rest of the paper. Each resulting patch is then separately embedded in a coordinate system of its own using an ordinal embedding algorithm, such as the recent LOE algorithm [53]. In the hypothetical scenario when LOE recovers the actual ground truth coordinates of each patch, such local coordinates agree with the global coordinates up to scaling and some unknown rigid motion (such as translation, rotation and possibly reflection), in other words, up to a similarity transformation. However, in most practical instances, it is unreasonable to expect that the LOE algorithm will recover the exact coordinates only from ordinal data. On a related note, we point out the recent work of Kleindessner and von Luxburg, who settled a long-known conjecture in the community claiming that, given knowledge of all ordinal constraints of the form $||x_i - x_j|| < ||x_k - x_l||$ between an unknown set of points $x_1, \ldots, x_n \in \mathbb{R}$ (for finite n), it is possible to approximately recover the ground truth coordinates of the points up to similarity transformations such as rotations, translations, rescalings, or reflections. Furthermore, the same authors show that the above statement holds even when we only have local information such as the distance comparisons between points in small neighborhoods of the graphs, thus giving hope for a local-to-global approach, in the spirit of the one we propose in the present paper. Finally, we also mention here a somewhat related problem, investigated in [31] in the context of localization of sensor networks, where the authors propose an algorithm that accurately recovers the ground truth coordinates based only on connectivity information in the random geometric graph model, under certain assumptions on the graph and noise model. However, this setting is quite different than the one we consider (the kNN graph versus random geometric graph) - as pointed out previously, knowledge of the sensing radius renders the density recovery problem and the closely related kNN reconstruction problems much easier to solve.

For every local patch reconstruction, there corresponds a scaling and an element of the Euclidean group Euc(2) of two-dimensional rigid transformations, and our goal is to estimate the scalings and group elements that will properly align all the patches in a globally consistent framework. The local optimal alignments between pairs of patches whose intersection is large enough, yields noisy measurements for the ratios of the above unknown group elements. Finding group elements from noisy measurements of their ratios is also known as the group synchronization problem, first encountered in the context of synchronization over $\mathbb{R}$ of clocks in a distributed network, from noisy measurements of their time offsets [32, 21]. In [50], Singer introduced spectral and semidefinite programming (SDP) relaxations for solving the angular synchronization problem over the group SO(2) of planar rotations, algorithms which served as a building block for the recent ASAP (As-Synchronized-As-Possible) graph localization algorithms [16, 17, 11]. Using the local alignments of the patches, ASAP formulates the graph realization problem as three consecutive synchronization problems that overall solve the synchronization problem over Euc(2): it uses the eigenvector method for the compact part of the group (reflections and rotations), and the least-squares method for the non-compact part (translations). We refer the reader to Figure 2 of [16] for a schematic overview of the 2D-ASAP algorithm, and to Figure 5.1 of this paper for the approach we employ in our present work, and present more details on the implementation of ASAP in the Appendix.

While in the present paper we have used the ASAP algorithm exclusively for two-dimensional experiments, we remark that the approach extends naturally to higher dimensions. In the three-dimensional case, ASAP has been recently used as a scalable robust approach to the molecule problem, well studied in the structural biology community [17]. For the d-dimensional general case, one can extend ASAP by first synchronizing over $\mathrm{O}(d)$ (or over Z_2, followed by $\mathrm{SO}(d)$), and then over R^d to recover the translations via a simple least-squares approach. Another recent application of group synchronization is to the *Structure from Motion* problem [3], a fundamental task in computer vision where one is asked to recover three-dimensional structure from a collection of images. In general, one can apply the synchronization problem whenever the underlying problem exhibits a group structure, and there are readily available (possibly

noisy) pairwise measurements of ratios of the group elements. Finally, we point out that the LOE approach that can be used to obtain the local patch embeddings required by ASAP, has a natural extension to the d-dimensional case, thus rendering the entire pipeline amenable to dealing with higher-dimensional data.

The rest of the paper is organized as follows. Section 2 is a summary of other existing methods for related problems, including the metric and ordinal embedding problems. Section 3 details our proposed linear programming based formulation for kNN embedding, and implicitly density estimation. Section 4 summarizes the approach used for breaking the initial graph into patches relying on spectral clustering and rigidity theory. Section 5 presents an addition to the existing ASAP algorithm by incorporating scaling synchronization, as well as a brief outline of the standard ASAP algorithm. In Section 6 we remark on the connection to the density estimation problem, and describe the post-processing step performed via Total-Variation Maximum Penalized Likelihood Estimation. Section 7 details the results of several experiments and compares to the existing LOE algorithm. We discuss several possible research directions in Section 8. Finally, we summarize in Appendix 9.1 the main steps of the ASAP algorithm introduced in [16], and in Appendix 9.2 basic notions from the rigidity theory literature.

2. Related Work.

2.1. Multidimensional Scaling. Broadly speaking, multidimensional scaling refers to a number of related problems and methods. In Classical Multidimensional Scaling [55], one is given all Euclidean Squared-Distance (ESD) measurements $\Delta_{ij} = \|\vec{x}_i - \vec{x}_j\|_2^2$ on a set of points $X = \{\vec{x}_i\}_{i=1}^n$ and wishes to approximate the points themselves, assuming that they approximately lie in a low-dimensional space $d \ll n$. The standard method is to apply Principal Component Analysis to the Gram matrix $K = X^T X$, which has a linear relationship with the ESD matrix given by

$$\begin{aligned} \Delta_{ij} &= K_{ii} - 2K_{ij} + K_{jj} \\ K = -\frac{1}{2} V \Delta V, \qquad &\text{where } V = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T, \end{aligned} \tag{2.1}$$

assuming the points have mean 0. Let $\Pi_{\sigma_1,\ldots,\sigma_r}(A)$ denote the projection of A onto its r leading singular vectors. Recall that solution of

$$\hat{B} = \arg\min_B \|A - B\|_F^2 \text{ s.t. } \operatorname{rank}(B) \leq r$$

is given by $\hat{B} = \Pi_{\sigma_1,\ldots,\sigma_r}(A)$. Then Classical Multidimensional Scaling can be written as

$$\begin{aligned} \hat{K} = &\arg\min_{K \in \mathbb{S}_+^n, \Delta(K)} \|\Delta - \Delta(K)\|_F^2 \\ &\text{s.t. } \Delta(K)_{ij} = K_{ii} - 2K_{ij} + K_{jj}, \\ &\sum_{i,j} K_{ij} = 0, rank(K) \leq d \\ =&\Pi_{\sigma_1,\ldots,\sigma_d}\left(-\frac{1}{2} V \Delta V\right) \end{aligned}$$

$$\hat{X} = \Lambda^{1/2}\Phi^T, \text{ where } \hat{K} = \Phi\Lambda\Phi^T.$$

We note that the solution for the coordinates themselves is unique only up to rigid transformations, and that solutions do not exist for all possible input Δ. In particular, the matrix Δ is only a ESD corresponding to n points in $\mathbb{R}^d$ if $K = -\frac{1}{2} V \Delta V$ gives a true Gram matrix (i.e. $K \in \mathbb{S}_+^n$) that has rank $\leq d$.

One can generalize Classical Multidimensional Scaling to incorporate additional nonnegative weights W_{ij} on each distance. In particular, this makes sense when some distances are missing, or most distances are noisy, but some are known. The optimization involves minimizing an energy known in the literature as *stress*. There are a few variants, but one popular stress is Kruskal's stress [35]

$$\min_X \sigma(X) = \sqrt{\frac{\sum_{ij} W_{ij} \left(D_{ij} - D_{ij}(X)\right)^2}{\sum_{ij} W_{ij} D_{ij}^2}} \text{ s.t. } X \in \mathbb{R}^{d \times n}.$$

Here D denotes the unsquared Euclidean distance, i.e., $D_{ij} = \sqrt{\Delta_{ij}}$ and $D_{ij}(X) = \|x_i - x_j\|_2$. One approach to minimize this energy is to iteratively minimize a majorizing function of two variables, where we say that $\sigma(X, Y)$ majorizes $\sigma(X)$ if $\sigma(X, Y) \geq \sigma(X)$ and $\sigma(X, X) = \sigma(X)$. For non-classical multidimensional scaling, one specifies metric or non-metric. The above is metric Multidimensional Scaling since the input matrix D represents explicit measurements of the metric.

A further generalization of metric Multidimensional Scaling is non-metric Multidimensional Scaling, or Ordinal Embedding, in which the input D is assumed to be an increasing function applied to distance measurements [47, 48]. This may be the case if D represents dissimilarity between points, as opposed to measured distances. The problem can again be expressed with stress functionals

$$\begin{aligned} &\min_X \sigma(X) = \sqrt{\frac{\sum_{ij} W_{ij} \left(D_{ij} - f(D_{ij}(X))\right)^2}{\sum_{ij} W_{ij} D_{ij}^2}}, \\ &\text{s.t. } X \in \mathbb{R}^{d \times n}, f \text{ increasing} \end{aligned}$$

and is usually solved with isotonic regresion [34].

2.2. Semidefinite Programming methods. Semidefinite Programming methods (SDP) have been applied frequently to MDS and related problems. Classical MDS can be stated as an SDP, with a closed form solution. Any formulation of the problem that optimizes over the Gram matrix requires the semidefinite constraint $K \in \mathbb{S}_+^n$. Indeed, for metric MDS, if one penalizes the squared error on the squared distance measurements, the problem can be written as

$$\begin{aligned} &\min_{X \in \mathbb{R}^{d \times n}} \sum_{ij} W_{ij} (\Delta_{ij} - \Delta_{ij}(X))^2 \\ = &\min_{K \in \mathbb{S}_+^n, X \in \mathbb{R}^{d \times n}} \sum_{ij} W_{ij} (\Delta_{ij} - (K_{ii} - 2K_{ij} + K_{jj}))^2 \\ &\text{s.t. } K = X^T X. \end{aligned}$$

Constraints of the form $K = X^T X$ are usually not allowed however, and are typically relaxed to [54, 45]

$$\begin{bmatrix} I & X \\ X^T & K \end{bmatrix} \succeq 0.$$

via Schur's Lemma. Furthermore, one encourages K to be approximately low-rank by introducing a nuclear norm or trace penalty $\|K\|_* = \|\sigma(K)\|_1 = tr(K)$, as a convex relaxation of a rank constraint. Intuitively, since the ℓ_1 norm promotes sparsity, the nuclear norm should promote few nonzero singular values. Elsewhere [61], it is argued that one should maximize $tr(K)$, in the spirit of the popular Maximum Variance Unfolding approach [61]. Neither minimizing nor maximizing the trace actually imposes an exact rank constraint, which is non-convex and NP-hard. One approach that could achieve exact rank constraints would be to use the Majorized Penalty Approach of Gao and Sun [20] with an alternating minimization method.

A group of methods have studied the graph realization problem, where one is asked to recover the configuration of a cloud of points given a sparse and noisy set of pairwise distances between the points [10, 8, 9, 7, 62]. One of the proposed approaches involves minimizing the following energy

$$\min_{p_1,\ldots,p_n\in\mathbb{R}^2} \sum_{(i,j)\in E} \left(\|p_i - p_j\|^2 - d_{ij}^2\right)^2. \tag{2.2}$$

which unfortunately is nonconvex, but admits a convex relaxation into a SDP program. We refer the reader to Section 2 of [16] for several variations of this approach, some of which have been shown to be more robust to noise in the measured distances.

2.3. Local Ordinal Embedding. Terada and von Luxburg [53] have recently proposed an algorithm for ordinal embedding and kNN embedding specifically, called Local Ordinal Embedding. LOE minimizes a soft objective function that penalizes violated ordinal constraints, with a scale parameter $\delta > 0$ included,

$$\min_{X\in R^{d\times n}} \sum_{i<j,k<l,(i,j,k,l)\in\mathcal{C}} \max\left[0, D_{ij}(X) + \delta - D_{kl}(X)\right]^2.$$

An advantage of this energy in contrast to ones that normalize by the variance of X (to guarantee nondegeneracy) is its relatively simple dependence on X, making the above energy easier to minimize. In their work, they introduce a few different algorithms to minimizing the above energy, one based on majorization minimization and another one based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation of Newton's method. The same group also presented some theoretical results on the uniqueness of solutions to the ordinal embedding problem [59], as well as a different approach to kNN embedding, via density estimation [58].

3. A Linear Program Alternative to SDP embedding. In contrast to the SDP methods which cast embedding problems in terms of the Gram matrix K our Linear Program (LP) approach for kNN embedding optimizes over the variables D (the distance matrix), R (the radius at each node), and the slack variables. The radius at each node i, denoted by R_i is defined to be the distance between node i and its k-th closest neighbor. Thus R_i is the radius of the neighborhood at node i. In kNN embedding, the objective and constraints can be written as linear constraints in D, R and the slack variables, altogether leading to a linear program (LP) which is computationally cheaper to solve than an SDP. Although SDP-based methods can encompass a larger class of problems, they currently do not approach the scalability or numerical maturity of LP and SOCP solvers.

Our proposed LP approach requires solving a linear program for candidate distance matrix D and radii R, then feeding the resulting D into a standard mdscale (see Algorithm 1), where by $\mathcal{T}$ we mean the

Algorithm 1 LP approach

$$\begin{aligned}
(D^*, R^*) = \underset{R\in\mathbb{R}^{n\times 1}, D\in\mathbb{R}^{n\times n}_{+,sym},\alpha,\beta}{\arg\min} \quad & \sum_{ij\in E(G)} \alpha_{ij} + \sum_{ij\notin E(G)} \beta_{ij} \\
\text{subject to} \quad & D_{ij} \le R_i + \alpha_{ij}, \text{ if } j \in \mathcal{N}_i \\
& D_{ij} > R_i - \beta_{ij}, \text{ if } j \notin \mathcal{N}_i \\
& \alpha_{ij} \ge 0, \beta_{ij} \ge 0 \\
& R_i > 0, \forall i = 1, \ldots, n \\
& \sum_{i=1}^{n} R_i = V \\
& D_{ij} + D_{ik} \le D_{kl}, (i,j,k) \in \mathcal{T} \\
X = & \text{mds}\,(D^*, d)
\end{aligned}$$

set of triangle inequalities we considered (ordered set (i,j,k)). If $(i,j,k) \in \mathcal{T}$, the same holds true for the two other permutations. The full set of triangle inequalities are necessary, though not sufficient, for the matrix D to correspond to an Euclidean distance matrix. If one omits slack variables, there are $n(n-1)/2$ distance values to solve for along with n radii, and thus $n(n+1)/2$ unknowns in total. Considering the ordinal constraints, for the upper bounds on the entries D_{ij}, there are n ways to choose i, and for each i there are k ways to choose j, thus $nk/2$ constraints (accounting for symmetric distances). For the lower bounds on the entries D_{ij} there are n ways to choose i and for each i there are $n-k-1$ ways to choose j, giving $n(n-k-1)/2$ constraints. So there are $n(n-1)/2$ ordinal constraints on relating the $n(n-1)/2$ distances and n radii. In other words, the intuition behind the added triangle inequalities is that they help to better constrain the system. There are on the order of n^3 triangle inequalities (choose any three points), so for large n, there are many more constraints than unknowns.

To avoid the added complexity from imposing all triangle inequalities, one could consider models that impose only a fraction of such constraints via either imposing them locally, for k-hop neighboring triples of points, or globally, such as picking edges via an Erdős-Rényi model, or mixing the two approaches.

We remark that dropping triangle inequalities altogether could certainly speed up the embedding process. The resulting non-metric D may correspond to an increasing function of distance (e.g., distance squared), which suggests that non-metric MDS would be appropriate.

In general, even if the recovered distance metric corresponds to a metric distance, this is not a guarantee that the distance is realizable in a low-dimensional space. That requires a rank constraint on D, which is non-convex and is computationally intractable for an LP or SDP. The ultimate embedding into a low-dimensional space thus potentially gives up some structure in both the LP and SDP formulation, and it can be argued that this effect is lessened via the local to global approach.

4. Breaking up the kNN graph into patches. The first step we use in breaking the kNN graph into patches is to apply normalized spectral clustering [57] to a symmetrized version of the graph. Normalized spectral clustering segments the nodes of a graph into $N \ll n$ clusters by performing k-means on the n entries of the leading N eigenvectors of the random-walk normalized graph Laplacian. It is shown [57] that normalized spectral clustering minimizes a relaxation of the normalized graph cut problem. From there, we enlarge the clusters with their 1-hop neighborhood, so that the resulting patches have significant overlap, a prerequisite for the ASAP synchronization algorithm. The higher the overlap between the patches, the more robust the pairwise group ratio estimates would be, thus leading overall to a more accurate final global solution. Finally, we check each patch for global rigidity. We comment more about global rigidity and its role here later in this section. If a patch is not globally rigid, we drop a constant fraction of the added nodes [1] with the lowest degree while retaining all nodes that were in the original cluster generated by k-means in the corresponding patch. This uses the heuristic that low-degree nodes tend to render a graph not globally rigid. After dropping nodes, we check the remaining patch for globally rigidity again. We stop the pruning process when the patch contains fewer than $4/3$ the number of nodes in the original cluster, or the patch is globally rigid.

We refer the readers to Appendix 9.2 for for a brief description of global rigidity, and relevant results in the literature, and use the remainder of this section as a brief discussion of the main definitions. In the *graph realization problem* (GRP), one is given a graph $G = (V, E)$ together with a non-negative distance measurement d_{ij} associated with each edge, and is asked to compute a realization of G in $\mathbb{R}^d$. In other words, for any pair of adjacent nodes i and j, the distance $d_{ij} = d_{ji}$ is available, and the goal is to find a d-dimensional embedding $p_1, p_2, \ldots, p_n \in \mathbb{R}^d$ such that $\|p_i - p_j\| = d_{ij}$, for all $(i,j) \in E$. The main difference between the GRP and the problem we aim to address in our paper is the input information available to the user. Unlike the GRP problem where distances are available to the user, here we only have information of the adjacency matrix of the graph and have the knowledge that it represents a kNN graph. Both problems aim to recover an embedding of the initial configuration of points.

A graph is globally rigid in $\mathbb{R}^d$ if there is a unique (up to the trivial Euclidean isometries) embedding of the graph $\mathbb{R}^d$ such that all distance constraints are preserved. It is well known that a necessary

[1] At each round we choose to drop a quarter of the nodes

condition for global rigidity is 3-connectivity of the graph. Since the problem at hand that we are trying to solve is harder (as we do not have distance information available) we require that the patches we generate are globally rigid graphs. Even in the favorable scenario when we do have available distance measurements (which we do not in the present problem, but only ordinal information), any algorithm seeking an embedding of the graph would fail if the graph were to have multiple non-congruent realizations.

5. ASAP synchronization with global scaling. Before applying the ASAP algorithm to the embedded patches, we introduce an additional step that further improves our approach. In the graph realization problem, distance information is readily available to the user and thus the local embedding of the patches are on the same scale as the ground truth solution. One need only estimate the rigid transformation that aligns the local frame of each patch with the global solution. However, in the kNN embedding problem, distances are not known and thus the scale of one patch relative to another must be approximated. Any ordinal embedding approach has no way of relating the scaling of the local patch to the global scale. To this end we augment the ASAP algorithm of [16, 17] by introducing an additional step where we synchronize local scaling information to recover an estimate for the global scaling of each patch thus overall synchronizing over the group of similarity transformations.

The pipeline of steps is as follows. Given a set of patches, $\{P_i\}_{i=1}^N$, we can create a patch graph in which two patches are connected if and only if they have sufficiently many nodes in common[2]. We then construct a matrix $\Lambda \in \mathbb{R}^{N\times N}$ as

$$\Lambda_{ij} = \begin{cases} \text{median}\left\{D_{a,b}^{P_i}/D_{a,b}^{P_j}\right\}_{a\neq b\in P_i\cap P_j} & \text{if } P_i \sim P_j,\ i \leq j, \\ 1/\Lambda_{ji} & \text{if } P_i \sim P_j,\ i > j, \\ 0 & \text{otherwise,} \end{cases}.$$

The matrix Λ approximates the relative scales between patches. If all distances in all patches were recovered correctly up to scale, and all patches had sufficient overlap with each other, then each row of Λ would be a scalar multiple of the others and each column of Λ would be scalar multiple of the others. Therefore the desired Λ matrix is rank-1, and in order to get a consistent estimate of global scaling, we compute the best rank-1 approximation of Λ (up to a scalar multiple), given by its leading eigenvector. We use this approximation of global scaling to rescale the embedded patches before running ASAP.

To illustrate the importance of the scaling synchronization step in Figure 7.1 we compare ASAP synchronized embeddings of a set of points with and without the scaling synchronization step. Clearly, the scaling synchronization step significantly improves the recovery of the original points.

We summarize the ASAP method here and present more details on its implementation in the Appendix. After applying the optimal scaling to each patch embedding, we perform the ASAP synchronization-based algorithm to integrate all patches in a global framework, as illustrated in the pipeline shown in Figure 5.1. We estimate, for each patch P_i, an element of the Euclidean group $\text{Euc}(2) = \text{O}(2) \times\mathbb{R}^2$ which, when applied to that patch embedding P_i, aligns all patches as best as possible in a single coordinate system. In doing so, we start by estimating, for each pair of overlapping patches P_i and P_j, their optimal relative rotation and reflection, i.e., an element R_{ij} of the orthogonal group O(2) that best aligns P_j with P_i. Whenever the patch embeddings perfectly match the ground truth, $R_{ij} = O_iO_j^{-1}$. Finding group elements $\{O_i\}_{i=1}^N$ from noisy measurements of their ratios is also known as the group synchronization problem, an NP-hard problem for which spectral and SDP relaxations exist [50]. We rely on the spectral relaxation for synchronization over O(2), and estimate a consistent global rotation (and possibly reflection) of each patch via an eigenvector problem, using the top $d = 2$ eigenvectors of the associated graph Connection Laplacian [51], thus solving (5.1). Finally, we estimate the optimal translations of each patch

[2]In general aligning two patches in $\mathbb{R}^d$ can be done provided the two patches overlap in at least $d+1$ points, which stems from the fact that the union of two globally rigid graphs P_i and P_j that intersect in at least $d+1$ nodes is itself a globally rigid graph.

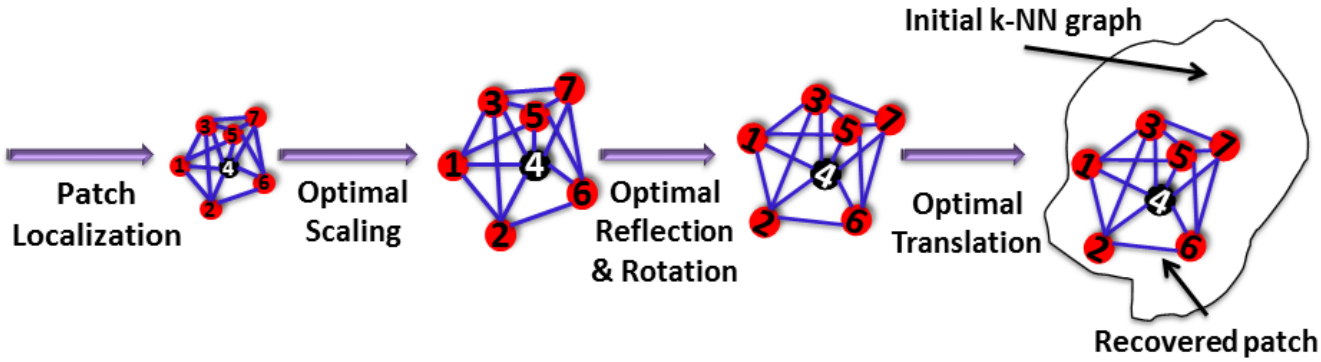

FIG. 5.1. *ASAP and scale synchronization diagram.*

via least squares on an overdetermined linear system.

$$\min_{O_1,\ldots,O_N \in O(2)} \sum_{P_i \sim P_j} \|O_i^{-1} O_j - R_{ij}\|_F^2. \tag{5.1}$$

6. Density Estimation. In this section, we remark on the explicit connection between the graph embedding problem considered in this paper and the density estimation problem. In particular, one may approach the problem of recovering the unknown coordinates underlying the kNN graph by first aiming to estimate the density function that generates the coordinates. Suppose for example that one is able to estimate the pointwise density $u : \Omega \subseteq \mathbb{R}^d \to [0, 1]$, up to some constant multiple, evaluated at each vertex of the graph, x_i. Next, as outlined in [58], one can assign weights to the originally unweighted kNN graph, defined by $w(x_i, x_j) = (u^{-1/d}(x_i) + u^{-1/d}(x_j))/2$. Furthermore, it can be shown that the shortest path distance on resulting weighted kNN graphs converges to the Euclidean distance of the original points as the number of points increases. In other words, applying multidimensional scaling to the shortest path distances on the weighted kNN graph will yield increasingly accurate embeddings of the original points $\{x_i\}_{i=1}^n$ as $n \to +\infty$.

In contrast to finding an approximate embedding from a density estimate, under certain conditions, the reverse process is also straightforward. With sufficiently many points and sufficiently strong priors on the distribution, the methodology of Maximum Penalized Likelihood Estimation (MPLE) applies [18]. One first assumes that the locations correspond to points drawn independently identically distributed according to some unknown underlying spatial distribution. MPLE approximates the most likely spatial distribution given the points observed and some assumed prior distribution on the space of distributions. The data fidelity term comes in the form of a log-likelihood term, a function of the distribution estimate and the point locations, and is given by

$$L(u, \{x_i\}_{i=1}^n) = \sum_{i=1}^n \log(u(x_i)),$$

and the penalty term, $P(u)$ enforces the prior distribution on the space of distributions. Typical choices for $P(u)$ include the H^1-seminorm regularizer, $P(u) = \frac{\lambda}{2} \int_\Omega |\nabla u|^2 dx$, enforcing smoothness, and Total Variation (TV) norm regularization, $P(u) = \lambda \int_\Omega |\nabla u| dx$, which enforces smoothness, but also allows for edges. Therefore, general MPLE seeks to optimize the following energy over all probability distributions on the spatial domain $\Omega \subseteq \mathbb{R}^d$

$$\hat{u} = \underset{u \geq 0, \int_\Omega u dx = 1}{\arg\max} \; L(u, \{x_i\}_{i=1}^n) - P(u).$$

The form and scale of P encodes different types and amounts of regularity in the resulting density estimate u. In practical settings, cross-validation should be performed to determine the appropriate amount of regularity to impose on a given data set.

For the purpose of using kNN graphs to recover densities, we will include a post-processing step for a subset of the embedding experiments, to which we apply a standard implementation of TV MPLE [40] to the embedded points. TV is a good choice of penalty because we will be applying it to points that are drawn from a piecewise constant density. The good density estimates based on good embeddings shown in Section 7 illustrate that there is in fact a strong connection between the embedding and density estimation problems.

The actual implementation of the TV MPLE relies on the Split Bregman (equivalently Alternating Direction Method of Multipliers) minimization technique in which one introduces a splitting and equality constraints that are enforced by performing saddle-point optimization of the augmented Lagrangian. This results in an iterative update procedure given by

$$\begin{aligned}\left(\hat{u},\hat{d}\right) &= \underset{u\geq 0,d}{\arg\min}\left\{\|d\|_1 - \sum_{i=1}^{n}\log(u(x_i)) + \frac{\rho}{2}\|\nabla u - d + y\|_2^2 + \frac{\gamma}{2}\left(\|u\|_1 - 1 + z\right)^2\right\},\\ y &= y + \tilde{\nabla}\hat{u} - \hat{d}\\ z &= z + \|\hat{u}\|_1 - 1.\end{aligned}$$

The first minimization step is actually replaced by minimizing over u, and d individually, making use of the shrinkage proximal operator associated with the ℓ^1 norm.

7. Experimental Section. We experiment with three different densities, piecewise constant half-plane and square densities, and the two-dimensional Gaussian distribution. We test Laplacian Eigenmaps [6], the LOE BFGS and MM method [53], and ASAP with LOE BFGS used for the patch embedding. Throughout the rest of the paper, we refer to the latter approach as ASAP LOE BFGS. Because LOE was already compared with several methods in [53], attaining better performance than LOE suggests better performance than a number of relevant methods such as the method of Kamada and Kawai [30], the method of Fruchterman and Reingold [19], and t-SNE [56]. The input to each method consists of kNN adjacency matrices corresponding to points sampled from one of the three distributions. For each distribution, sample size, n, and graph parameter k, we only sample the points once so that each method gets the same adjacency matrix. We evaluate the methods based on runtime and error, which we define as the percentage of edge disagreement between the kNN adjacency matrix of the embedded points and the original kNN adjacency matrix. In other words, for n points in $\mathbb{R}^d$ represented by the columns of $X \in \mathbb{R}^{d\times n}$, and given k, if we let $A_X^k \in \{0,1\}^{n\times n}$ denote the adjacency matrix of the corresponding kNN graph, then the ordinal error of $\tilde{X}$ in recovering X is given by

$$\text{error}(\tilde{X}, X) = \frac{1}{n^2}\sum_{i,j=1}^{n}\left|\left(A_{\tilde{X}}^k\right)_{ij} - \left(A_X^k\right)_{ij}\right|.$$

Also to allow for a fair comparison, we set varying limits on the number of LOE iterations $\{5, 10, 50, 100, 300, 500\}$, and we use varying maximum patch sizes for ASAP (which relates indirectly to the number of patches). In this way, the LOE methods and the ASAP method give for each distribution and values n and k, an error-runtime Pareto curve (with low values in both coordinates being best). We show in Figure 7.2, adjacency matrix error versus runtime of these methods applied to sample sizes $n = \{500, 1000, 5000\}$ and $k = \lceil 2\log(n)\rceil$, i.e., sparse adjacency matrices. We show in Figure 7.3, adjacency matrix error versus runtime of these methods applied to sample sizes $n = \{500, 1000, 5000\}$ and $k = \lceil\sqrt{n\log(n)}\rceil$, i.e., dense adjacency matrices. Though adjacency matrix error better captures the extent to which the methods solve the ordinal problem, for completeness we also evaluate the methods using the Procrustes error [49] that corresponds to registering the embedded points with the original points. We show in Figure 7.4, Procrustes error versus runtime of these methods applied to sample sizes $n = \{500, 1000, 5000\}$ and $k = \lceil 2\log(n)\rceil$, i.e., sparse adjacency matrices. We show in Figure 7.5, Procrustes error versus runtime of these methods applied to sample sizes $n = \{500, 1000, 5000\}$ and $k = \lceil\sqrt{n\log(n)}\rceil$, i.e., dense adjacency matrices. We show in Figure 7.9, scaled adjacency matrix error and procrustes error versus increasing

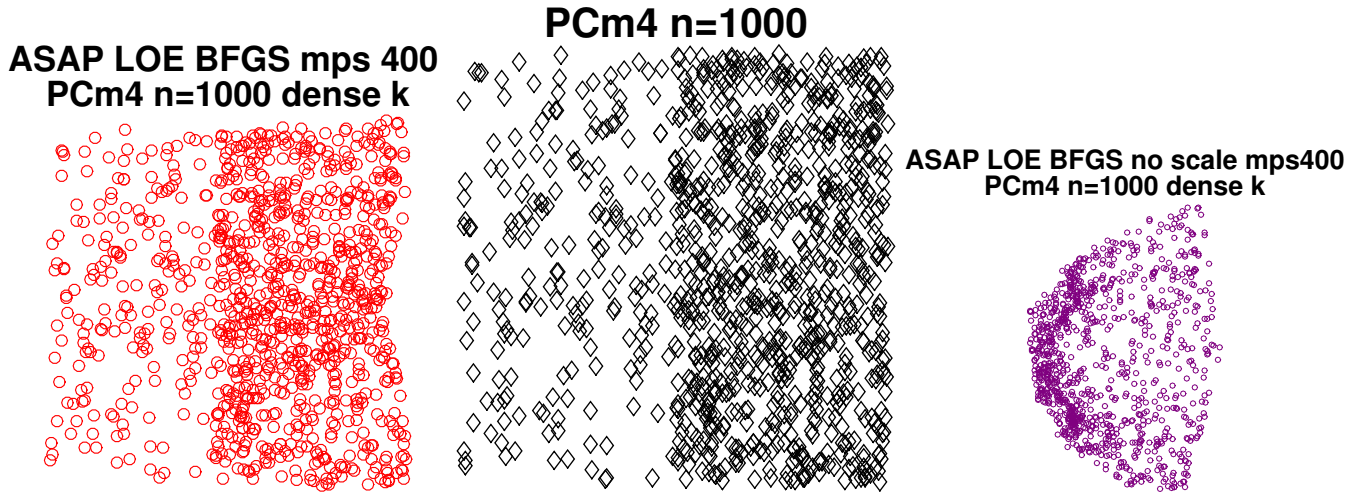

FIG. 7.1. *ASAP LOE BFGS max patch size 400,* $n = 1000$*,* $k = 14$ *Left: with scale synchronization: time 463.869980 s, 0.006602 A error, 0.0031053 Procrustes error Middle: Original embedding of the points Right: without scale synchronization: time 468.174236 s, 0.037506 A error, 0.1132305 Procrustes error*

values of k for $n = \{5000\}$ points drawn from the piecewise constant half-planes distribution using the method ASAP LOE BFGS with max patch size 300. To further illustrate how the methods perform, we plot the Procrustes aligned points for $n = 1000$ sampled from each of the three densities in Figure 7.6. In each instance plotted, the ASAP LOE BFGS with max patch size 400 took less time to run and yields smaller adjacency matrix and Procrustes errors than the LOE BFGS with 100 maximum iterations. We only run LOE MM for $n = 500$ because of difficulties we had when trying to get the provided R implementation [3] to run on our Linux-based remote computing resource. We ran into no problems with the LOE BFGS implementation. The computers used have 12 CPU cores which are Intel(R) Xeon(R) X5650 @ 2.67GHz, and have 48GB ram. The R implementation of LOE does not (as far as the authors are aware) take advantage of multiple cores, and runs a single process on a single core. In contrast, our ASAP Matlab implementation uses the Multicore package [4] to divide up the local embedding problems among the available cores. Ideally, one would run these experiments many times over and average the results (to get an estimate of average performance), but this is effect already partially accomplished by running the LOE and ASAP methods with multiple parameters to get a more holistic measurement of performance.

In several instances we find that ASAP + LOE BFGS is either faster than or better-performing than direct LOE BFGS, or both. This is partly due to the massively parallel embedding step in ASAP, which can take advantage of multiple cores as the problem scales. One would expect that as n continues to grow, if more processors are made available and memory increases sufficiently, the advantage of embedding parallelization would continue to increase.

We also see in Figure 7.9 and Figure 7.10 that for large n, adjacency matrix error and Procrustes error remain relatively small and stable over a range of small increasing k.

In Figure 7.7 we show the results of applying TV MPLE to some of the embeddings shown in Figure 7.6. The regularization parameter used is .0001 . This is not obtained by cross-validation, but it simply seems to perform well on the originally sampled points. The densities of the approximate embeddings are as expected, with ASAP LOE BFGS recovering the density best, with LOE BFGS behind, and LE doing the worst. This shows suggests that better embedding results do lead to better density estimation, if that

[3] The package is available for download at
http://cran.r-project.org/web/packages/loe/index.html

[4] The package is available for download at
http://www.mathworks.com/matlabcentral/fileexchange/13775-multicore-parallel-processing-on-multiple-cores

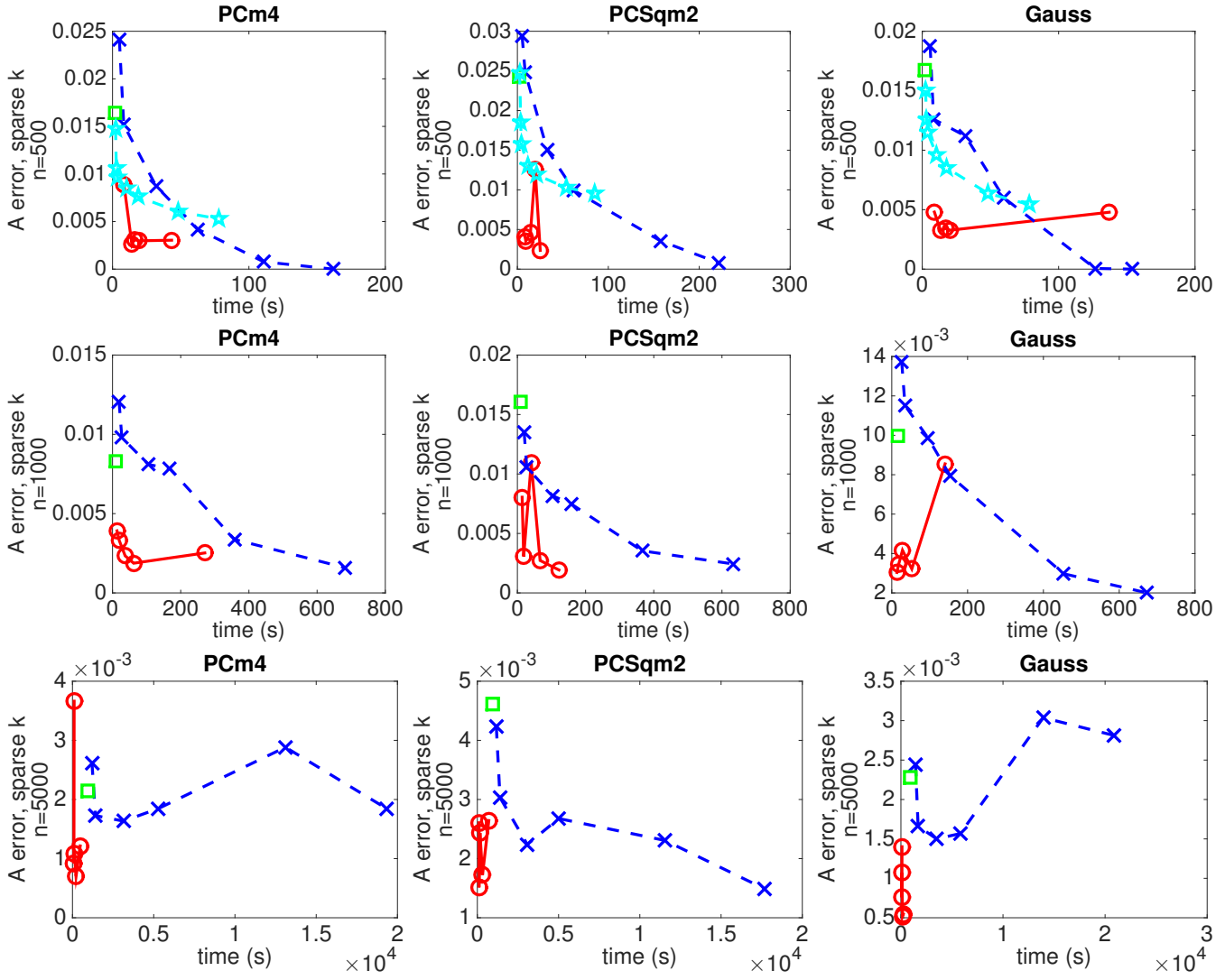

FIG. 7.2. *Adj. matrix error vs. time,* $n = \{500, 1000, 5000\}$*, k sparse,* ○ *ASAP LOE BFGS,* × *LOE BFGS,* □ *LE ,* ⋆ *LOE MM*

is the end goal.

In Figure 7.8 we show adjacency matrix error vs runtime and Procrustes error versus runtime for ASAP LOE BFGS run on a data set of $n = 50,000$ points with $k = 22$. Despite this increase in the size of the problem, ASAP LOE BFGS is able to produce reasonable results in a reasonable amount of time. On the other hand, for $n = 50,000$, we were unable to get any LOE BFGS results because the full setup of the problem in R, with some small Matlab overhead, does not fit within the memory constraints of the machine.

8. Summary and discussion. Our experiments have demonstrated that the computation efficiency of existing methods for the kNN embedding problem can be significantly improved, while maintaining or improving spatial and ordinal accuracy when run in a distributed setting. Our application of the local to global, divide-and-conquer ASAP method renders the problem of kNN embedding in low-dimensions significantly more tractable, distributing the embedding steps, and using fast spectral methods to combine them. We expect that such improvements will make it possible to use kNN embeddings in a broader range of data sets and settings. We summarize below some potential future research directions that may further improve on our proposed method.

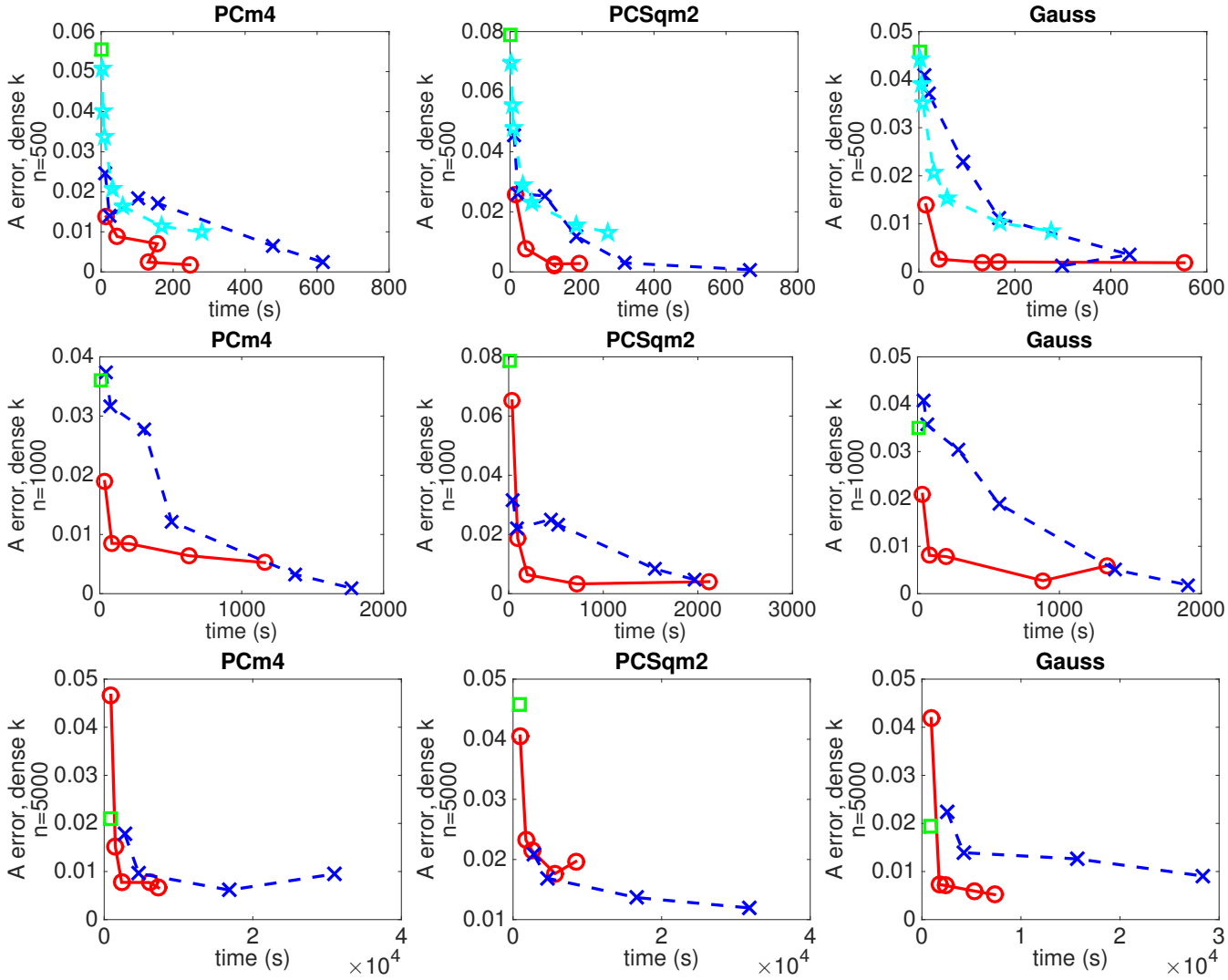

FIG. 7.3. *Adj. matrix error vs. time,* $n = \{500, 1000, 5000\}$*, k dense,* ○ *ASAP LOE BFGS,* × *LOE BFGS,* □ *LE ,* ⋆ *LOE MM*

8.1. Patch extraction via redundant spectral clustering. In contrast to the 1-hop neighborhood expanded spectral clusters used for patches discussed in Section 4, we discuss a possible alternative for extracting patches, which guarantees the redundant coverage of each node thus benefiting ASAP. Suppose we seek N patches in such that each node appears in P distinct patches. First, we perform spectral clustering on the initial graph, clustering into $c = N/P$ clusters, via repeated instances of the k-means algorithm with different random initializations. Next, we pick the P partitions in such a way that minimizes the resulting Jaccard index between each partition [26]. This results in N patches, no two of which are very similar, but that cover every node exactly P times. We also expect that the resulting patches should also be very dense, thus leading to more noise-robust embeddings.

8.2. Alternating minimization method. In the interest of preserving ordinal structure exactly in a low-dimensional setting, we present an alternating minimization approach. We approximate a rank constraint on the Gram matrix K by minimizing the residual of K projected onto the current low-rank approximation of K [20] (see Algorithm 2).

Because we are attempting to solve a non-convex problem, we have no expectation of converging to the global minimum of this problem. One subproblem involves projecting onto the nearest low-rank matrix (as discussed above), and the other subproblem solves ordinal embedding with a regularizer dependent on the previous low-rank approximation. To project onto the nearest low-rank matrix, and SVD computation

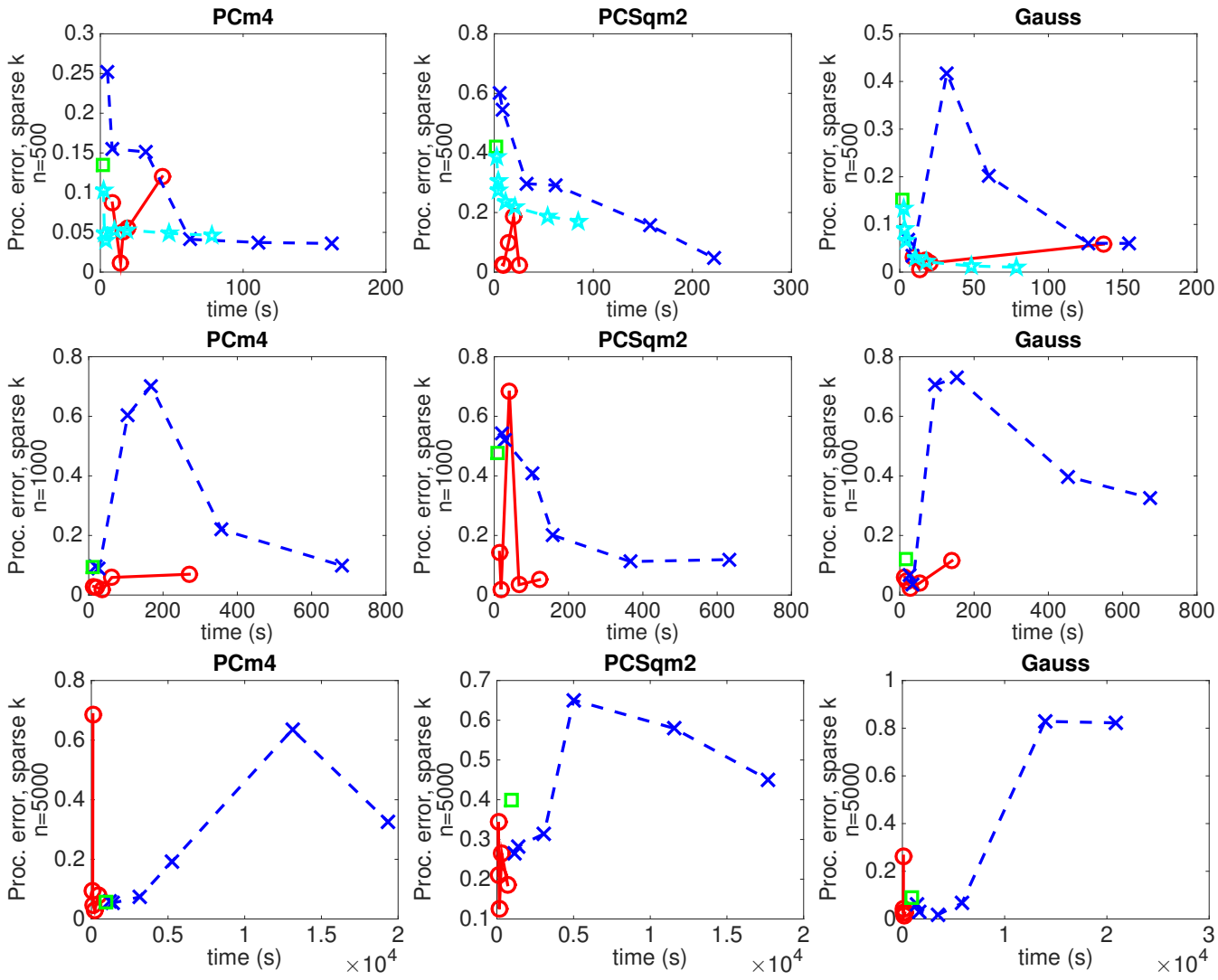

FIG. 7.4. *Procrustes error vs. time,* $n = \{500, 1000, 5000\}$, *k sparse,* ○ *ASAP LOE BFGS,* × *LOE BFGS,* □ *LE ,* ⋆ *LOE MM*

Algorithm 2 Alternating minimization method

Initialize $L = 0$ or $L = \text{LP-Embedding}(A)$

while L doesn't satisfy the ordinal constraints

$$\begin{cases} (K^*, R^*) = & \arg\min_{K \in \mathbb{S}^n_+, R} trace\left(K(I - U_L V_L^T)^T\right) \\ \text{subject to} & ij \in E(G) \Rightarrow (K_{ii} - 2K_{ij} + K_{jj}) \leq R_i \\ & ij \notin E(G) \Rightarrow (K_{ii} - 2K_{ij} + K_{jj}) > R_i \\ L = & \Pi_{\sigma_1,\ldots,\sigma_d}(K^*) \\ L = & (1/2)(L + L^T) \end{cases}$$

$X = \Lambda^{1/2}\Phi^T$, where $L = \Phi\Lambda\Phi^T$

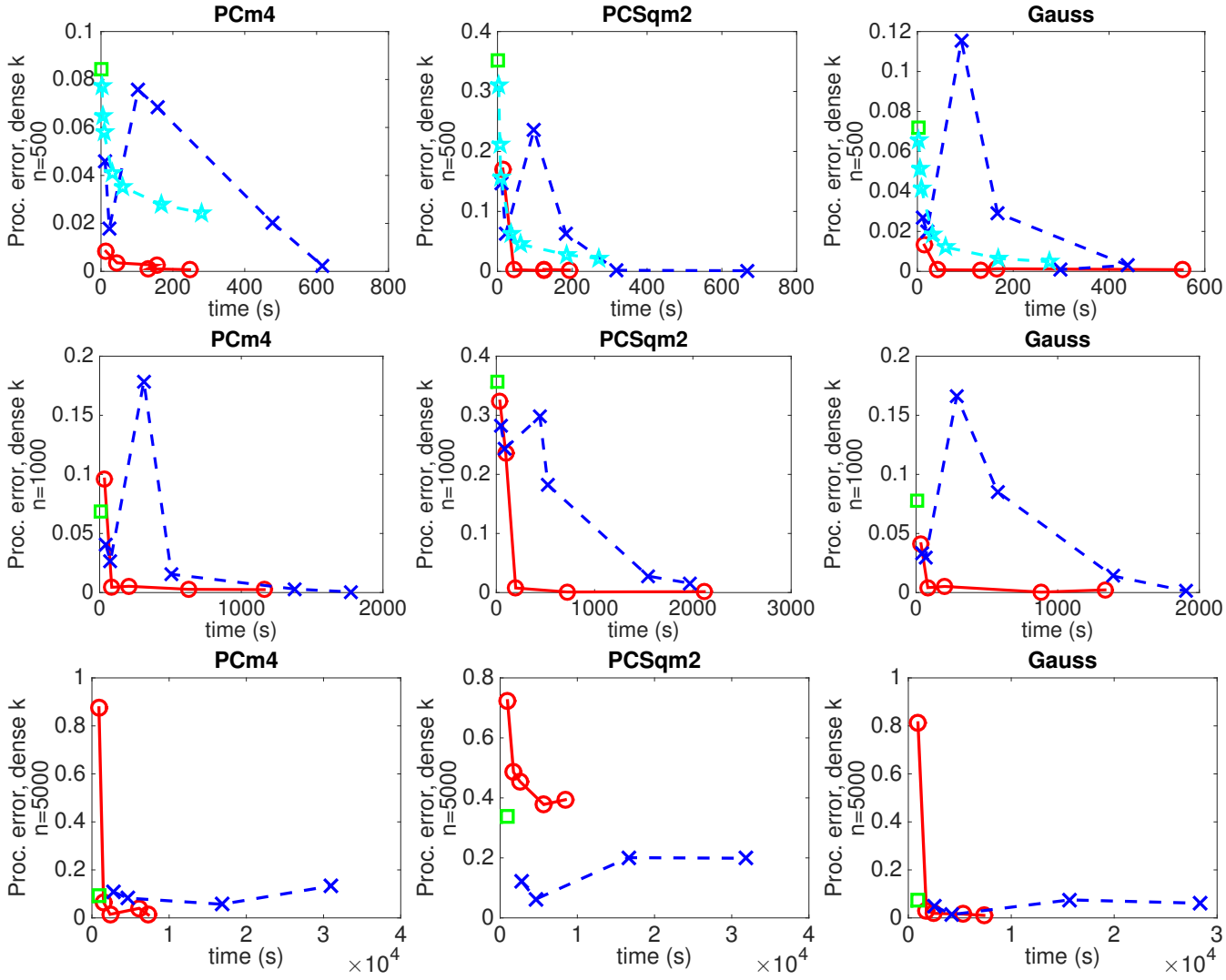

FIG. 7.5. *Procrustes error vs. time,* $n = \{500, 1000, 5000\}$*, k dense,* ○ *ASAP LOE BFGS,* × *LOE BFGS,* □ *LE ,* ⋆ *LOE MM*

is needed, and the ordinal subproblem is still an SDP, which may need to be solved many times. It would likely help performance to initialize the low-rank approximation with the approximate solution from another, fast ordinal problem solver. It is not clear, however, that fast convergence would be attainable, nor whether embedding points that exactly satisfy the ordinal constraints in a given low-dimensional space is worth the extra effort, especially if noisy measurements are expected.

8.3. Rigidity for Ordinal Constraints. Another possible future research direction relates to developing an analogous rigidity theory for graphs arising from ordinal constraints information. For example, one may encode the given ordinal information into a graph $\tilde{G}(\tilde{V}, \tilde{E})$ whose vertex set $\tilde{V}$ is given by all $\binom{n}{2}$ pairs of nodes $u \in \tilde{G}$ with $u = (i, j) \in V(G) \times V(G)$, and there exists a directed edge $(u, v) \in \tilde{E}$ if and only if one has available ordinal constraints for a quadruple $\{i, j, k, l\}$, such that $u = (i, j) \in \tilde{V}$, $v = (k, l) \in \tilde{V}$ and $\text{dist}(x_i, x_j) < \text{dist}(x_k, x_l)$. The goal of ordinal embedding is to find a realization of the vertices of G in $\mathbb{R}^d$, such that all the given ordinal constraints encoded in the graph $\tilde{G}$ are preserved.

A related problem worth investigating, for a set of ordinal constraints given as quadruples (i, j, k, l), is whether there are infinitely many or finitely many possible realizations, or a unique realization subject to the given ordinal constraints, and up to local permissible displacements of the points (for example, allowing each point to move only within a ball of small fixed radius r). The latter condition excludes the trivial case when one may slightly perturb a satisfactory embedding P to produce yet another embedding

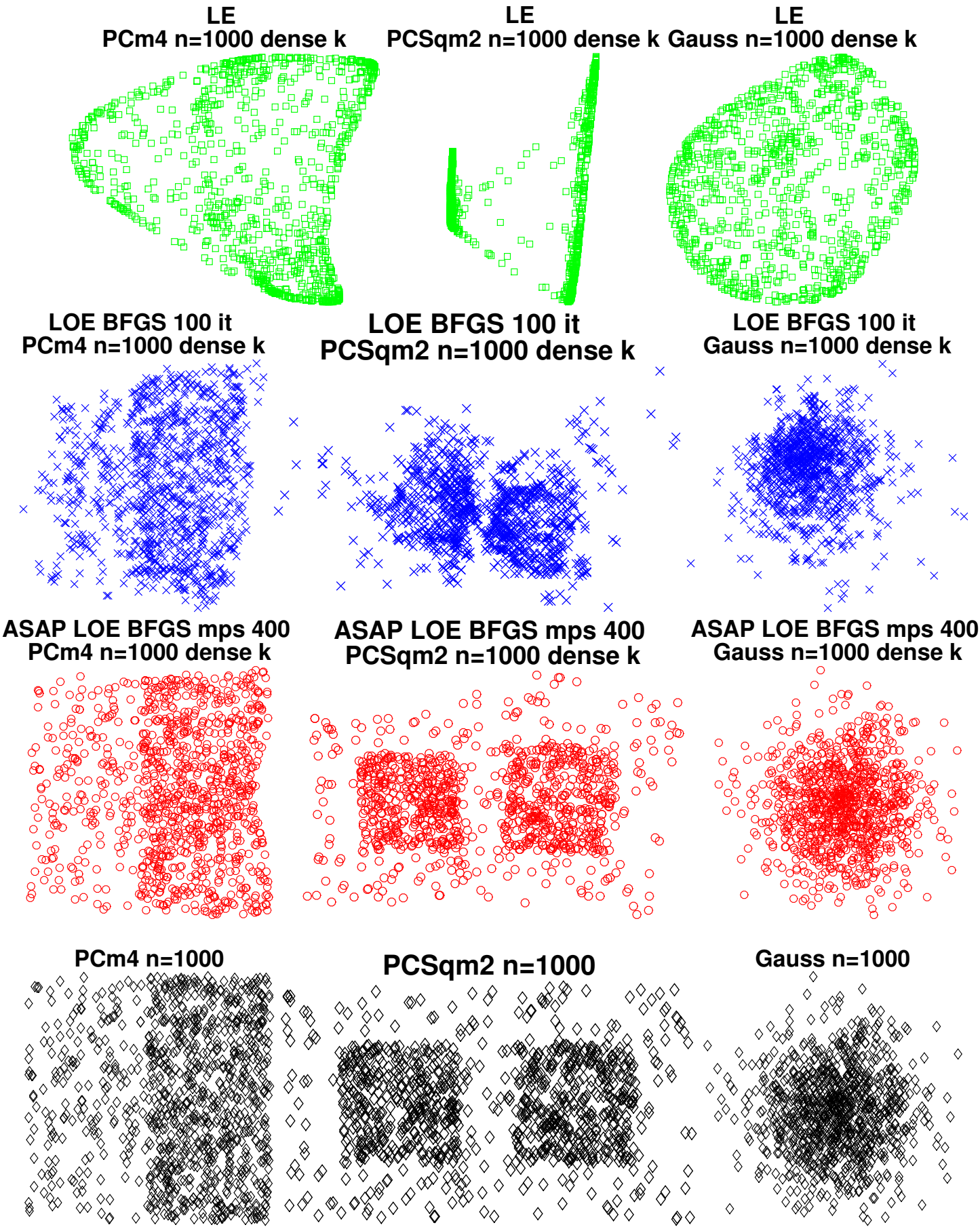

FIG. 7.6. *Example embeddings for $n = 1000$, k dense, and rows corresponding to LE, LOE BFGS maxIt=100, ASAP LOE BFGS max patch size 400 (the latter one being always faster than the corresponding LOE BFGS).*

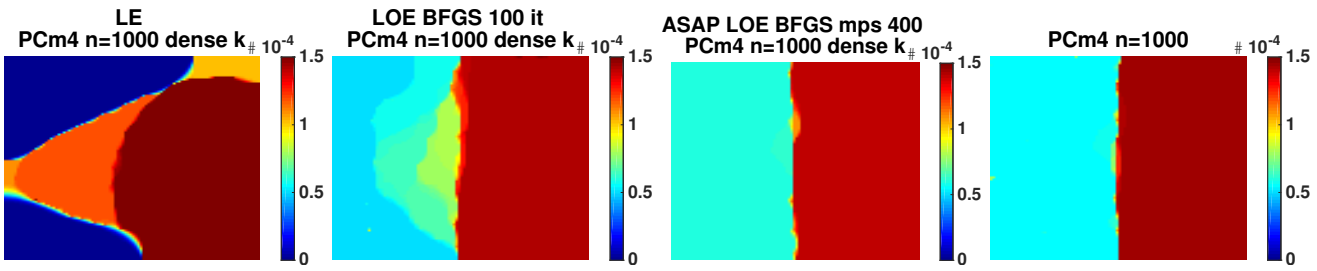

FIG. 7.7. *TV MPLE applied to example embeddings for $n = 1000$, k dense, and columns corresponding to LE, LOE BFGS maxIt=100, ASAP LOE BFGS max patch size 400 (see Figure 7.5)*

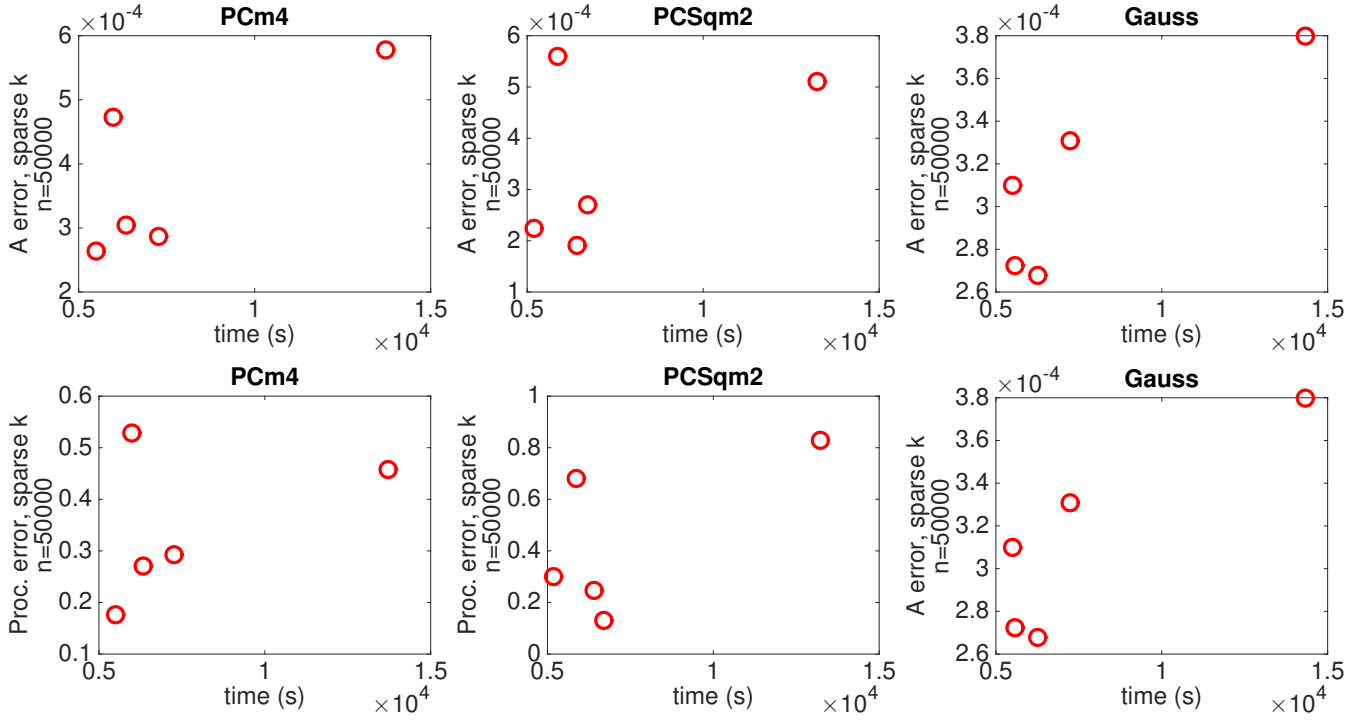

FIG. 7.8. *Top row: A errors vs time for $n = 50,000$, $k = 22$ sparse, Bottom row: A errors vs time for $n = 50,000$, k sparse*

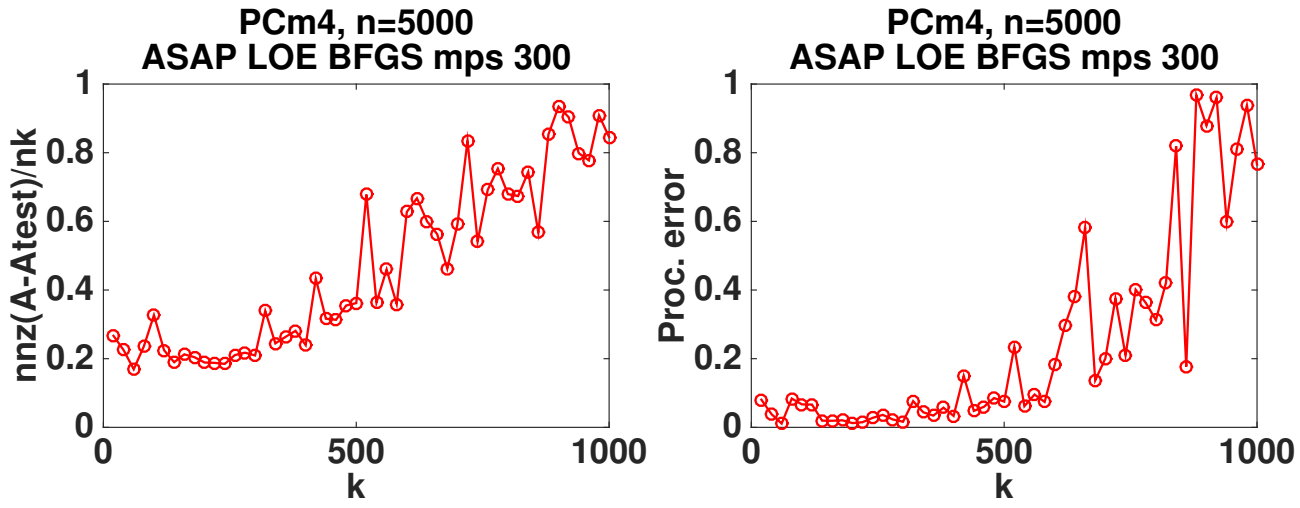

FIG. 7.9. *ASAP LOE BFGS max patch size 300, $n = 5000$, k increasing by 20, Left: number of differences in adjacency matrix divided by number of edges, nk, Right: Procrustes error.*

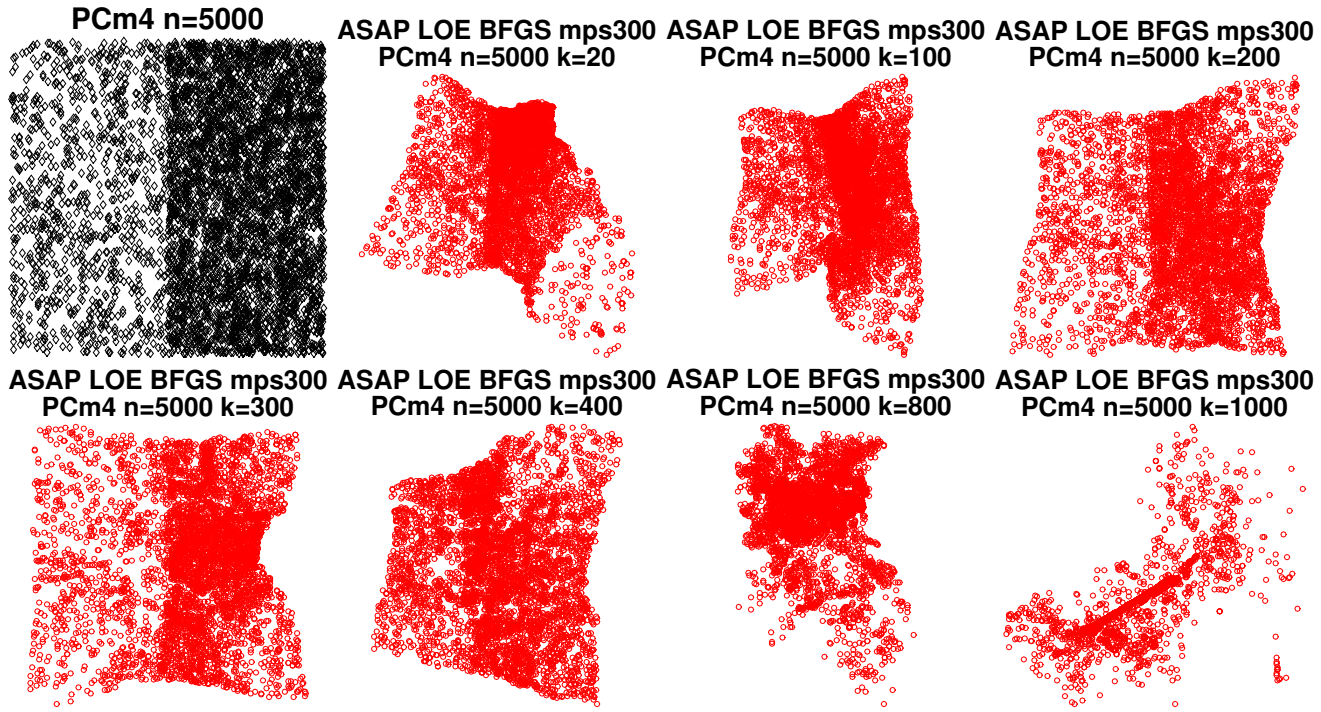

FIG. 7.10. *ASAP LOE BFGS max patch size 300, $n = 5000$, k increasing by 20, Top left : originally sampled points, Remaining plots : recovered embeddings*

Q that satisfies all given ordinal constraints, with P and Q not necessarily congruent. We expect that such a characterization would allow for a principled approach to extracting patches from the graph of constraints $\tilde{G}$, thus leading to a more robust local-to-global approach, such as the one considered here.

Acknowledgements. The authors would like to thank Andrea Bertozzi for her support throughout this project. M.C. acknowledges support from AFOSR MURI grant FA9550-10-1-0569. J.W. acknowledges support from NSF grant no. DGE-1144087 and the UC Lab Fees Research grant 12-LR-236660. Part of this work was undertaken while M.C. and J.W. were attending the Semester Program on Network Science and Graph Algorithms at the Institute for Computational and Experimental Research in Mathematics (ICERM) at Brown University - we thank our hosts for their warm hospitality. We would also like to thank Andrea Bertozzi for organizing the research cluster at ICERM, Ulrike von Luxburg and Mauro Maggioni for helpful conversations, and Yoshikazu Terada for providing the LOE code.

9. Appendix.

9.1. ASAP Implementation.

This section details aspects of the group synchronization problem used in the ASAP algorithm [16]. In the first step, ASAP solves a synchronization problem over $\mathbb{Z}_2$ for the possible reflections of the patches using the eigenvector method, while in the second step, it solves a synchronization problem over SO(2) for the rotations also using the same eigenvector method. Note that we choose to combine these two steps into a single one, and synchronize over the orthogonal group $O(2) = Z_2 \times SO(2)$, similar to the approach in [17]. By pairwise aligning overlapping patches we consider the following $2n \times 2n$ matrix given by

$$R_{ij} = \begin{cases} r_{ij} & (i,j) \in E^P \ (P_i \text{ and } P_j \text{ have enough common points}) \\ O_{3\times 3} & (i,j) \notin E^P \ (P_i \text{ and } P_j \text{ cannot be aligned}) \end{cases}$$

The least squares solution to synchronization over $R_1, \ldots, R_n \in O(d)$ minimizes the following sum of squared deviations

$$\underset{R_1,\ldots,R_n \in O(d)}{\text{minimize}} \sum_{(i,j)\in E} w_{ij} \|R_i^{-1} R_j - R_{ij}\|_F^2 \tag{9.1}$$

where $\|\cdot\|$ denotes the Frobenius norm, w_{ij} are non-negative weights representing the confidence in the noisy pairwise measurements R_{ij}. Spectral and semidefinite programming relaxations for solving an instance of the above synchronization problem were originally introduced and analyzed by Singer, [50], in the context of angular synchronization, over the group SO(2) of planar rotations, where one is asked to estimate n unknown angles $\theta_1, \ldots, \theta_n \in [0, 2\pi)$ given m noisy measurements δ_{ij} of their offsets $\theta_i - \theta_j \mod 2\pi$. The difficulty of the problem is amplified on one hand by the amount of noise in the offset measurements, and on the other hand by the fact that $m \ll \binom{n}{2}$, i.e., only a very small subset of all possible pairwise offsets are measured. In general, one may consider other groups $\mathcal{G}$ (such as SO(d), O(d)) for which there are available noisy measurements g_{ij} of ratios between the group elements $g_{ij} = g_i g_j^{-1}, g_i, g_j \in \mathcal{G}$.

ASAP relies on the spectral relaxation of the above minimization problem, via the graph Connection Laplacian L. Letting $W \in \mathbb{R}^{nd\times nd}$ with blocks $W_{ij} = w_{ij} R_{ij}$, and $D \in \mathbb{R}^{nd\times nd}$ diagonal with $D_{ii} = d_i I_d$ where $d_i = \sum_j w_{ij}$, the graph Connection Laplacian is defined as

$$L = D - R, \qquad \text{with} \quad L\bar{R}^T = 0$$

Note that $L \succeq 0$, and in the noiseless case $LR^T = 0$. We recover the estimated rotations from the bottom d eigenvectors of L, by projecting each of the resulting $d \times d$ matrices to the closest rotation matrix, via a simple SVD decomposition. In practice, we choose to work with the $\mathcal{L} = D^{-1}R$, similar to the symmetric matrix $D^{-1/2}RD^{-1/2}$ via

$$\mathcal{H} = D^{-1/2}(D^{-1/2}RD^{-1/2})D^{1/2}.$$

Note that for the normalized graph Connection Laplacian

$$\mathcal{L} = D^{-1/2}LD^{-1/2} = I_{nd} - D^{-1/2}WD^{-1/2},$$

Bandeira, Singer, and Spielman proved recently [5] a Cheeger-type inequality, providing a deterministic worst case performance guarantee for the synchronization problem over the group O(d) of orthogonal transformations.

If we let h_i denote the $d\times d$ matrix corresponding to the i^{th} submatrix in the $d\times N$ matrix $[v_1^{\mathcal{H}}, \ldots, v_d^{\mathcal{H}}]$, in the noise free case, h_i is the solution that aligns patch P_i in the global coordinate system (up to a global orthogonal transformation). We denote by h the $dN \times d$ matrix formed by concatenating the true orthogonal transformation matrices $h_1, \ldots, h_N$ and by G^P the patch graph, with nodes corresponding to the patches $P_1, \ldots, P_n$ and edges to pairwise alignments of patches which overlap. If G^P is complete, it can be easily seen that H is a rank d matrix since $H = hh^T$, and its top three eigenvectors are given by the columns of h

$$Hh = hh^T h = hNI_3 = Nh$$

With a bit of extra work, it can be shown that in the general and more realistic case when G^P is a sparse connected graph

$$Hh = Dh, \text{ hence } D^{-1}Hh = \mathcal{H}h = h, \tag{9.2}$$

with the columns of h as eigenvectors of $\mathcal{H}$, with eigenvalue $\lambda = 1$ of multiplicity d. It can be shown that $\lambda = 1$ is the largest eigenvalue of $\mathcal{H}$.

Note that if the user has readily available information about the embedding of certain patches and possibly of their ground truth reflection and rotations (often referred to as *anchor* nodes), then it is possible to incorporate such constraints in the synchronization problem. We refer the readers to Section 7 of [17] for an analysis of the synchronization problem $\mathbb{Z}_2$ with anchor information, which we solve via

Quadratically Constrained Quadratic Programming. In the setting where one has further additional information that certain subsets of nodes represent the same (unknown) group element, in [15] we consider and compare several algorithms for synchronization over $\mathbb{Z}_2$, based on spectral and semidefinite programming relaxations (SDP), and message passing algorithms. In the final step, ASAP solves a synchronization problem over $\mathbb{R}^2$ for the translations by solving an overdetermined linear system of equations using the least squares method, a solution which yields the estimated coordinates of all the nodes up to a global rigid transformation.

Finally, we remark that from a computational point of view, all steps of the algorithm can be implemented in a distributed manner and scale linearly in the size of the network, except for the eigenvector computation, which is almost linear, since each iteration of the power method is linear in the number of edges of the graph, but the number of iterations is greater than $O(1)$ as it depends on the spectral gap. We refer the reader to Section 7 in [16] for a complexity analysis of the 2D-ASAP algorithm, and also demonstrate its scalability by reconstructing a kNN-graph with 50,000 nodes, as illustrated in Figure 7.8. Several other additional experiments demonstrate the robustness of the proposed approach to noise and to sparse connectivity of the kNN graph providing the ordinal information.

9.2. Rigidity Theory Appendix. One of the main questions in the field of rigidity theory asks whether one can uniquely determine (up to rigid transformations, such as translations, rotations, reflections) the coordinates of a set of points $p_1, \ldots, p_n$ given a partial set of distances $d_{ij} = ||p_i - p_j||$ between n points in $\mathbb{R}^d$. To make our paper self-contained, this short appendix if a very brief summary of the main definitions and results related to local and global rigidity from the literature (e.g., [14, 24, 25, 44, and references therein]). Readers who are unfamiliar with rigidity theory may use this short Appendix as a glossary. As previously discussed in Section 4, one of the steps of the divide-and-conquer approach proposed for the kNN-recovery problem relies to testing whether the underlying resulting patches are globally rigid. As observed in our numerical simulations detailed in Figures 7.2,7.3,7.4,7.5,7.6 the final reconstruction is more accurate when we rely on global rigidity as a postprocessing step for the partitions obtained via spectral clustering. The intuition behind our approach is as follows. In the case when distance information is available, testing for global rigidity is a crucial step in making sure that each of the local patches has a unique embedding in its own reference frame, approximatively consistent with the ground truth, up to a rigid transformation. Since in the kNN-recovery problem, we do not have distance information but only ordinal data, thus we are faced with solving even a harder problem, we expect that the global rigidity check will improve the accuracy of the local patch embeddings. One specific example where our current rigidity heuristics improved results was in performing ASAP LOE BFGS with max patch size 300, on $n = 5000$ points drawn from the constant half-plane distribution, letting $k = 18$. In that example, performing the rigidity check and pruning gave a runtime of 107.056 s, an ordinal error of 0.00107096, and 0.0585465 Procrustes error, while skipping the rigidity check and pruning gave a runtime of 192.606 s, an ordinal error of 0.00154208 A error, and 0.175992 Procrustes error.

A *bar and joint framework* in $\mathbb{R}^d$ is defined as an undirected graph $G = (V, E)$ ($|V| = n, |E| = m$) together with a *configuration* p which assigns a point p_i in $\mathbb{R}^d$ to each vertex i of the graph. The edges of the graph correspond to distance constraints, that is, $(i, j) \in E$ if an only there is a bar of length d_{ij} between points p_i and p_j. We say that a framework $G(p)$ is *locally rigid* if there exists a neighborhood U of $G(p)$ such that $G(p)$ is the only framework in U with the same set of edge lengths, up to rigid transformations. In other words, there is no continuous deformation that preserves the given edge lengths. A configuration is *generic* if the coordinates do not satisfy any non-zero polynomial equation with integer coefficients (or equivalently algebraic coefficients).

Local rigidity in $\mathbb{R}^d$ has been shown to be a generic property, in the sense that either all generic frameworks of the graph G are locally rigid, or none of them are. A consequence of the seminal results of Gluck [22] and Asimow and Roth [4] asserts that the dimension of the null space of the rigidity matrix is the same at every generic point, and hence local rigidity in $\mathbb{R}^d$ is a generic property, meaning that either all generic frameworks of the graph G are locally rigid, or none of them are. With probability one, the rank of the rigidity matrix that corresponds to the unknown true displacement vectors equals the rank of

the randomized rigidity matrix. A similar randomized algorithm for generic local rigidity was described in [24, Algorithm 3.2]. In other words, *generic local rigidity* in $\mathbb{R}^d$ can be considered a combinatorial property of the graph G itself, independent of the particular realization. Using this observation, generic local rigidity can therefore be tested efficiently in any dimension using a randomized algorithm [25]: one can just randomize the displacement vectors $p_1, \ldots, p_n$ while ignoring the prescribed distance constraints that they have to satisfy, construct the so called *rigidity matrix* corresponding to the framework of the original graph with the randomized points and check its rank. This is approach we use to make sure the obtained patches are local rigid.

Since local generic rigidity does not imply unique realization of the framework, it is possible that there exist multiple non-congruent realizations that satisfy the prescribed distances (which we do not even have available in the kNN recovery problem) One may consider for example, the $2D$-rigid graph with $n = 4$ vertices and $m = 5$ edges consisting of two triangles that can be folded with respect to their joint edge. We call a framework $G(p)$ *globally rigid* in $\mathbb{R}^d$ if all frameworks $G(q)$ in $\mathbb{R}^d$ which are $G(p)$-equivalent (have all bars the same length as $G(p)$) are congruent to $G(p)$ (i.e., related by a rigid transformation). Hendrickson proved two key necessary conditions for global rigidity of a framework at a generic configuration:

THEOREM 9.1 (Hendrickson [25]). *If a framework $G(p)$, other than a simplex, is globally rigid for a generic configuration p in $\mathbb{R}^d$ then:*

- *The graph G is vertex $(d+1)$-connected;*
- *The framework $G(p)$ is edge-2-rigid (or, redundantly rigid), in the sense that removing any one edge leaves a graph which is infinitesimally rigid.*

We say that a graph G is *generically globally rigid* in $\mathbb{R}^d$ if $G(p)$ is globally rigid at all generic configurations p [12, 13]. Though it has been conjectured for many years that global rigidity is a generic property, this fact was shown to be true only very recently. The seminal work of [13, 24] proves that global rigidity is a generic property of the graph in each dimension. The conditions of Hendrickson as stated in Theorem 9.1 are necessary for generic global rigidity. They are also sufficient on the line, and in the plane [27]. However, by a result of Connelly [12], $K_{5,5}$ in 3-space is generically edge-2-rigid and 5-connected but is not generically globally rigid.

One of the tools used in testing for global rigidity of frameworks relies on the notions on stress matrices, more popular perhaps in the engineering community. A *stress* is defined an assignment of scalars w_{ij} to the edges of the given graph G such that for every node $i \in V$ it holds that

$$\sum_{j:\,(i,j)\in E} \omega_{ij}(p_i - p_j) = 0. \tag{9.3}$$

Alternatively, it can be show that a stress is a vector w in the left null space of the rigidity matrix: $R_G(p)^T w = 0$. A stress vector can be rearranged into an $n \times n$ symmetric matrix Ω, known as the *stress matrix*, such that for $i \neq j$, the (i,j) entry of Ω is $\Omega_{ij} = -\omega_{ij}$, and the diagonal entries for (i,i) are $\Omega_{ii} = \sum_{j:\,j\neq i} \omega_{ij}$. Since all row and column sums are zero, it follows that the all-ones vector $(1\ 1\ \cdots\ 1)^T$ is in the null space of Ω as well as each of the coordinate vectors of the configuration p. Therefore, it follows that for generic configurations the rank of the stress matrix is at most $n - (d+1)$. The following pairs of theorems give sufficient and necessary conditions for generic global rigidity:

THEOREM 9.2 (Connelly [13]). *If p is a generic configuration in $\mathbb{R}^d$, such that there is a stress, where the rank of the associated stress matrix Ω is $n - (d+1)$, then $G(p)$ is globally rigid in $\mathbb{R}^d$.*

THEOREM 9.3 (Gortler, Healy, and Thurston [24]). *Suppose that p is a generic configuration in $\mathbb{R}^d$, such that $G(p)$ is globally rigid in $\mathbb{R}^d$. Then either $G(p)$ is a simplex or there is a stress where the rank of the associated stress matrix Ω is $n - (d+1)$.*

Based on the latter theorem, the authors of [24] also provided a randomized polynomial algorithm for checking generic global rigidity of a graph [24, Algorithm 3.3], which we use to test for global rigidity of the patches in the kNN-recovery problem. If a given patch is generically locally rigid then their algorithm picks a random stress vector of the left null space of the rigidity matrix associated to this patch, and converts it into a stress matrix. If the rank of the stress matrix is exactly $n - (d+1)$, then we conclude that the patch is globally rigid, and if the rank is lower, then the respective patch is not globally rigid.

REFERENCES

[1] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. J. Kriegman, and S. Belongie. Generalized non-metric multidimensional scaling. In *International Conference on Artificial Intelligence and Statistics*, pages 11–18, 2007.

[2] N. Ailon. Active learning ranking from pairwise preferences with almost optimal query complexity. In *NIPS*, pages 810–818, 2011.

[3] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, Zurich, Switzerland, October 13-15, 2012*, pages 81–88, 2012.

[4] L. Asimow and B. Roth. The rigidity of graphs. *Trans. Amer. Math. Soc.*, 245:279–289, 1978.

[5] A. S. Bandeira, A. Singer, and D. A. Spielman. A cheeger inequality for the graph connection laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013.

[6] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[7] P. Biswas, H. Aghajan, and Y. Ye. Semidefinite programming algorithms for sensor network localization using angle of arrival information. In *Proc. 39th Annu. Asilomar Conf. Signals, Systems, and Computers*, pages 220–224, Oct. 2005.

[8] P. Biswas, T. C. Lian, T. C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks*, 2(2):188–220, 2006.

[9] P. Biswas, T. Liang, K. Toh, Y. Ye, and T. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371, 2006.

[10] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pages 46–54, New York, 2004. ACM.

[11] K. N. Chaudhury, Y. Khoo, and A. Singer. Global registration of multiple point clouds using semidefinite programming. *SIAM Journal on Optimization, accepted*, 2013.

[12] R. Connelly. On generic global rigidity. *Applied Geometry and Discrete Mathematics*, 4:147–155, 1991.

[13] R. Connelly. Generic global rigidity. *Discrete Comput. Geom*, 33:549–563, 2005.

[14] R. Connelly and W. J. Whiteley. Global rigidity: The effect of coning. *Discrete and Computational Geometry*, 2009.

[15] M. Cucuringu. Synchronization over Z_2 and community detection in bipartite networks. in progress.

[16] M. Cucuringu, Y. Lipman, and A. Singer. Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Trans. Sen. Netw.*, 8(3):19:1–19:42, Aug. 2012.

[17] M. Cucuringu, A. Singer, and D. Cowburn. Eigenvector synchronization, graph rigidity and the molecule problem. *Information and Inference*, 1(1):21–67, 2012.

[18] P. P. B. Eggermont and V. N. LaRiccia. *Maximum Penalized Likelihood Estimation: Regression*, volume 2. Springer, 2001.

[19] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[20] Y. Gao and D. F. Sun. A majorized penalty approach for calibrating rank constrained correlation matrix problems. 2010.

[21] A. Giridhar and P. R. Kumar. Distributed clock synchronization over wireless networks: Algorithms and analysis. In *45th IEEE Conference on Decision and Control*, pages 4915–4920, 2006.

[22] H. Gluck. Almost all simply connected closed surfaces are rigid. *Geometric Topology, Lecture Notes in Mathematics*, 438:225–239, 1975.

[23] M. Gonzalez, X. Huang, D. S. H. Martinez, C. H. Hsieh, Y. R. Huang, B. Irvine, M. B. Short, and A. L. Bertozzi. A third generation micro-vehicle testbed for cooperative control and sensing strategies. In *ICINCO (2)*, pages 14–20, 2011.

[24] S. J. Gortler, A. D. Healy, and D. P. Thurston. Characterizing generic global rigidity. *AMERICAN JOURNAL OF MATHEMATICS*, 4:897, 2010.

[25] B. Hendrickson. Conditions for unique graph realizations. *SIAM J Comput*, 21:65–84, 1992.

[26] P. Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[27] B. Jackson and T. Jordán. Connected rigidity matroids and unique realizations of graphs. *Journal of Combinatorial Theory, Series B*, 94(1):1–29, 2005.

[28] K. G. Jamieson and R. D. Nowak. Active ranking using pairwise comparisons. In *NIPS*, volume 24, pages 2240–2248, 2011.

[29] K. G. Jamieson and R. D. Nowak. Low-dimensional embedding using adaptively selected ordinal data. In *Communication, Control, and Computing (Allerton)*, pages 1077–1084. IEEE, 2011.

[30] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.

[31] A. Karbasi and S. Oh. Distributed sensor network localization from local connectivity: Performance analysis for the hop-terrain algorithm. *SIGMETRICS Perform. Eval. Rev.*, 38(1):61–70, June 2010.

[32] R. Karp, J. Elson, D. Estrin, and S. Shenker. Optimal and global time synchronization in sensornets. Technical report, Center for Embedded Networked Sensing, University of California, Los Angeles, 2003.

[33] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.
[34] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.
[35] J. B. Kruskal and M. Wish. *Multidimensional scaling*, volume 11. Sage, 1978.
[36] Y. Lan, J. Guo, X. Cheng, and T.-Y. Liu. Statistical consistency of ranking methods in a rank-differentiable probability space. In *NIPS*, pages 1241–1249, 2012.
[37] D. S. H. Martinez, M. Gonzalez, X. Huang, B. Irvine, C. H. Hsieh, Y. R. Huang, M. B. Short, and A. L. Bertozzi. An economical testbed for cooperative control and sensing strategies of robotic micro-vehicles. In *Informatics in Control, Automation and Robotics*, pages 65–75. Springer, 2013.
[38] B. McFee and G. R. Lanckriet. Partial order embedding with multiple kernels. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 721–728. ACM, 2009.
[39] B. McFee and G. R. Lanckriet. Metric learning to rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 775–782, 2010.
[40] G. O. Mohler, A. L. Bertozzi, T. A. Goldstein, and S. J. Osher. Fast tv regularization for 2d maximum penalized likelihood estimation. *Journal of Computational and Graphical Statistics*, 20(2):479–491, 2011.
[41] H. Ouyang and A. Gray. Learning dissimilarities by ranking: from sdp to qp. In *Proceedings of the 25th international conference on Machine learning*, pages 728–735. ACM, 2008.
[42] M. Quist and G. Yona. Distributional scaling: An algorithm for structure-preserving embedding of metric and nonmetric spaces. *The Journal of Machine Learning Research*, 5:399–420, 2004.
[43] R. Rosales and G. Fung. Learning sparse metrics via linear programming. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 367–373. ACM, 2006.
[44] B. Roth. Rigid and flexible frameworks. *The American Mathematical Monthly*, 88:6–21, 1981.
[45] D. Shamsi, N. Taheri, Z. Zhu, and Y. Ye. On Sensor Network Localization Using SDP Relaxation. *ArXiv e-prints*, Oct. 2010.
[46] B. Shaw and T. Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 937–944. ACM, 2009.
[47] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function i. *Psychometrika*, 27(2):125–140, 1962.
[48] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function ii. *Psychometrika*, 27(3):219–246, 1962.
[49] R. Sibson. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society. Series B*, pages 234–238, 1978.
[50] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Appl. Comput. Harmon. Anal.*, 30(1):20–36, 2011.
[51] A. Singer and H.-T. Wu. Vector diffusion maps and the connection laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012.
[52] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*, 2011.
[53] Y. Terada and U. V. von Luxburg. Local ordinal embedding. In *Proceedings of the 31st International Conference on Machine Learning*, pages 847–855, 2014.
[54] K. Toh, P. Biswas, and Y. Ye. SNLSDP- a MATLAB software for sensor network localization, October 2008.
[55] W. S. Torgerson. *Theory and methods of scaling.* Wiley, 1958.
[56] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.
[57] U. von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
[58] U. von Luxburg and M. Alamgir. Density estimation from unweighted k-nearest neighbor graphs: a roadmap. In *Advances in Neural Information Processing Systems*, pages 225–233, 2013.
[59] U. von Luxburg and M. Kleindessner. Uniqueness of ordinal embedding. In *Proceedings of The 27th Conference on Learning Theory*, pages 40–67, 2014.
[60] F. Wauthier, M. Jordan, and N. Jojic. Efficient ranking from pairwise comparisons. In *Proceedings of the 30th International Conference on Machine Learning*, pages 109–117, 2013.
[61] K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*, volume 6, pages 1683–1686, 2006.
[62] Z. Zhu, A. M. C. So, and Y. Ye. Universal rigidity: Towards accurate and efficient localization of wireless networks. In *Proc. IEEE INFOCOM*, 2010.