SMILE: Robust Network Localization via Sparse and Low-Rank Matrix Decomposition

Lillian Clark
lilliamc@usc.edu
Electrical and Computer Engineering
University of Southern California
Los Angeles, California, USA

Sampad Mohanty
sbmohant@usc.edu
Computer Science
University of Southern California
Los Angeles, California, USA

Bhaskar Krishnamachari
bkrishna@usc.edu
Electrical and Computer Engineering
University of Southern California
Los Angeles, California, USA

ABSTRACT

Motivated by collaborative localization in robotic sensor networks, we consider the problem of large-scale network localization where location estimates are derived from inter-node radio signals. Well-established methods for network localization commonly assume that all radio links are line-of-sight and subject to Gaussian noise. However, the presence of obstacles which cause non-line-of-sight attenuation present distinct challenges. To enable robust network localization, we present Sparse Matrix Inference and Linear Embedding (SMILE), a novel approach which draws on both the well-known Locally Linear Embedding (LLE) algorithm and recent advances in sparse plus low-rank matrix decomposition. We demonstrate that our approach is robust to noisy signal propagation, severe attenuation due to non-line-of-sight, and missing pairwise measurements. Our experiments include simulated large-scale networks, an 11-node sensor network, and an 18-node network of mobile robots and static anchor radios in a GPS-denied limestone mine. Our findings indicate that SMILE outperforms classical multidimensional scaling (MDS) which ignores the effect of non-line of sight (NLOS), as well as outperforming state-of-the-art robust network localization algorithms that do account for NLOS attenuation including a graph convolutional network-based approach. We demonstrate that this improved accuracy is not at the cost of complexity, as SMILE sees reduced computation time for very large networks which is important for position estimation updates in a dynamic setting, e.g for mobile robots.

KEYWORDS

network localization, graph signal processing, low-rank matrix approximation

1 INTRODUCTION

Robotic sensor networks operating in GPS-denied environments can benefit from collaborative localization [26]. When distance measurements between network nodes (e.g. mobile robots or static beacons) are available, the *network localization* problem seeks to exactly recover the positions of each node in space. Intuitively, network localization algorithms leverage pairwise links to constrain the position estimate of every node in the network. If the locations of some nodes are known, these nodes are considered *anchors*. When sufficient conditions on the number of anchors and the graph induced by the distance measurements are met, we can recover the positions of all nodes [16].

However, in realistic scenarios the distance measurements are typically noisy. For example, radio signals can be used to estimate distance based on received signal strength, but signals are subject to noise from the wireless channel [31]. When this noise is assumed to be Gaussian with zero-mean[1], the redundancy offered by many links in a highly connected network serves to mitigate the problem. This explains why localization performance improves with the number of anchors used as reference points.

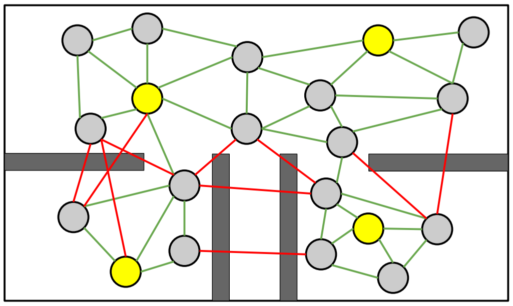

Figure 1: Illustration of network localization with NLOS. The objective is to accurately determine the localization of agents (grey) based on the known positions of anchors (yellow) given the constrains of noisy LOS links (green) and links with additional NLOS attenuation (red).

For many applications, including robotic exploration of unknown environments, signals see significant degradation from the presence of walls and obstacles. These non-line-of-sight (NLOS) links are difficult to model without an *a priori* map but have a significant affect on the relationship between received signal strength (RSS) and distance [10, 14]. Therefore, the ability to infer whether a pair of transmitters and receivers is NLOS can greatly improve network localization performance.

Additionally, when network nodes are spread over long distances, as may also be the case for robotic exploration, it is possible for the packets used to measure RSS to be dropped. When signal strength measurements are unavailable at long distances, our large-scale network localization approach must be robust to missing pairwise measurements.

Contribution: In this work, we examine the network localization problem in the presence of noisy signals where pairwise measurements may be NLOS or missing. We make two key observations: (1) the Euclidean distance matrix of the true node positions (in d dimensions) will have rank $d + 2$, which we show in Sec. 3, and (2) the matrix capturing the additional NLOS attenuation is commonly sparse and non-negative, i.e., there are a limited number

[1] Gaussian noise in dB is also referred to as log-normal fading.

Table 1: List of Abbreviations

Abbr.	Description
EDM	Euclidean Distance Matrix
LLE	Locally Linear Embedding
MDS	Multidimensional Scaling
NLOS	Non Line of Sight
PSVD	Partial SVD
RMSE	Root Mean Squared Error
SDP	Semi Definite Programming
SVD	Singular Value Decomposition

of walls which can only degrade signal strength. Given these observations, our approach draws on recent advances in sparse plus low-rank decomposition to first extract the positively biased NLOS attenuation, which we refer to as sparse matrix inference. After extracting the NLOS attenuation, we leverage a popular method for dimensionality reduction, locally linear embedding (LLE), to recover a set of weights which allow linear reconstruction to determine the exact positions of all nodes given the anchors' positions.

Specifically, we extend the discrete optimization approach for sparse matrix recovery which is presented in [3] to (1) handle missing pairwise measurements by imputation and (2) approximate the unknown sparsity structure of the sparse component of the matrix. Additionally, we provide details for solving for node positions using the LLE weight matrix directly (using least squares method) without the need for additional eigen-decomposition and coordinate frame alignment used conventionally. Our algorithm combines recent advances in sparse plus low rank decomposition and well established and interpretable algorithms like MDS and LLE and outperforms them in the face of NLOS attenuation.

Evaluation: We demonstrate that SMILE significantly improves localization accuracy over baseline methods which ignore the affect of NLOS attenuation. Further, we draw parallels between this approach and another method for graph signal processing which has recently been demonstrated as promising, namely Graph Convolutional Networks (GCNs) [36]. We compare the performance of SMILE and a state-of-the-art GCN implementation on large-scale simulated networks and demonstrate an improvement in localization accuracy and reduced computation time for network of more than 1000 nodes. Finally, we evaluate performance on two real-world networks: an outdoor wireless sensor network with 11 nodes, and 5 mobile robots and 13 static radios in a GPS-denied limestone mine with significant NLOS attenuation.

The paper is organized as follows. In Section 2 we provide a brief overview of seminal and recent work in network localization, and in Section 3 we formally define the problem and notation. SMILE is introduced and explained in detail in Section 4, including sparse matrix inference and subsequent steps for position estimation. We also draw parallels between our approach and the existing graph learning-based method. In Section 5 we provide implementation details as well as experimental results on real and simulated data. We give concluding remarks in Section 6.

2 BACKGROUND AND RELATED WORK

Network localization: Network localization is well-researched, and is commonly formulated as a least squares problem [2, 9, 24]. Multidimensional scaling (MDS) and its extensions are popular methods for solving this problem [29]. Classical MDS uses the distance matrix to compute a matrix of scalar products, typically called the Gram matrix, that captures pairwise correlation of the position vectors. The principal components from eigen-decomposition of this matrix are then used to recover relative node positions [1]. Rather than compute over the entire distance matrix, Locally Linear Embedding (LLE) [27, 30] applies principal component analysis to small neighborhoods, which improves performance when the reduction from the noisy (high-dimensional) data to the low-dimensional true positions is non-linear, and has shown promise in sensor network localization [18].

Exploiting sparsity: If the data is well-described by a particular statistical model (e.g. Gaussian or log-normal), we can instead form the maximum likelihood estimation problem [25], and solve using semi-definite programming (SDP) methods [4, 5]. Recent works extend SDP methods to consider non-Gaussian noise [38] and, more specifically, NLOS noise [8, 19, 22]. Jin *et al.* demonstrate that when the NLOS noise has a certain structure, namely non-negative and sparse, a sparsity-promoting term in the objective function can improve the performance of SDP approaches [19]. However, SDP methods suffer with respect to complexity and are intractable for very large networks [36].

Sparse and low rank decomposition: Another approach is to recover the matrix of NLOS attenuation directly. In recent years, sparse and low-rank matrix recovery has drawn attention due to its relevance in signal processing, statistics, and machine learning [34]. Bertsimas *et al.* recently proposed a discrete optimization approach to sparse and low-rank recovery which uses alternating minimization [3]. Our work leverages this method, extends it to the case of missing measurements and unknown sparsity, and demonstrates that it can serve as an important component of a robust network localization algorithm.

Graph convolutional networks: To effectively exploit the relational information of graph-structured data, graph neural networks (GNNs) have recently become a popular method for approaching optimization problems in wireless networks [21]. Yan *et al.* recently demonstrated promising results in the application of Graph Convolutional Networks (GCNs) to the network localization problem. Their approach maintains accurate localization despite NLOS attenuation, and is scalable to large-scale networks at an affordable computation cost. However, we will see that the learned model is unable to exactly recover positions for a completely observed distance matrix in the absence of noise. In this work we propose a novel network localization algorithm which builds on the principles of multidimensional scaling for exact recovery in the absence of noise, exploits the sparsity of NLOS attenuation for improved localization accuracy, and scales to very large networks at an affordable computation cost.

3 PROBLEM FORMULATION

Let A be the set of anchors whose positions are known, where $|A| = n_A$. Let B be the set of agents whose positions are unknown, where $|B| = n_B$. Let $C = A \cup B$ be the set of nodes which includes both anchors and agents, with $n_A + n_B = n$. We define the $(n \times n)$ true distance matrix $\mathbf{D}$ as

$$\mathbf{D}[i,j] = ||\mathbf{p}_i - \mathbf{p}_j||$$

where $\mathbf{p}_i = [x_i, y_i]^T$ is the position of node i. We establish the convention that the first n_A rows and n_A columns pertain to the anchors. $\mathbf{D}^{\circ 2}$ is the Euclidean distance matrix (EDM, containing squared distances i.e $\mathbf{D}$ squared entry-wise), which is zero along the diagonal and symmetric [17]. If matrix $\mathbf{P} \in \mathbb{R}^{n \times 2}$ with $\mathbf{P}[i,:] = \mathbf{p}_i^T$ be the position matrix, we can see that $\mathbf{D}^{\circ 2}$ has rank 4. In general, the EDM of a configuration of points embedded in $\mathbb{R}^d$ has rank at most $d + 2$ (and exactly $d + 2$ if the points are in general position as opposed to more special or coincidental cases, e.g., points in 3D which lie on a line). This is briefly justified below, with further details provided in [13].

$$\begin{aligned}
\mathbf{D}[i,j]^2 &= ||\mathbf{p}_i - \mathbf{p}_j||^2 \\
&= \langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j \rangle \\
&= \langle \mathbf{p}_i, \mathbf{p}_i \rangle + \langle \mathbf{p}_j, \mathbf{p}_j \rangle - 2\langle \mathbf{p}_i, \mathbf{p}_j \rangle \\
&= \mathbf{p}_i^T \mathbf{p}_i + \mathbf{p}_j^T \mathbf{p}_j - 2\mathbf{p}_i^T \mathbf{p}_j \\
\implies \mathbf{D}^{\circ 2} &= \underbrace{\mathbf{diag}(\mathbf{P}\mathbf{P}^\mathbf{T})\mathbf{1}^\mathbf{T}}_{rank-1} + \underbrace{\mathbf{1}\mathbf{diag}(\mathbf{P}\mathbf{P}^\mathbf{T})^T}_{rank-1} - \underbrace{2\mathbf{P}\mathbf{P}^\mathbf{T}}_{rank-2}
\end{aligned}$$

Since $rank(\mathbf{A}{+}\mathbf{B}) \leq rank(\mathbf{A}){+}rank(\mathbf{B})$ and $rank(\mathbf{C}\mathbf{C}^\mathbf{T}) = rank(\mathbf{C}^\mathbf{T}\mathbf{C}) = rank(\mathbf{C})$, we have $\mathbf{D}^{\circ 2}$ as low-rank with rank = 4.

Let $\mathbf{O}$ be the matrix that contains the distances that we observe, where $\mathbf{O}[i,j]$ is a function of the strength of radio signal transmitted by node i and received by node j. In general, our observation can be captured as

$$\mathbf{O} = \mathbf{\Omega} \circ [\mathbf{D} + \mathbf{N} + \mathbf{S}]$$

where $\mathbf{\Omega}$ is the observation mask and takes on values of 1 when a measurement is available and 0 otherwise (for instance, when the transmitter and receiver are out of range). $\mathbf{N}$ is an asymmetric matrix capturing noise in the observations, which we assume is Gaussian and zero-mean. $\mathbf{S}$ captures the additional NLOS attenuation, and as in [19] we assume $\mathbf{S}$ is non-negative and sparse. We also assume $\mathbf{S}$ is symmetric which relies on assumptions that an attenuating obstacle will affect a radio link in both directions equally.

We make the realistic assumption that nodes are in general position, meaning that in 2D they do not lie on a straight line. We also assume that an upper bound on the distance between any two nodes, $d_{\max}$, is known or can be approximated. Our objective is finding an estimate for the locations of all nodes $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1, ..., \hat{\mathbf{p}}_n]^T$ which is consistent with the prior information - (1) the observations in $\mathbf{O}$ as well as (2) the known anchor positions $\mathbf{P_A} = [\mathbf{p}_1, ..., \mathbf{p}_{n_A}]^T$.

4 SMILE

In this section we propose Sparse Matrix Inference and Linear Embedding, our novel large-scale network localization algorithm, which is illustrated in Fig. 2. Details of the algorithm are presented in Algorithm 1. In subsection 4.1, we discuss our method of extracting NLOS attenuation via sparse matrix inference, which results in a low-rank approximation of the Euclidean distance matrix. Then in subsection 4.2, we discuss our method of transforming this matrix into an estimate of the locations of all nodes.

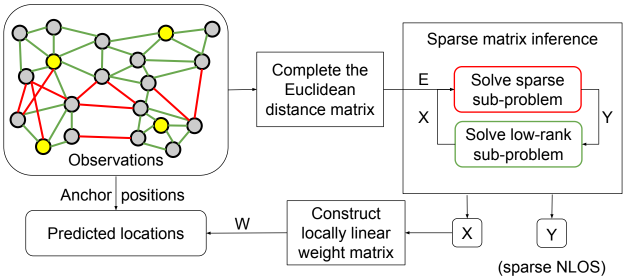

Figure 2: An overview of Sparse Matrix Inference and Linear Embedding (SMILE).

4.1 Sparse Matrix Inference

For a given input matrix $\mathbf{E} \in \mathbb{R}^{n \times n}$, for which $\mathbf{E} = \mathbf{X} + \mathbf{Y}$, sparse matrix inference seeks to find the low-rank component $\mathbf{X} \in \mathbb{R}^{n \times n}$ and sparse component $\mathbf{Y} \in \mathbb{R}^{n \times n}$ which solves:

$$\begin{aligned}
\min_{\mathbf{X},\mathbf{Y}} f(\mathbf{X},\mathbf{Y}) &= ||\mathbf{E} - \mathbf{X} - \mathbf{Y}||_F^2 + \lambda||\mathbf{X}||_F^2 + \mu||\mathbf{Y}||_F^2 \\
\text{s.t.} \quad & rank(\mathbf{X}) \leq \alpha, ||\mathbf{Y}||_0 \leq \beta
\end{aligned} \tag{1}$$

where $||\mathbf{X}||_F$ denotes the Frobenius norm. In this subsection we describe how to decompose $\mathbf{E}$ $\mathbf{X}$ and $\mathbf{Y}$ using alternating minimization, and how to use this technique on the problem defined in Sec. 3.

Firstly, we square the observed matrix (line 2). Temporarily assuming $\mathbf{N} = \mathbf{0}$ and no missing observations, i.e $\mathbf{\Omega} = \mathbf{1}\mathbf{1}^\mathbf{T}$,

$$\begin{aligned}
\mathbf{E} = \mathbf{O}^{\circ 2} &= (\mathbf{D} + \mathbf{S})^{\circ 2} \\
&= \underbrace{\mathbf{D}^{\circ 2}}_{\text{low-rank}} + \underbrace{\mathbf{S}^{\circ 2} + 2\mathbf{D} \circ \mathbf{S}}_{\text{sparse}}
\end{aligned} \tag{2}$$

where the expression in parentheses is non-negative and sparse (because $\mathbf{S}$ is non-negative and sparse), and $\mathbf{D}^{\circ 2}$ is low-rank. Thus the problem is amenable to the sparse matrix inference framework.

Let us consider the case with noise $N \neq 0$ and no missing observations

$$\begin{aligned}
\mathbf{E} = \mathbf{O}^{\circ 2} &= (\overbrace{\mathbf{D} + \mathbf{N}}^{\tilde{\mathbf{D}}} + \mathbf{S})^{\circ 2} = (\tilde{\mathbf{D}} + \mathbf{S})^{\circ 2} \\
&= \tilde{\mathbf{D}}^{\circ 2} + \underbrace{\mathbf{S}^{\circ 2} + \tilde{\mathbf{D}} \circ \mathbf{S}}_{\text{sparse}}
\end{aligned} \tag{3}$$

The last expression is exactly the same as the formulation in 2 except that $\tilde{\mathbf{D}}^{\circ 2} = \mathbf{D}^{\circ 2} + \mathbf{N}^{\circ 2} + 2\mathbf{N} \circ \mathbf{D}$ may no longer be low rank due to the addition of $\mathbf{N}^{\circ 2} + 2\mathbf{N} \circ \mathbf{D}$ where entries of the first term are now i.i.d from a Chi-Squared Distribution $\mathcal{X}_1^2$. However, in our ablation study, we empirically verify that as long as the standard deviation of the normal noise in entries of $\mathbf{N}$ is not too large, using the estimated low-rank matrix from the sparse matrix plus low-rank inference, SMILE is able to recover $\mathbf{P}$ fairly faithfully under RMSE loss.

Table 2: SMILE parameters

Parameter	Symbol
k	Number of neighbors
$\hat{\beta}_i$	Initial sparsity estimate
η	Step size
T	Number of random initializations
λ	Low-rank matrix regularizer
μ	Sparse matrix regularlizer
ϵ	Inner loop tolerance
ϵ_β	Outer loop tolerance

Missing measurements: The approach presented in [3] assumes a complete input matrix. While more complex approaches to matrix completion exist, for example by finding the sum total length of the shortest path between two nodes [32, 33], we take a simpler approach. When $\Omega \neq \mathbf{1}\mathbf{1}^\mathbf{T}$, we complete the observation matrix by filling in missing values with $d_{\max}$ (line 1). Even in the face of missing observations, we show that the problem remains amenable to the sparse matrix inference framework as long as the missing observations are not too large of a fraction of the total number of observations in $\mathbf{O}$.

$$
\begin{aligned}
\mathbf{E} = \tilde{\mathbf{O}}^{\circ 2} &= [\Omega \circ \mathbf{O}]^{\circ 2} \\
&= [\Omega \circ (\mathbf{D} + \mathbf{N} + \mathbf{S}) + (\mathbf{1} - \Omega)d_{max}]^{\circ 2} \\
&= [\Omega \circ \mathbf{D} + \Omega \circ (\mathbf{N} + \mathbf{S}) + (\mathbf{1} - \Omega)d_{max}]^{\circ 2} \\
&= [\mathbf{D} - (\mathbf{1} - \Omega) \circ \mathbf{D} + \Omega \circ (\mathbf{N} + \mathbf{S}) + (\mathbf{1} - \Omega)d_{max}]^{\circ 2} \\
&= [\mathbf{D} + \Omega \circ \mathbf{N} + \underbrace{\Omega \circ \mathbf{S} + (\mathbf{1} - \Omega) \circ (d_{max}\mathbf{I} - \mathbf{D})}_{\tilde{S}\text{ (sparse)}}]^{\circ 2} \qquad (4) \\
&= [\underbrace{\mathbf{D} + \Omega \circ \mathbf{N}}_{\tilde{\mathbf{D}}} + \tilde{\mathbf{S}}]^{\circ 2} = [\tilde{\mathbf{D}} + \tilde{\mathbf{S}}]^{\circ 2} \\
&= \tilde{\mathbf{D}}^{\circ 2} + \underbrace{\tilde{\mathbf{S}}^{\circ 2} + 2\tilde{\mathbf{D}} \circ \tilde{\mathbf{S}}}_{\text{sparse}}
\end{aligned}
$$

Thus, $\tilde{\mathbf{O}}$ decomposes similarly to $\mathbf{O}$ as long as $(\mathbf{1} - \Omega) + \mathbf{S}$ is still sparse, i.e. the total number of NLOS and missing measurements is low. $\tilde{\mathbf{D}}^{\circ 2}$ is still amenable to SMILE like in the previous case.

Alternating minimization: As presented in [3], we alternate between solving two sub-problems. For a given $\mathbf{X}$, we estimate $\mathbf{Y}$ by composing a sparse matrix with non-zero entries at the largest indices of $(\mathbf{E} - \mathbf{X})$. This is described further in Algorithm 2. Then for a given $\mathbf{Y}$, we estimate $\mathbf{X}$ by reducing the rank of $(\mathbf{E} - \mathbf{Y})$. This repeats until the value of the objective function in Eq. 1 has converged, corresponding to the inner loop (lines 11-18). We initialize $\mathbf{X}$ with a random, low-rank matrix and repeat this process for $T > 0$ random initialization, ultimately selecting the decomposition which minimizes the objective function (lines 5-19).

Unknown sparsity: The approach presented in [3] assumes the sparsity of $\mathbf{Y}$ is known, however in realistic settings this may not be true. We observed empirically that for a matrix with known components, solving the sparse matrix inference problem for increasing estimate $\hat{\beta}$ causes the objective function to decrease, and around the true value of β it converges (Fig. 3. This motivates the outer loop (lines 3-21). To reduce computation time, we can initialize $\hat{\beta}$ with the approximate sparsity, increase the search step size η, or increase the converge tolerance ϵ_β. However this may come at the cost of localization accuracy.

Algorithm 1 SMILE

```
Input: O (observation matrix), P_A (anchor positions), α (desired rank)
Output: $\hat{\mathbf{P}}_\mathbf{B}$ (agent location estimates)
 1: $\tilde{\mathbf{O}} = \mathbf{O} + (1 + \Omega)d_{\max}$
 2: $\mathbf{E} = \tilde{\mathbf{O}}^{\circ 2}$
 3: $\hat{\beta} \leftarrow \hat{\beta}_i$
 4: while $\Delta f / f > \epsilon_\beta$ do
 5:     for $t$ in $\{1, ..., T\}$ do
 6:         $\mathbf{X}' \in \mathbb{R}^{n \times n} \leftarrow$ random
 7:         $\mathbf{U}, \Sigma, \mathbf{V}^T = \text{PSVD}(\mathbf{X}', \alpha)$
 8:         $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$
 9:         $\mathbf{Y} \in \mathbb{R}^{n \times n} \leftarrow \mathbf{0}$
10:         $f = ||\mathbf{E} - \mathbf{X} - \mathbf{Y}||_F^2 + \lambda||\mathbf{X}||_F^2 + \mu||\mathbf{Y}||_F^2$
11:         while $\Delta f / f > \epsilon$ do
12:             $\mathbf{Y}' = \text{compose\_sparse}(\mathbf{E} - \mathbf{X}, \hat{\beta})$
13:             $\mathbf{Y} = \frac{1}{1+\mu}\mathbf{Y}' \circ (\mathbf{E} - \mathbf{X})$
14:             $\mathbf{X}' = \frac{1}{1+\lambda}(\mathbf{E} - \mathbf{Y})$
15:             $\mathbf{U}, \Sigma, \mathbf{V}^T = \text{PSVD}(\mathbf{X}', \alpha)$
16:             $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$
17:             $f = ||\mathbf{E} - \mathbf{X} - \mathbf{Y}||_F^2 + \lambda||\mathbf{X}||_F^2 + \mu||\mathbf{Y}||_F^2$
18:         end while
19:     end for
20:     $\hat{\beta} \leftarrow \hat{\beta} + \eta$
21: end while
22: $\mathbf{W} \in \mathbb{R}^{n \times n} \leftarrow \mathbf{0}$
23: for each node $i$ do
24:     find $k$ nearest neighbors $NN(i)$
25:     for each pair of neighbors $(j, l)$ do
26:         $\mathbf{H} \in \mathbb{R}^{k \times k}$ where $\mathbf{H}[j, l] = \frac{1}{2}(\mathbf{X}[i, l] + \mathbf{X}[j, i] - \mathbf{X}[j, l])$
27:         solve $\mathbf{H}\mathbf{w} = 1$ for $\mathbf{w}$
28:     end for
29:     $\mathbf{W}_i \leftarrow \mathbf{w} / \sum \mathbf{w}$ at indices of neighbors
30: end for
31: $\mathbf{m} = (\mathbf{I} - \mathbf{W})_A \mathbf{P}_A$
32: solve $(\mathbf{W} - \mathbf{I})_B \hat{\mathbf{P}}_B = \mathbf{m}$ for $\hat{\mathbf{P}}_B$
33: return $\hat{\mathbf{P}}_\mathbf{B}$
```

Algorithm 2 compose_sparse

```
Input: $\mathbf{M} \in \mathbb{R}^{n \times n}$ (matrix), $\beta$ (sparsity)
Output: B (binary matrix)
1: sorted indices = argsort(M)
2: one indices = sorted indices[-$\beta$:]
3: $\mathbf{B} \in \mathbb{R}^{n \times n} \leftarrow \mathbf{0}$
4: B[one indices] = 1
5: return B
```

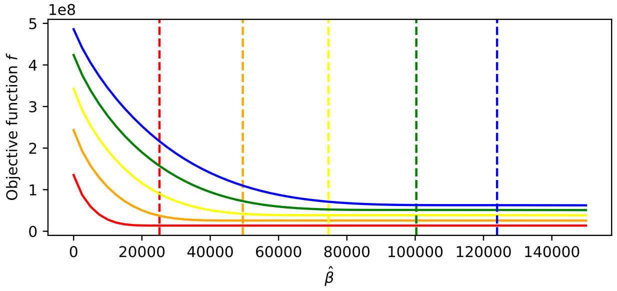

Figure 3: The final value of the sparse matrix inference objective function f (Eq. 1) when we assume different levels of sparsity $\hat{\beta}$, plotted alongside the true sparsity β (dotted line) for various simulated matrices $\mathbf{E} \in \mathbb{R}^{500\times500}$.

4.2 Linear Embedding

Next we find the set of k nearest neighbors $NN(i)$ for each node i using the estimated EDM gotten from sparse matrix inference. For each node, we compute a $k \times k$ matrix $\mathbf{H}$ which captures the pairwise similarity between the nodes in $NN(i) \cup \{i\}$ (line 25) using the corresponding sub-matrix in the estimated EDM. $\mathbf{H}$ is calculated using the same approach used for classical MDS, but for local neighborhoods [1]. From $\mathbf{H}$ we can recover the desired weight matrix $\mathbf{W}$ (lines 26-27), first by solving for $\mathbf{w}$ and then inserting it at the row position corresponding to the node index of i with the entries in $\mathbf{w}$ aligned with the column belonging to the nearest neighbours and zeros for non-neighbours. At this point, each row of $\mathbf{W}$ corresponds to a node and captures how the node's position can be expressed as a linear combination of its neighbors. This is a meaningful representation of the positions of all nodes, but is not yet a node embedding. Typically, LLE then computes a sparse matrix $(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$ whose eigenvectors corresponding to the two smallest non-zero eigenvalues result in a solution up to rotation and translation.

Using the anchors' positions: In this setting, we can compute a solution using $\mathbf{W}$ and the anchor location $\mathbf{P}_A$ directly. This is possible because, by construction, $\mathbf{WP} = \mathbf{P}$. This is essentially a system of n equations with $n_B < n$ unknowns. Treating $\mathbf{W}-\mathbf{I} \in \mathbb{R}^{n\times n}$ as a block matrix, let $(\mathbf{I}-\mathbf{W})_A = (\mathbf{I}-\mathbf{W})[:, : \mathbf{n_A}] \in \mathbb{R}^{n\times n_A}$ be the sub-matrix corresponding to the anchors and $(\mathbf{W}-\mathbf{I})_B = (\mathbf{I}-\mathbf{W})[:, -\mathbf{n_B} :] \in \mathbb{R}^{n\times n_B}$ be the sub-matrix corresponding to the agents, where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. Similarly treating $P \in \mathbb{R}^{n\times 2}$ as a block matrix, let $P_A = P[: n_A, :] \in \mathbb{R}^{n_A\times 2}$ and $P_B = P[-n_B, :] \in \mathbb{R}^{n_B\times 2}$ correspond to the block matrices for positions of anchors and agents respectively. With some manipulation we get,

$$\begin{aligned}
\mathbf{WP} = \mathbf{P} \implies \overbrace{(\mathbf{W}-\mathbf{I})}^{T}\mathbf{P} = \mathbf{0} &\implies \mathbf{TP} = \mathbf{0} \\
&\implies \begin{bmatrix}\mathbf{T_A} & | & \mathbf{T_B}\end{bmatrix}\begin{bmatrix}\mathbf{P_A} \\ - \\ \mathbf{P_B}\end{bmatrix} = \mathbf{0} \\
\implies \mathbf{T_A P_A} + \mathbf{T_B P_B} = \mathbf{0} &\implies \mathbf{T_A P_A} = -\mathbf{T_B P_B} \\
&\implies \underbrace{(\mathbf{I}-\mathbf{W})_A \mathbf{P}_A}_{\mathbf{m},\ \text{known}} = \underbrace{(\mathbf{W}-\mathbf{I})_B}_{\text{known}}\ \underbrace{\mathbf{P}_B}_{\text{unknown}}
\end{aligned} \tag{5}$$

(lines 29-30) and from the last expression we can estimate a solution $\hat{\mathbf{P}}_\mathbf{B}$ for the unknown agent positions using the least squares minimization.

4.3 Comparison with Graph Convolutional Networks

We highlight several interesting parallels between SMILE and GCN. Briefly, the approach presented by Yan *et al.* computes the position estimates

$$\hat{\mathbf{P}} = \mathbf{A}'\phi(\mathbf{A}'(\mathbf{A}\odot\mathbf{O})\mathbf{Z}^{(1)})\mathbf{Z}^{(2)} \tag{6}$$

where $\mathbf{A}$ is the thresholded adjacency matrix for a given threshold θ_{GCN}, $\mathbf{A}'$ is the row-normalized augmented thresholded adjacency matrix, $\phi(\cdot)$ is a nonlinear activation function, and $\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}$ are learned weights. Note that the graph signal is the observed matrix.

1. θ_{GCN} **and** k**:** This approach introduces a threshold such that edges between nodes in the graph are present only if the observed distance is less than the threshold. Decreasing this threshold is similar to decreasing the number of nearest neighbors, and both have the benefit of noise truncation. However, as the threshold is increased, GCN experiences over-smoothing due to the aggregation step. Intuitively, if every node is connected to every other node, the aggregation step causes all node embedding to collapse at a single point. This means that more information, i.e., additional pairwise measurements, actually hurts the algorithm. SMILE does not experience this issue, and we will see in Fig. 6 that increasing k improves performance at the cost of increased runtime.
2. **Low-pass filtering and PSVD:** Repeated multiplication by the normalized adjacency matrix acts as a low pass filter [23]. This reduces the rank of the graph signal, similar to the process of extracting the low-rank component of the noisy euclidean distance matrix. However, repeated multiplication by the normalized adjacency matrix reduces the rank of the observed matrix by some amount, which corresponds to distances (rather than squared distances). Our approach applies rank reduction to exactly reduce the Euclidean distance matrix to rank $d+2$, informed by the principles of the problem formulation.
3. **Convolution and linear embedding:** Each convolution layer multiplies the adjacency matrix and input matrix by a set of weights. Intuitively, this makes the node features at the next layer a weighted sum of the neighbor features. The learned GCN weights are analogous to the set of weights $\mathbf{W}$ in SMILE which allow linear reconstruction of agents from

the anchors positions. However, the repeated convolutions and nonlinear activation prevent a straightforward analysis. The interpretability of $\mathbf{W}$ is an advantage of our approach.

Qualitatively, we expect SMILE to outperform GCN because (1) it is not subject to oversmoothing and makes careful and productive use of additional pairwise measurements, (2) it finds a Euclidean distance matrix with the exact expected rank, and (3) it relies on principled methods to determine a weight matrix relating nodes to their neighbors, which is thus interpretable. In the next section, we compare these approaches quantitatively.

5 RESULTS

In this section we evaluate the performance of SMILE with respect to localization accuracy and computation time. We also consider performance under different noise settings, as an ideal method is both robust to high-levels noise and able to exactly recover positions when possible. Localization accuracy is measured by the root-mean-squared-error (RMSE) given by $||\mathbf{P}_B - \hat{\mathbf{P}}_B||_F$. Robustness considers the affect of Guassian noise, NLOS attenuation, and missing pairwise measurements. The experiments consider networks in 2D, thus the desired rank of EDM is $\alpha = 4$. The SMILE parameters are set to $k = 50, \hat{\beta}_i = 5\frac{n^2}{100}, \eta = \frac{n^2}{100}, T = 1, \lambda = 0.01, \mu = 0.1, \epsilon = 0.001$, and $\epsilon_\beta = 0.01$. For comparison, we train a two-layer GCN according to [36] with distance threshold 1.2. More details of our implementation are available online [2].

5.1 Simulation results

Our simulated scenarios consider n nodes randomly placed over a 5m × 5m square area, with the first n_A nodes considered anchors. Noise $\mathbf{N}$ is drawn from a zero-mean Gaussian, $\mathbf{N}[i,j] \sim \mathcal{N}(0,\sigma^2)$. NLOS noise is drawn from a uniform distribution, $\mathbf{S}[i,j] = \mathbf{S}[j,i] \sim \mathcal{U}[0,10]$ with probability p_{NLOS} and 0 otherwise. Both matrices are zero along the diagonal, $\mathbf{N}[i,i] = \mathbf{S}[i,i] = 0$. The observation mask limits our observed distance measurements by a threshold θ such that $\Omega[i,j] = 1$ if $\mathbf{O}[i,j] \leq \theta$ and 0 otherwise.

Comparison with state of the art: Firstly we consider the setting where $n = 500, n_A = 50, \sigma = 0.1, p_{\text{NLOS}} = \frac{1}{10}$, and $\theta = d_{\max}$, for which an example dataset is available[3]. Fig. 4 illustrates the performance of SMILE, which achieves an RMSE of 0.06 in 5.22 seconds, and GCN which achieves an RMSE of 0.11 in 5.73 seconds. GCN's performance is consistent with that reported in [36], which reports RMSE for various other methods, including SDP with sparsity promoting regularization [19] which achieves an RMSE of 0.26. SMILE achieves the highest reported localization accuracy, and Fig. 5 illustrates that not only is the error low on average but the error density has a smaller tail.

In Fig. 6, we investigate the localization accuracy and computation time as we vary k, the number of neighbors used for linear embedding. We observe that RMSE remains consistent for $n_A \geq 50$. While the minimum of 0.048 is reached at $k = 130$, this comes at the expense of increased computation time. Note that selecting k is analogous to setting the GCN threshold, however we do not observe the degradation in performance that GCN is prone to when the threshold is too high. This comes from the aggregation component of convolution, which causes over-smoothing if a node has too many neighbors and results in the position estimates collapsing to a point.

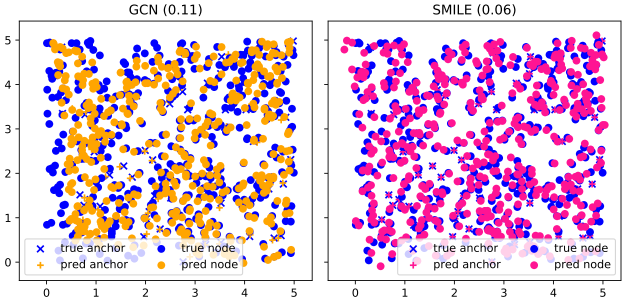

Figure 4: Localization with GCN (Yan et al. [36]) and novel SMILE for a 500 node network with 50 anchors. GCN achieves RMSE 0.11 and SMILE achieves RMSE 0.06.

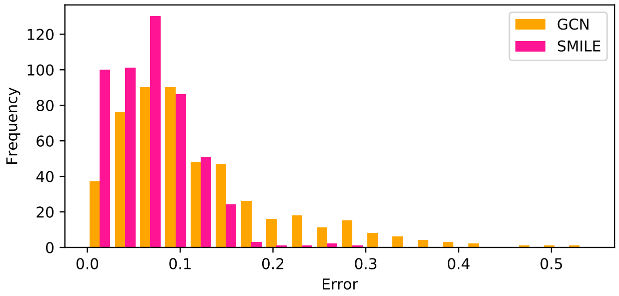

Figure 5: Error density for GCN and SMILE on the data from Fig. 4, where SMILE results in a more desirable distribution.

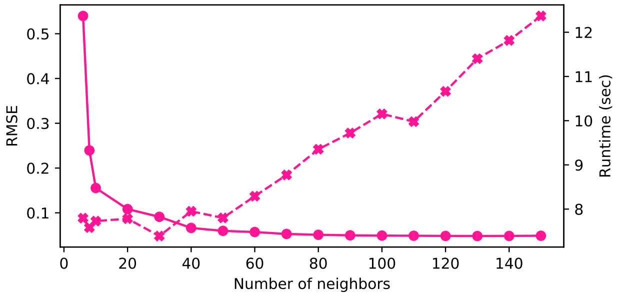

Figure 6: RMSE (solid line) and runtime (dashed line) trade-off as we vary the number of neighbors k. Each point is the average of 10 trials.

Robustness: Fig. 7 demonstrates the performance of SMILE and GCN as p_{NLOS} varies from 0 to $\frac{1}{2}$. From this, we observe that SMILE outperforms GCN for up to 30% NLOS links. Beyond this, the assumption that $\mathbf{S}$ is sparse is no longer true, and performance suffers as expected. The same is true when θ falls below 3.5, and $(1 - \Omega)$ is no longer sparse, i.e not enough entries in the observations matrix

[2] https://github.com/ANRGUSC/smile-network-localization
[3] https://github.com/Yanzongzi/GNN-For-localization

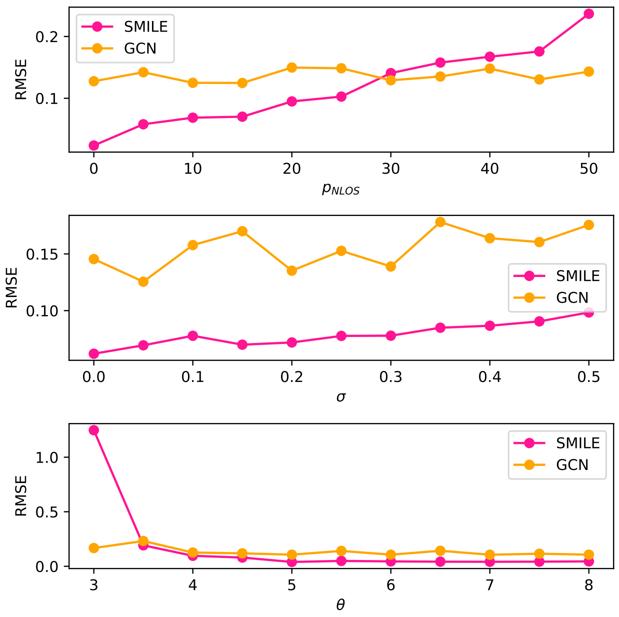

Figure 7: Performance of SMILE and GCN as the probability of NLOS links, standard deviation of Gaussian noise, and threshold for distances which can be measured are varied ($T = 2$).

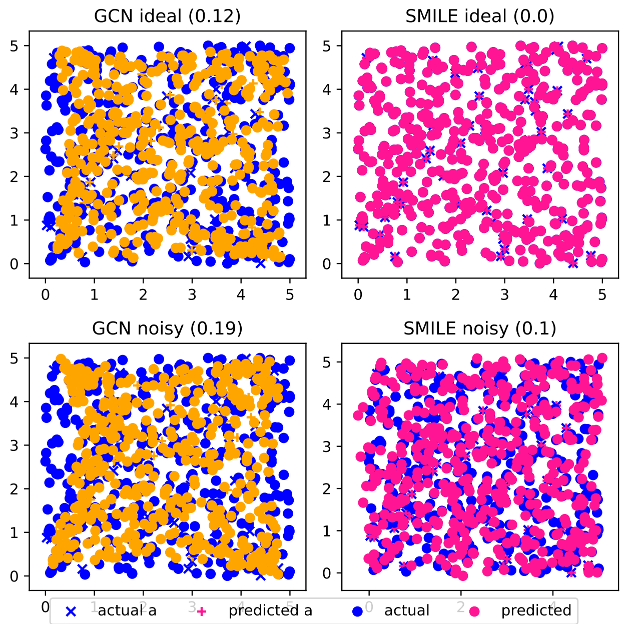

Figure 8: Performance of SMILE and GCN in an ideal setting ($p_{\text{NLOS}} = 0, \sigma = 0$) and a noisy setting ($p_{\text{NLOS}} = \frac{1}{10}, \sigma = 0.5$). SMILE is robust to noise without sacrifice performance in ideal settings.

$\tilde{\mathbf{O}}$. Notably, we do not observe a clear trend in the performance of GCN as σ increases. We posit that this is because the learned approach does not rely on exact decomposition, even in the absence of noise.

To test this, we compare the performance of GCN and SMILE on an ideal dataset and consider the results in Fig. 8. While the performance of both methods is better in this ideal setting, we observe that GCN cannot exactly recover positions. While being robust to Gaussian noise and sparse non-Gaussian noise, SMILE is also accurate in ideal scenarios.

Complexity: As we increase the number of nodes, accuracy of both methods increases (accuracy similarly increases with the percentage of anchors). However, complexity increases with the size of the network. Because our approach is iterative, we measure complexity numerically (computation time) in lieu of analytically as it is difficult to predict when the sparse matrix inference will converge. Fig. 9 shows that the computation time of both SMILE and GCN remain reasonable as n increases. For least squares optimization and SDP, we have seen that the compute time for very large networks becomes intractable [36]. For the 1000 node network, the time to predict $\hat{\mathbf{P}}$ using SMILE is 29.84 seconds while the time to train and predict $\hat{\mathbf{P}}$ using GCN is 46.90 seconds. Note that this is a fair comparison because the learned approach requires training a specific model for each network; the same learned weights are not applicable to a new network, and neither is the SMILE weight matrix $\mathbf{H}$. The low compute times for SMILE are likely because the linear embedding component depends on the number of neighbors (rather than the number of nodes) and the sparse matrix inference component adapts the step size to the number of nodes. We highlight that scalability is not at the cost of localization accuracy. For $n = 1500$, SMILE achieves RMSE of 0.05 while GCN achieves RMSE of 0.08.

Ablation: Thus far we have compared our approach to an existing solution, but it is also interesting to consider the role of each component of SMILE and, in particular, whether simpler existing approaches from the literature are sufficient. Fig. 10 considers a dataset with $\sigma = 0.3, p_{\text{NLOS}} = \frac{1}{10}$, and $\theta = 5$. We compare the performance of (A) no rank reduction, (B) rank reduction via PSVD, and (C) sparse and low-rank matrix recovery. In conjunction, we consider (1) classical MDS with the Kabsch algorithm for coordinate system registration [1] and (2) LLE-based embedding using anchor positions. Specifically, the top left plot represents a naive baseline: multidimensional scaling assuming Gaussian zero-mean noise. The bottom right plot represents SMILE.

We observe several key takeaways from this figure.

(1) **De-noising:** Moving from left to right, each column of this plot contains increasing more sophisticated noise reduction and decreases the final RMSE. In particular, sparse matrix inference increases localization accuracy by almost an order of magnitude.
(2) **Nearest neighbors:** Methods in the top row use all available links in eigen-decomposition, while methods in the bottom row use only the nearest neighbors to determine weights for linear reconstruction. This local-neighborhood approach consistently improves localization accuracy. Note

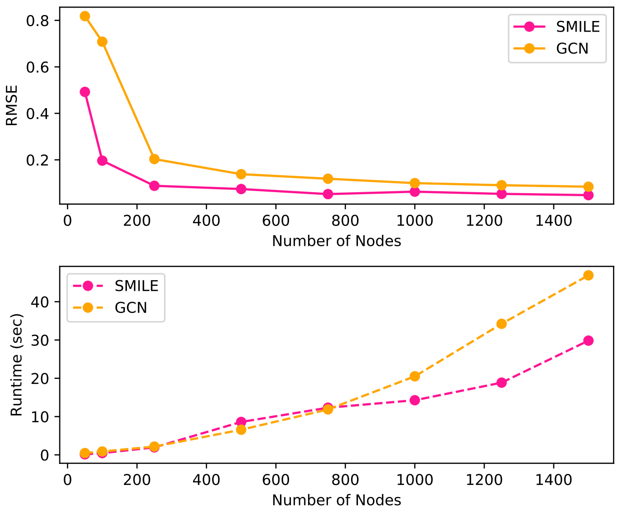

Figure 9: Compute time for different sizes of networks for GCN and SMILE. Previous work has shown that learned approaches (GCN, MLP, NTK) scale better than optimization approaches (LS, ECM, SDP) [36]. SMILE outperforms GCN for very large networks.

that if $\theta \geq d_{\max}$, i.e. no measurements are missing, the performance of (1) and (2) are similar. This is likely because LLE's strength comes from relying more on nearby measurements, and our approach to matrix completion puts missing measurements at effectively long distances. In setting where long-distance measurements are unreliable, which is commonly the case in realistic wireless communication [31], using nearest neighbors is advantageous.

(3) **SMILE:** The combination of both sparse matrix inference and linear embedding is more robust to NLOS attenuation and missing measurements than either component in isolation.

5.2 Experimental results

In this section we apply SMILE to two real-world small scale datasets, and discuss its performance, our findings, and directions for future work.

Sensor network: First we consider a network of 11 Mica2 motes placed randomly in a parking lot [37] [4]. The maximum distance between any two nodes is 10.57m. We consider received signal strength (RSS) measurements between pairs (RSS is averaged over 20 packets). To estimate distance, we use the following log-distance path loss model

$$RSS(d) = P_{\text{tx}} - PL(d_0) - 10\gamma \log_{10}(\frac{d}{d_0}) + \mathcal{N}(0, \sigma^2) \qquad (7)$$

where P_{tx} is the transmit power, $PL(d_0)$ is the path loss at a reference distance, and γ is the path loss exponent [31]. For this data, use the model $\gamma = 2.9$ and $P_{\text{tx}} - PL(d_0) = -49.12dB$ for $d_0 = 1$. Given the ground truth location estimates, Fig. 11 shows that the noise in this dataset has mean 0.35m with standard deviation 2.94. Note that even though we believe this data to be in open space (all LOS), seven links see a measurement error of greater than 5m.

Figure 12 shows the performance of GCN and SMILE on this network, where we update the SMILE parameters to $\lambda = 0.05$, $\mu = 0.05$, and $T = 10$ because the network is smaller. If we assume four of these nodes are anchors, GCN achieves RMSE of 1.67 with threshold 5.2m, and SMILE achieves a comparable RMSE of 1.63 with $k = 3$, guessing $\beta = 11$. These results are the average performance of 10 trials, and the best parameters for each algorithm were selected empirically.

Robotic network: Secondly, we consider measurements from a robotic network with 18 nodes, 13 of which are stationary beacons at known locations, and 5 of which are mobile (legged and wheeled) robots[5]. Each robot carries a Streamcaster 4200 from Silvus Technologies, which are also used as beacons. We consider a set of pairwise RSS measurements from a single timestamp of robotic exploration. They nodes are spread over a large area in an underground limestone mine with distances between nodes of up to 341 meters. Ground truth positions are available via a simultaneous localization and mapping algorithm, and we assume these are accurate [7]. Estimates of whether links are LOS is also available via a LiDAR-based predictive model [10]. Fig. 13 plots the error density for LOS, NLOS, and missing links from this data, where we note that unfortunately only 14% of measurements are available and LOS. We augment this real dataset with simulated data such that missing measurements are sampled from a zero-mean Gaussian with standard deviation equivalent to the observed LOS noise ($\sigma = 36.13$).

Fig. 14 show the results of our approach on this realistic dataset, where edges indicate error in distance. GCN achieves RMSE of 45.8 on this simulated data with threshold 110, while SMILE achieves an improved RMSE of 34.22 for $k = 17$. These results are the average performance of 10 trials, and the best parameters for each algorithm were selected empirically. We observe that the average localization error for unknown nodes is roughly the standard deviation of the noise on LOS links, while the standard deviation of NLOS error is 59.60 (mean 30.14m). This indicates that SMILE is able to achieve localization accuracy comparable to a distance-based model on a single LOS link for all nodes, even those which have many NLOS neighbors, and shows promising performance.

During robotic exploration, these anchors were deployed autonomously [15, 28]. This means anchors are only located in places the robots have already visited, while the exploration objective encourages the robots to move away from these anchors. In fact, the exploring robots then tend to be outside the convex hull defined by the anchors. This configuration appears to stress network localization, especially for smaller networks with significant noise. Prior work exists in robotic motion which minimizes localization uncertainty [6, 11], and network localization algorithms which specifically address this geometric setting may be an interesting direction for future work.

5.3 Discussion

One advantage of SMILE is that we compute **Y** which approximates the sparsity structure of **S**. Therefore, we can use this method to

[4] http://anrg.usc.edu/www/download_files/RSSLocalizationDataSet_11nodes.txt

[5] https://github.com/NeBula-Autonomy

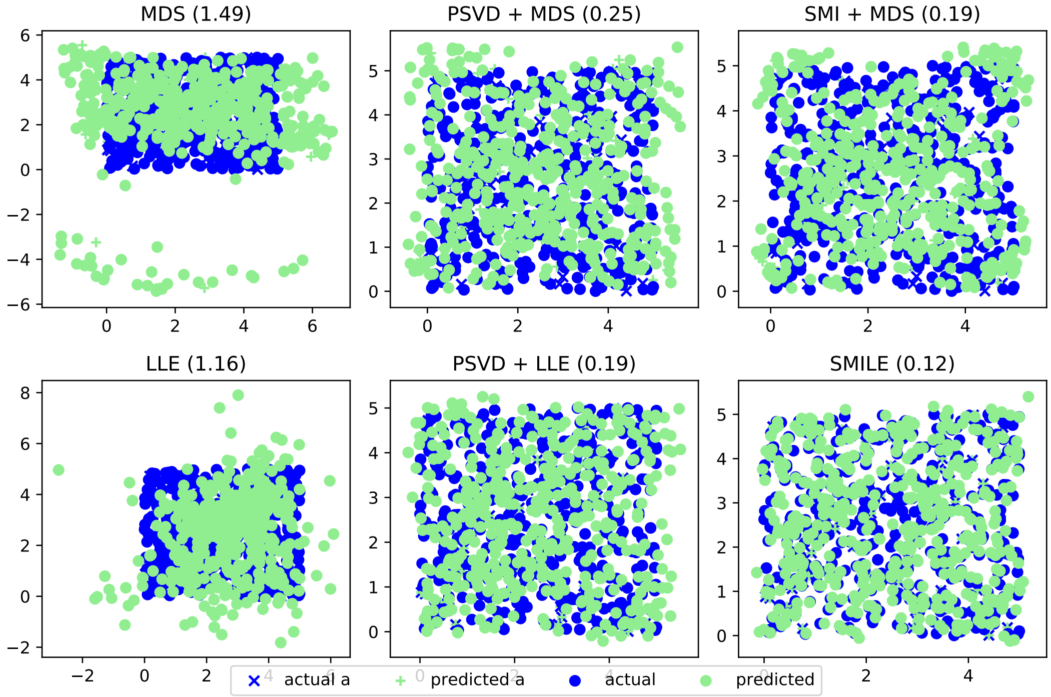

Figure 10: Ablation study with $\sigma = 0.3, \theta = 5$. Top row, left to right: classical MDS achieves RMSE 1.49, de-noising via PSVD improves RMSE to 0.25, and sparse matrix inference further improves RMSE to 0.19. Bottom row, left to right: LLE with direct position recovery achieves RMSE 1.16, de-noising via PSVD improves RMSE to 0.19, and and sparse matrix inference further improves RMSE to 0.12.

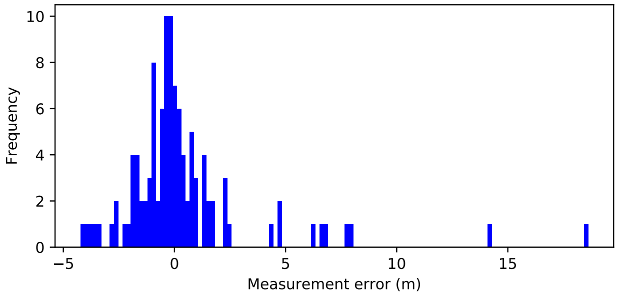

Figure 11: Measurement error for 11 node wireless sensor network outdoors, comparing distance from RSS (Eq. 7) with ground truth.

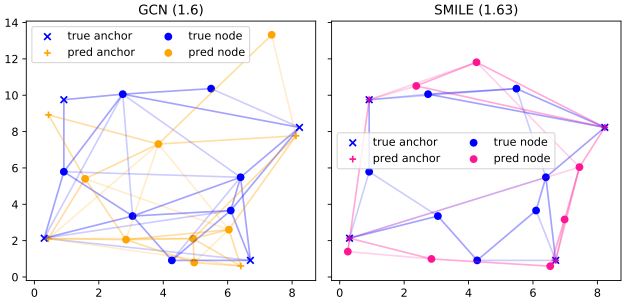

Figure 12: Localization accuracy of GCN (RMSE of 1.6) and SMILE (RMSE of 1.63) on the sensor network. Edges in the left graph represent observed distances less than θ_{GCn}. Edges in the right graph connect each node i to its nearest neighbors $NN(i)$.

estimate which links are NLOS and where walls, obstacles, or other potentially adversarial sources of attenuation may be present. This means SMILE could be useful as a complementary modality for simultaneous localisation and mapping (SLAM) algorithm for multi-robot systems, and has potential applications in robotic exploration for disaster mitigation and military applications [10].

While it is impossible to unique localize a network in the absence of anchors (given the possibility of translations and rotations), anchor-less localization problems have significant overlap with other useful signal processing problems. In face, SMILE could be directly applied to anchor-less localization problems by replacing lines 29-30 in the algorithm with the embedding step of conventional LLE, as LLE typically does not require anchors [30]. Thus, our approach may be applicable to problems outside the sensor networks domain. For instance, consider a matrix of user data, where each row corresponds to a specific user and a small number of

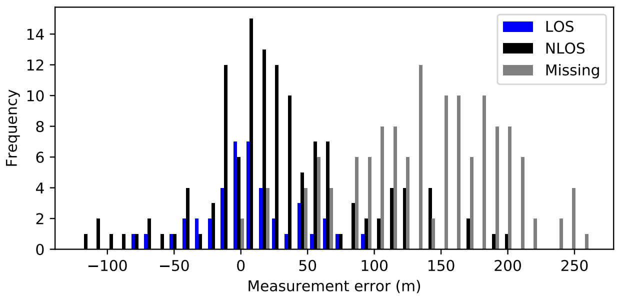

Figure 13: Measurement error for 18 node robotic network in GPS-denied environment.

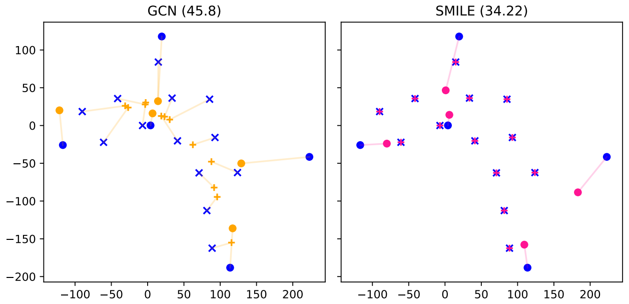

Figure 14: Localization accuracy of GCN (RMSE 45.8) and SMILE (RMSE 34.22) on the robotic network. Edges indicate error in distance.

users obfuscate their data with the addition of non-negative noise. SMILE could be applied to the problem of determining how similar users are, by extracting the sparse obfuscation matrix and finding an embedding of users in low-dimensional space.

We considered a modification to LLE in which we replace the k nearest neighbors with the n_A anchors as reference points. We expected this would minimize the risk of errors propagating, as each of the anchors is at a known location. Interestingly, we did not see an improvement in performance. Similarly, we considered a modification to the sparse matrix inference approach in which we project $\mathbf{X}$ and $\mathbf{Y}$ on to the space of matrices which are non-negative and symmetric. We expected this would improve performance as $\mathbf{D}$ and $\mathbf{S}$ are non-negative and symmetric. Preliminary experiments did not demonstrate a significant improvement in localization accuracy, and in some cases accuracy suffered. Low-rank matrix approximation which constrains the solution to have the characteristics of Euclidean distance matrix, without sacrificing localization accuracy, is a direction for further study.

Following [36], we considered the probability of each link being NLOS as independent. However, in realistic settings there will be some correlation based on the structure of the environment itself. For example, in Fig. 1 we illustrate 20 nodes in an indoor setting with three rooms and one hallway. Nodes in the smaller rooms are more subject to NLOS attenuation, given their proximity to two walls. Currently, we estimate that links with the weakest RSS are most likely to be NLOS. However it may instead be the case that certain nodes are more likely to have NLOS links, is in Fig. 1. Exploiting patterns like this in the structure of $\mathbf{S}$ is an interesting direction for future work.

6 CONCLUSION

We present Sparse Matrix Inference and Linear embedding, a novel network localization algorithm which is robust to noisy, NLOS, and missing measurements. Our approach outperforms the state of the art on large, simulated networks, achieving high localization accuracy with low computation times. We see promising results for small networks in real-world settings, and in the future we'd like to collect real-world data for larger networks. Other directions for future work include extensions to make the approach distributed [4, 12], or consider time-series RSSI measurements and Bayesian estimation [20, 35].

ACKNOWLEDGMENTS

Thanks to Kiran Yedavalli for the sensor network data and Team CoSTAR for the robotic network data. This work was funded in part by NASA Space Technology Research Fellowship Grant No. 80NSSC19K1189.

REFERENCES

[1] Jonathan Bachrach and Christopher Taylor. 2005. Localization in sensor networks. *Handbook of sensor networks: Algorithms and Architectures* 1 (2005), 277–289.

[2] Amir Beck, Petre Stoica, and Jian Li. 2008. Exact and approximate solutions of source localization problems. *IEEE Transactions on Signal Processing* 56, 5 (2008), 1770–1778. https://doi.org/10.1109/TSP.2007.909342

[3] Dimitris Bertsimas, Ryan Cory-Wright, and Nicholas AG Johnson. 2021. Sparse Plus Low Rank Matrix Decomposition: A Discrete Optimization Approach. *arXiv preprint arXiv:2109.12701* (2021).

[4] Pratik Biswas, Tzu Chen Lian, Ta Chung Wang, and Yinyu Ye. 2006. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks* 2, 2 (2006), 188–220. https://doi.org/10.1145/1149283.1149286

[5] Pratik Biswas and Yinyu Ye. 2004. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*. 46–54.

[6] Justin Cano and Jerome Le Ny. 2022. Ranging-Based Localizability-Constrained Deployment of Mobile Robotic Networks. *arXiv preprint arXiv:2202.00756* (2022).

[7] Yun Chang, Kamak Ebadi, Christopher E Denniston, Muhammad Fadhil Ginting, Antoni Rosinol, Andrzej Reinke, Matteo Palieri, Jingnan Shi, Arghya Chatterjee, Benjamin Morrell, et al. 2022. LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments. *arXiv preprint arXiv:2205.13135* (2022).

[8] Hongyang Chen, Gang Wang, Zizhuo Wang, Hing-Cheung So, and H Vincent Poor. 2011. Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks. *IEEE Transactions on Wireless Communications* 11, 1 (2011), 108–116.

[9] Tashnim JS Chowdhury, Colin Elkin, Vijay Devabhaktuni, Danda B Rawat, and Jared Oluoch. 2016. Advances on localization techniques for wireless sensor networks: A survey. *Computer Networks* 110 (2016), 284–305.

[10] Lillian Clark, Jeffrey A. Edlund, Marc Sanchez Net, Tiago Stegun Vaquero, and Ali-akbar Agha-Mohammadi. 2022. PropEM-L: Radio Propagation Environment Modeling and Learning for Communication-Aware Multi-Robot Exploration. In *Robotics: Science and Systems (RSS)*.

[11] Lillian Clark, Joseph Galante, Bhaskar Krishnamachari, and Konstantinos Psounis. 2021. A queue-stabilizing framework for networked multi-robot exploration. *IEEE Robotics and Automation Letters* 6, 2 (2021), 2091–2098.

[12] Jose A Costa, Neal Patwari, and Alfred O Hero III. 2006. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 2, 1 (2006), 39–64.

[13] Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. 2015. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine* 32, 6 (2015), 12–30.

[14] J Fink and V Kumar. 2010. Online Methods for Radio Signal Mapping with Mobile Robots. In *2010 IEEE ICRA*. IEEE, 1940–1945.

[15] Nobuhiro Funabiki, Benjamin Morrell, Jeremy Nash, and Ali-akbar Agha-mohammadi. 2020. Range-aided pose-graph-based SLAM: Applications of deployable ranging beacons for unknown environment exploration. *IEEE Robotics and Automation Letters* 6, 1 (2020), 48–55.

[16] David Kiyoshi Goldenberg, Arvind Krishnamurthy, Wesley C Maness, Yang Richard Yang, Anthony Young, A Stephen Morse, and Andreas Savvides. 2005. Network localization in partially localizable networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, Vol. 1. IEEE, 313–326.

[17] John Clifford Gower. 1985. Properties of Euclidean and non-Euclidean distance matrices. *Linear algebra and its applications* 67 (1985), 81–97.

[18] Neeraj Jain, Shekhar Verma, and Manish Kumar. 2015. Locally linear embedding for node localization in wireless sensor networks. In *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 126–130.

[19] Di Jin, Feng Yin, Michael Fauß, Michael Muma, and Abdelhak M Zoubir. 2020. Exploiting sparsity for robust sensor network localization in mixed LOS/NLOS environments. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5915–5915.

[20] Di Jin, Feng Yin, Carsten Fritsche, Fredrik Gustafsson, and Abdelhak M Zoubir. 2020. Bayesian cooperative localization using received signal strength with unknown path loss exponent: Message passing approaches. *IEEE Transactions on Signal Processing* 68 (2020), 1120–1135.

[21] Hongzi Mao, Malte Schwarzkopf, Shaileshh Bojja Venkatakrishnan, Zili Meng, and Mohammad Alizadeh. 2019. Learning scheduling algorithms for data processing clusters. In *Proceedings of the ACM special interest group on data communication*. 270–288.

[22] Stefano Marano, Wesley M Gifford, Henk Wymeersch, and Moe Z Win. 2010. NLOS identification and mitigation for localization based on UWB experimental data. *IEEE Journal on selected areas in communications* 28, 7 (2010), 1026–1035.

[23] Hoang Nt and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550* (2019).

[24] Neal Patwari, Joshua N Ash, Spyros Kyperountas, Alfred O Hero, Randolph L Moses, and Neiyer S Correal. 2005. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal processing magazine* 22, 4 (2005), 54–69.

[25] Neal Patwari, Alfred O Hero, Matt Perkins, Neiyer S Correal, and Robert J O'dea. 2003. Relative location estimation in wireless sensor networks. *IEEE Transactions on signal processing* 51, 8 (2003), 2137–2148.

[26] Amanda Prorok, Alexander Bahr, and Alcherio Martinoli. 2012. Low-cost collaborative localization for large-scale multi-robot systems. In *2012 IEEE International Conference on Robotics and Automation*. 4236–4241. https://doi.org/10.1109/ICRA.2012.6225016

[27] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[28] Maira Saboia, Lillian Clark, Vivek Thangavelu, Jeffrey A Edlund, Kyohei Otsu, Gustavo J Correa, Vivek Shankar Varadharajan, Angel Santamaria-Navarro, Thomas Touma, Amanda Bouman, et al. 2022. ACHORD: Communication-Aware Multi-Robot Coordination with Intermittent Connectivity. *arXiv preprint arXiv:2206.02245* (2022).

[29] Nasir Saeed, Haewoon Nam, Mian Imtiaz Ul Haq, and Dost Bhatti Muhammad Saqib. 2018. A survey on multidimensional scaling. *ACM Computing Surveys (CSUR)* 51, 3 (2018), 1–25.

[30] Lawrence K Saul and Sam T Roweis. 2000. An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html* (2000).

[31] T Schwengler. 2016. Wireless & Cellular Communications, Version 3.9. *Telecommunication Systems Laboratory, Colorado, USA*. 2 (2016).

[32] Yi Shang, Wheeler Rumi, Ying Zhang, and Markus Fromherz. 2004. Localization from connectivity in sensor networks. *IEEE Transactions on parallel and distributed systems* 15, 11 (2004), 961–974.

[33] Yi Shang and Wheeler Ruml. 2004. Improved MDS-based localization. In *IEEE INFOCOM 2004*, Vol. 4. IEEE, 2640–2651.

[34] Fei Wen, Lei Chu, Peilin Liu, and Robert C Qiu. 2018. A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning. *IEEE Access* 6 (2018), 69883–69906.

[35] Henk Wymeersch, Jaime Lien, and Moe Z Win. 2009. Cooperative localization in wireless networks. *Proc. IEEE* 97, 2 (2009), 427–450.

[36] Wenzhong Yan, Di Jin, Zhidi Lin, and Feng Yin. 2021. Graph neural network for large-scale network localization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5250–5254.

[37] Kiran Yedavalli, Bhaskar Krishnamachari, Sharmila Ravula, and Bhaskar Srinivasan. 2005. Ecolocation: a sequence based technique for RF localization in wireless sensor networks. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 285–292.

[38] Feng Yin, Carsten Fritsche, Di Jin, Fredrik Gustafsson, and Abdelhak M Zoubir. 2015. Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms. *IEEE Transactions on Signal Processing* 63, 6 (2015), 1448–1463.