Improved Reconstruction of Random Geometric Graphs

Varsha Dani ✉
Department of Computer Science, Rochester Institute of Technology, Rochester, NY, USA

Josep Díaz ✉
Department of Computer Science, Polytechnic University of Catalonia, Barcelona, Spain

Thomas P. Hayes ✉
Department of Computer Science, University of New Mexico, Albuquerque, NM, USA

Cristopher Moore ✉
Santa Fe Institute, NM, USA

Abstract

Embedding graphs in a geographical or latent space, i.e. inferring locations for vertices in Euclidean space or on a smooth manifold or submanifold, is a common task in network analysis, statistical inference, and graph visualization. We consider the classic model of random geometric graphs where n points are scattered uniformly in a square of area n, and two points have an edge between them if and only if their Euclidean distance is less than r. The reconstruction problem then consists of inferring the vertex positions, up to the symmetries of the square, given only the adjacency matrix of the resulting graph. We give an algorithm that, if $r = n^\alpha$ for $\alpha > 0$, with high probability reconstructs the vertex positions with a maximum error of $O(n^\beta)$ where $\beta = 1/2 - (4/3)\alpha$, until $\alpha \geq 3/8$ where $\beta = 0$ and the error becomes $O(\sqrt{\log n})$. This improves over earlier results, which were unable to reconstruct with error less than r. Our method estimates Euclidean distances using a hybrid of graph distances and short-range estimates based on the number of common neighbors. We extend our results to the surface of the sphere in $\mathbb{R}^3$ and to hypercubes in any constant dimension.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Reconstruction algorithm, distances in RGG, d-dimensional hypercube, 3 dimensional sphere

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.48

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2107.14323`

Funding *Josep Díaz*: Partially supported by PID-2020-112581GB-C21 (MOTION).
Thomas P. Hayes: Partially supported by NSF grant CCF-1150281.
Cristopher Moore: Partially supported by NSF grant IIS-1838251.

1 Introduction

Graph embedding is the art of assigning a position in some smooth space to each vertex, so that the graph's structure corresponds in some way to the metric structure of that space. If vertices with edges between them are geometrically close, this embedding can help us predict new or unobserved links, devise efficient routing strategies, and cluster vertices by similarity – not to mention (if the embedding is in two dimensions) give us a picture of the graph that we can look at and perhaps interpret. In social networks, this space might correspond literally to geography, or it might be a "latent space" whose coordinates measure ideologies, affinities between individuals, or other continuous demographic variables (e.g. [15]). In some applications the underlying space is known; in others we wish to infer it, including the number of dimensions, whether it is flat or hyperbolic, and so on.

The literature on graph embedding is vast, and we apologize to the many authors who we will fail to cite. However, despite the broad utility of graph embedding in practice (see [27] for a recent experimental review) many popular heuristics lack rigorous guarantees. Here we pursue algorithms that reconstruct the position of every vertex with high accuracy, up to a symmetry of the underlying space.

Many versions of the reconstruction problem, including recognizing whether a graph has a realization as a geometric graph, are NP-complete [5, 8, 9] in the worst case. Thus we turn to distributions of random instances, and design algorithms that succeed with high probability in the instance. For many inference problems, there is a natural generative model where a ground truth structure is "planted," and the instance is then chosen from a simple distribution conditioned on its planted structure. For community detection a.k.a. the planted partition problem, for instance, we can consider graphs produced by the stochastic block model, a generative model where each vertex has a ground-truth label, and each edge (u, v) exists with a probability that depends on the labels of u and v. Reconstructing these labels from the adjacency matrix then becomes a well-defined problem in statistical inference, which may or may not be solvable depending on the parameters of the model (e.g. [1,19,20]). In the same spirit, a series of papers has asked to what extent we can reconstruct vertex positions from the adjacency matrix in random geometric graphs, where vertex positions are chosen independently from a simple distribution.

Random geometric graphs

Let n be an integer and let $r > 0$ be real. Let $V = \{v_i\}_{i=1}^n$ be a set of n points chosen uniformly at random in the square $[0, \sqrt{n}]^2$. The *random geometric graph* $G \in \mathcal{G}(n, r)$ has vertex set V and edge set $E = \{(u, v) : \|u - v\| \le r\}$ where $\|u - v\|$ denotes the Euclidean distance. (We will often abuse notation by identifying a vertex with its position.)

This is a rescaling of the *unit disk model* where $r = 1$. We follow previous authors in varying the average degree of the graph by varying r rather than varying the density of points in the plane. Since the square has area n, the density is always 1: that is, the expected number of points in any measurable subset is equal to its area.

It is also natural to consider a Poisson model, where the points are generated by a Poisson point process with intensity 1. In that case the number of vertices fluctuates but is concentrated around n, and the local properties of the two models are asymptotically the same. The number of points in a region of area A is binomially distributed in the uniform model, and Poisson distributed with mean A in the Poisson model. In both cases, the probability that such a region of area is empty is at most e^{-A}; this is exact in the Poisson model, and is an upper bound on the probability $(1 - A/n)^n$ in the uniform model.

Random geometric graphs (RGGs) were first introduced by Gilbert in the early 1960s to model communications between radio stations [14]. Since then, RGGs have been widely used as models for wireless communication, in particular for wireless sensor networks. RGGs have also been extensively studied as mathematical objects, and much is known about their asymptotic properties [23, 26]. One well-known result is that $r_c = \sqrt{\log n/\pi}$ is a *sharp threshold* for connectivity for $G \in \mathcal{G}(n, r)$ in the square in both the uniform and Poisson models: that is, for any $\varepsilon > 0$, with high probability G is connected if $r > (1 + \varepsilon)r_c$ and disconnected if $r < (1 - \varepsilon)r_c$.

More generally, we can define RGGs on any compact Riemannian submanifold, by scattering n points uniformly according to the surface area or volume. We then define the edges as $E = \{(u, v) : \|u - v\|_g \le r\}$ where $\|\cdot\|_g$ is the geodesic distance, i.e. the arc length of the shortest geodesic between u and v. On the sphere in particular this includes the cosine distance, since $\|u - v\|_g$ is a monotonic function of the angle between u and v.

The reconstruction problem

Given the adjacency matrix A of a random geometric graph defined on a smooth submanifold M, we want to find an embedding $\phi : V \to M$ which is as close as possible to the true positions of the vertices. As a measure of accuracy, we focus on the max distance $\max_v \|\phi(v) - v\|$ where we identify each vertex v with its true position.

However, if we are only given A, the most we can ask is for ϕ to be accurate up to M's symmetries. In the square, for instance, applying a rotation or reflection to the true positions results in exactly the same adjacency matrix. Thus we define the *distortion* $d^*(\phi)$ as the minimum of the maximum error achieved by composing ϕ with some element of the symmetry group $\mathrm{Sym}(M)$,

$$d^*(\phi) = \min_{\sigma \in \mathrm{Sym}(M)} \max_{v \in V} \|(\sigma \circ \phi)(v) - v\| . \tag{1}$$

We will sometimes refer to the distortion of a subset of the vertices or of a single vertex.

As in previous work, our strategy is to estimate the distances between pairs of vertices, and then use geometry to find points with those pairwise distances. We focus on the case where $M = [0, \sqrt{n}]^2$ and $\|\cdot\|$ is the Euclidean distance. However, many of our results apply more generally, both in higher dimensions and on curved manifolds.

Our contribution

An intuitive way to estimate the Euclidean distance $\|u - v\|$ in a random geometric graph is to relate it to the graph distance $d_G(u, v)$, i.e. the number of edges in a topologically shortest path from u to v. The upper bound $\|u - v\| \le r d_G(u, v)$ is obvious. Moreover, if the graph is dense enough, then shortest paths are fairly straight geometrically and most of their edges have Euclidean length almost r, and this upper bound is not too far from the truth [4, 7, 13, 21].

As far as we know, the best upper and lower bounds relating Euclidean distances to graph distances in RGGs are given in [12]. In [11] these bounds were used to reconstruct with distortion $(1 + o(1))r$ when r is sufficiently large, namely if $r = n^\alpha$ for some $\alpha > 3/14$.

However, since the graph distance d_G is an integer, the bound $\|u - v\| \le r d_G(u, v)$ cannot distinguish Euclidean distances that are between two multiples of r. Thus, as discussed after the statement of Theorem 4 below, the methods of [11] cannot avoid a distortion that grows as $\Omega(r)$. Intuitively, the opposite should hold: as r grows the graph gets denser, neighborhoods get smoother, and more precise reconstructions should be possible.

We break this $\Omega(r)$ barrier by using a hybrid distance estimate. First we note that $r d_G(u, v)$ is a rather good estimate of $\|u - v\|$ if $\|u - v\|$ is just below a multiple of r, and we improve the bounds of [12] using a greedy routing analysis. We then combine $r d_G$ with a more precise short-range estimate based on the number of neighbors that u and v have in common. In essence, we use a quantitative version of the popular heuristic that two vertices are close if they have a large Jaccard coefficient (see e.g. [24] for link prediction, and [2] for a related approach to small-world graphs).

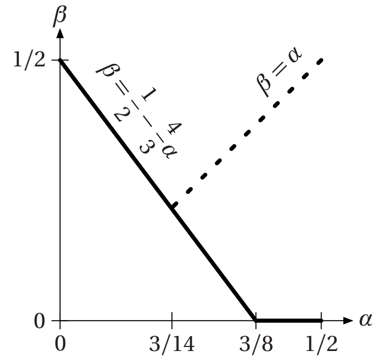

Figure 1 Our results (solid) compared to those of [11] (dotted). If $r = n^\alpha$, our reconstruction has distortion $O(n^\beta)$ where $\beta = 1/2 - (4/3)\alpha$, except for $\alpha > 3/8$ where the distortion is $O(\sqrt{\log n})$. The algorithm of [11] applies when $\alpha > 3/14$ and gives $\beta = \alpha$, i.e. distortion $\Theta(r)$. Our results apply for any constant $0 < \alpha < 1/2$ and give lower distortion than [11] when $\alpha > 3/14$.

As a result, we obtain a distortion d^* that decreases with r. Namely, if $r = n^\alpha$ for $\alpha > 0$, then $d^* = O(n^\beta)$ where $\beta = 1/2 - (4/3)\alpha$, until for $\alpha \geq 3/8$ where $d^* = O(\sqrt{\log n})$. (Note that any $\alpha > 0$ puts us well above the connectivity threshold.) Since it uses graph distances, the running time of our algorithm is essentially the same as that of All-Pairs Shortest Paths. To our knowledge, this is the smallest distortion achieved by any known polynomial-time algorithm. We compare our results with those of [11] in Figure 1.

We show that our results extend to higher dimensions and to some curved manifolds as well. With small modifications, our algorithm works in the m-dimensional hypercube for any fixed m (the distortion depends on m, but the running time does not). We also sketch a proof that it works on the surface of the sphere, using spherical rather than Euclidean geometry, solving an open problem posed in [11]. Our techniques are designed to be easy to apply on a variety of curved manifolds and submanifolds, although we leave the fullest generalizations to future work.

We use $N(u) = \{w : (u, w) \in E\}$ to denote the topological neighborhood of a vertex u, and $B(u, r)$ to denote the geometrical ball around it. Our results, as well as many of the cited results, hold *with high probability* (w.h.p.) in the random instance $G \in \mathcal{G}(n, r)$, i.e. with probability tending to 1 as $n \to \infty$. When we consider randomized algorithms, the probability is over both $\mathcal{G}(n, r)$ and the randomness of the algorithm.

Other related work

In the statistics community there are a number of consistency results for maximum-likelihood methods (e.g. [25]) but it is not clear how the accuracy of these methods scales with the size or density of the graph, or how to find the maximum-likelihood estimator efficiently. There are also results on the convergence of spectral methods, using relationships between the graph Laplacian and the Laplace-Beltrami operator on the underlying manifold (e.g. [3]). This approach yields bounded distortion for random dot-product graphs in certain regimes.

We assume that parameters of the model are known, including the underlying space and its metric structure (in particular, its curvature and the number of dimensions). Thus we avoid questions of model selection or hypothesis testing, for which some lovely techniques have been proposed (e.g. [10, 18, 22]). We also assume that the parameter r is known, since this is easy to estimate from the typical degree.

Organization of the extended abstract

In Section 2 we define the concept of a *deep* vertex. Intuitively, a vertex is deep if it is more than r from the boundary of the square, so that its ball of potential neighbors is entirely in the interior. However, since we are only given the adjacency matrix, we base our definition on the number of vertices two steps away from v in the graph, and show that these topological and geometric properties are closely related.

Section 3 shows that we can closely approximate Euclidean distances $\|u-v\|$ given the adjacency matrix whenever v is deep. We do this in two steps: we give a precise short-range estimate of $\|u-v\|$ when $d_G(u,v) \leq 2$, and a long-range estimate that uses the existence of a greedy path. By "hybridizing" these two distance estimates, switching from long to short range at a carefully chosen intermediate point, we obtain a significantly better estimate of $\|u-v\|$ than was given in [12]. We believe these distance estimation techniques may be of interest in themselves.

In Section 4, we use this new estimate of Euclidean distant to reconstruct the vertex positions up to a symmetry of the square, by starting with a few deep "landmarks" and then triangulating to the other vertices. This gives smaller distortion than the algorithm in [11], achieving the scaling shown in Figure 1.

Finally, in Section 5 we extend our method to random geometric graphs in the m-dimensional hypercube and on the surface of the sphere.

Due to space limitation, in this extended abstract, we limit ourselves to sketching the proofs and their main ideas, deferring the complete proofs to the full version of the paper, available on ArXiv: `https://arxiv.org/abs/2107.14323`

2 Deep vertices

Let G be a random geometric graph defined in the two-dimensional square $[0,\sqrt{n}]^2$. Because some of our arguments will break down for vertices near the boundary and corners of $[0,\sqrt{n}]^2$, it will be useful to have an easy way to tell these vertices apart from the rest. To this end, we introduce the notion of deep vertices.

▶ **Definition 1.** Let r be fixed. We say that a vertex $v \in V$ is *deep* if at least $11r^2$ vertices have graphical distance 2 or less from v.

Note that being deep is a topological property of the graph, rather than its embedding in the plane. We need such a definition since our reconstruction algorithm is only given access to the adjacency matrix. However, in the long version we show that with high probability all vertices that are deep in this topological sense are at least r from the boundary of the square. Moreover, with high probability there are many deep vertices.

3 Estimating Euclidean distances: Breaking the $\Omega(r)$ barrier

3.1 Estimating short-range distances

In this section we show how to estimate the Euclidean distance $\|u-v\|$ between two vertices that are topologically close, namely when $d_G(u,v) \leq 2$.

We first assume that $d_G(u,v) = 1$, i.e., that $\|u-v\| = x$ where $0 \leq x \leq r$. Then $\{N(v) \setminus N(u)\}$ consists of the points in the lune $L = B(v,r) \setminus B(u,r)$ shown in Figure 2. If v is deep, then $B(v,r)$ and therefore L lies in the interior of the square $[0,\sqrt{n}]^2$. Thus in expectation $|\{N(v) \setminus N(u)\}|$ is the area of L, which we denote $F(x)$. This suggests inverting F, estimating x as

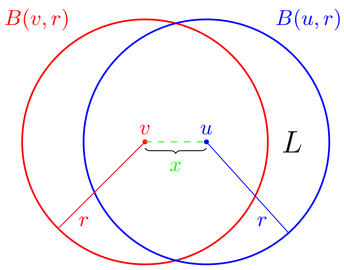

Figure 2 We can estimate the Euclidean distance $\|u-v\| = x$ of two vertices with $d_G(u,v) \le 2$ using the area of the lune $L = B(u,r) \cap B(v,r) \ne \emptyset$. Denoting this area $F(x)$, we can estimate x by appling the inverse F^{-1} to the number of points in $N(v) \setminus N(u)$.

$$\tilde{d}(u,v) = F^{-1}\left(|N(v) \setminus N(u)|\right) . \tag{2}$$

Since F is monotonic and is given explicitly as $F(x) = \pi r^2 - 2r^2 \arccos \frac{x}{2r} + \frac{x}{2}\sqrt{4r^2 - x^2}$, we can compute this inverse using binary search.

This estimate is w.h.p. an accurate estimate of $\|u-v\|$ for two reasons. First, $|\{N(v) \setminus N(u)\}|$ is concentrated around its expectation $F(x)$. In both the uniform and the Poisson models, with high probability we have

$$\Big| |\{N(v) \setminus N(u)\}| - F(x) \Big| \le \sqrt{F(x) \log n} .$$

Second, the derivative of F is large, so the derivative of F^{-1} is small. Specifically, since $F(x)$ satisfies the differential equation $F'(x) = \sqrt{4r^2 - x^2}$, we have $F'(x) \ge r\sqrt{3} = \Omega(r)$ for $0 < x < r$. Noting also that $F(x) = \Theta(xr)$, we obtain

$$\left| x - \tilde{d}(u,v) \right| \le \frac{\sqrt{F(x)\log n}}{F'(x)} = O\left(\sqrt{\frac{x \log n}{r}}\right) . \tag{3}$$

If $d_G(u,v) = 2$ in which case $r < x < 2r$, we switch from the difference in the two neighborhoods to their intersection $N(u) \cap N(v)$, namely the points in the lens-shaped region $B(u,r) \cap B(v,r)$ in Figure 2 which has area $\pi r^2 - F(x)$. As $x \to 2r$ the area of this region tends to zero, but so does $F'(x)$. Specifically, if $x = 2r - \varepsilon$ then

$$\pi r^2 - F(x) = \Theta(r^{1/2}\varepsilon^{3/2}) \quad \text{and} \quad F'(x) = \Theta(r^{1/2}\varepsilon^{1/2}) ,$$

so (3) becomes

$$\left| x - \tilde{d}(u,v) \right| = O\left(\frac{\sqrt{r^{1/2}\,\varepsilon^{3/2}\log n}}{r^{1/2}\,\varepsilon^{1/2}}\right) = O\left(\left(\frac{\varepsilon}{r}\right)^{1/4}\sqrt{\log n}\right) . \tag{4}$$

Putting this all together gives the main theorem of the section,

▶ **Theorem 2.** *Given a $G \in \mathcal{G}(n,r)$, where $r \ge 100\sqrt{\log n}$. With probability at least $1 - 2/n^2$ we have, for all vertices $v \ne w$ such that $d_G(v,w) \le 2$ and v is deep,*

$$\left| \|v-w\| - \tilde{d}(v,w) \right| \le 100\eta(\|v-w\|)\sqrt{\log n} , \tag{5}$$

where $\eta : [0, 2r] \to [0, 1]$ *is defined by*

$$\eta(x) = \begin{cases} \frac{\sqrt{\log n}}{r} & \text{for } 0 \le x \le \frac{\log n}{r}, \\ \sqrt{\frac{x}{r}} & \text{for } \frac{\log n}{r} \le x \le r, \\ \left(\frac{2r-x}{r}\right)^{1/4} & \text{for } r \le x \le 2r - \frac{(\log n)^{2/3}}{r^{1/3}}, \\ \frac{(\log n)^{1/6}}{r^{1/3}} & \text{for } 2r - \frac{(\log n)^{2/3}}{r^{1/3}} \le x \le 2r. \end{cases}$$

3.2 Estimating long-range distances

Next we show a fairly tight relationship between geometric and topological distance for all pairs of vertices, including distant ones. This is a slightly sharper version of [12, Thm 1.1]. The main difference is that, where before, a short path between two given vertices is found by finding vertices close to a straight line between the endpoints, our proof instead analyses a greedy algorithm generating a path that may deviate further from the straight line.

We start with the following geometrical lemma

▶ **Lemma 3.** *Let $B_1(v, r_1)$ and $B_1(u, r_2)$ be overlapping balls in $\mathbb{R}^2$, and let $d = \|u - v\|$. Consider the lens $L = B_1 \cap B_2$. Let δ denote the width of L, i.e., $\delta = r_1 + r_2 - d$. Then the area A of L satisfies*

$$A = \Theta\left(\delta^{3/2} \min\{r_1, r_2\}^{1/2}\right).$$

The main result of this section is the following theorem,

▶ **Theorem 4.** *Let $G \in \mathcal{G}(n, r)$. There exist absolute constants C_1, C_2, C_3 such that, for all $n \ge 1$ and all $r \ge C_1\sqrt{\log n}$, with probability at least $1 - C_2/n^2$, all pairs of vertices u, v satisfy*

$$\left\lceil \frac{\|u - v\|}{r} \right\rceil \le d_G(u, v) \le \left\lceil \frac{\|u - v\| + \kappa}{r} \right\rceil, \tag{6}$$

where

$$\kappa = \kappa(\|u - v\|) = C_3\left(\frac{\|u - v\|}{r^{4/3}} + \frac{\log n}{r^{1/3}}\right). \tag{7}$$

The lower bound of (6) is trivial. The gist of the upper bound is to show the existence of a short path $u \rightsquigarrow v$ using a greedy routing algorithm that moves as close as possible, in Euclidean distance, to v at each step. (Note that this is only for the purpose of analysis, since our reconstruction algorithm is only given the adjacency matrix!) Start from $x_0 = u$. Then for each $i \ge 0$, let x_{i+1} be the neighbor of x_i that minimizes $\|x_i - v\|$ as shown in Fig. 3 (note that x_{i+1} is unique with probability 1). The algorithm terminates if no neighbor of x_i is closer to v than x_i is. If $x_i = v$, we have found our path and the algorithm succeeds. Otherwise, the algorithm has gotten stuck in a local minimum, and never reaches v.

Then in the full version we prove that the algorithm succeeds with probability $1 - O(n^{-3})$. Moreover, with the help of Lemma 3, we can show that each step gets about $r - O(r^{-1/3})$ closer to v. This yields the upper bound of (6), and taking a union bound over all pairs u, v completes the proof. ◀

Let us discuss how we will use Theorems 2 and 4 to break the $\Omega(r)$ barrier in distance estimation, and thus in reconstruction. Suppose $r = n^\alpha$ where $0 < \alpha < 1/2$ is a constant. Then since $\|u - v\| = O(n^{1/2})$, we have from (7)

$$\kappa = O\left(\max\left(n^{\frac{1}{2} - \frac{4}{3}\alpha}, n^{-\frac{1}{3}\alpha} \log n\right)\right), \tag{8}$$

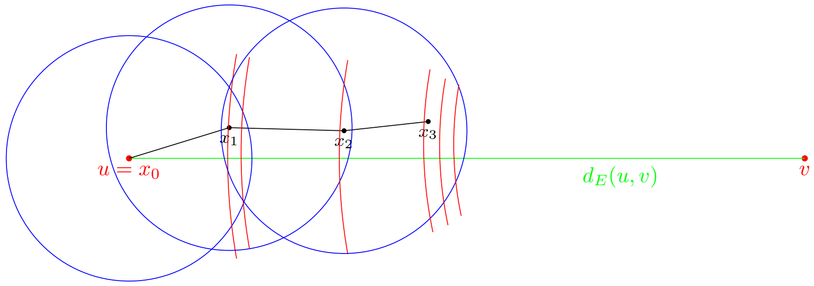

Figure 3 The greedy routing analysis of Theorem 4. At each step we go from x_i to the neighbor x_{i+1} closest to v. In the analysis, we consider the intersections of x_i's neighborhood with balls centered at v, with the radii of the latter chosen so that these intersections have area $\ln 2$, $2 \ln 2$, $3 \ln 2$, and so on. Each of these intersections contains a point with constant probability, so that most steps make significant progress towards v.

and since $\frac{1}{2} - \frac{4}{3}\alpha > -\frac{1}{3}\alpha$ we have

$$\kappa = O(n^{\beta}) \,, \quad \text{where} \quad \beta = \frac{1}{2} - \frac{4}{3}\alpha \,. \tag{9}$$

If $\alpha > 3/14$, then $\beta < \alpha$ and $\kappa = o(r)$. In this case the upper and lower bounds on $d_G(u, v)$ differ by at most 1, and moreover are equal for most pairs of vertices, making $d_G(u, v)$ a nearly-deterministic function of $\|u - v\|$. Using $\lceil x \rceil \le x + 1$ and multiplying through by r gives the bounds

$$d_G(u, v) r - (r + \kappa) \le \|u - v\| \le d_G(u, v) r \,,$$

so that $d_G(u, v) r$ is an estimate of $\|u - v\|$ with error $r + \kappa = (1 + o(1)) r$. Previous work [11,12] used essentially this bound to reconstruct the graph with a distortion of $(1+\varepsilon) r$ for arbitrarily small constant ε. This gives the performance shown by the dotted line in Figure 1.

But in fact $d_G(u, v) r$ is a much more accurate estimate of $\|u - v\|$ for certain pairs of vertices. If $\|u - v\|$ is just below a multiple of r, then rounding up the left and right sides of (6) doesn't change either very much. We state this with in the following corollary,

▶ **Corollary 5.** *With $\kappa = \kappa(\|u - v\|)$ defined as in* (7)*, suppose that for some $0 \le \delta < r$ and some integer $t \ge 0$ we have $tr - (\kappa + \delta) < \|u - v\| < tr - \kappa$. Then*

$$d_G(u, v) r - (\kappa + \delta) \le \|u - v\| \le d_G(u, v) r \,. \tag{10}$$

Thus, if $\|u - v\|$ is in one of these intervals, Theorem 4 lets us estimate $\|u - v\|$ from the adjacency matrix with error $\delta + \kappa$ instead of $r + \kappa$. Below we will combine this with the more precise estimate of short-range distances from Theorem 2 to achieve this error for all pairs u, v where v is deep, not just those for which $\|u - v\|$ is almost a multiple of r.

As a result, the error in our distance estimates and the distortion of our reconstruction is $O(r^{\beta})$ where β decreases from 1 to 0 as α increases as shown by the solid line in Figure 1. Specifically, we obtain a nontrivial result for any $\alpha > 0$ and a more accurate reconstruction than in [11] in the range $\alpha > 3/14$ where their theorem applies. At $\alpha = 3/8$ where $\beta = 0$ another source of error takes over, leaving us with $O(\sqrt{\log n})$ distortion.

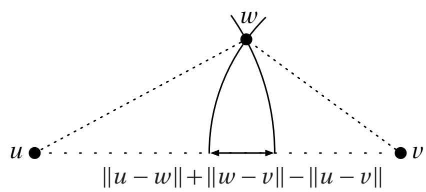

Figure 4 For any intermediate point w, the hybrid distance estimate $d_1(u,w)+d_2(w,v)$ is an upper bound on $||u-v||$ with error bounded by Lemma 8.

3.3 Hybrid distance estimates

In this subsection we combine the long-range estimates of Theorem 4 with the short-range estimates in Theorem 2, to estimate Euclidean distances with an error of $o(n)$. We start with the following definition:

▶ **Definition 6.** *Let $V \subset \mathbb{R}^2$ and $d : V^2 \to [0,\infty)$ and $\varepsilon : \mathbb{R} \to [0,\infty)$ be two functions satisfying, for all $u,v \in V$, $d(u,v) - \varepsilon(u,v) \le \|u-v\| \le d(u,v)$. Then we say d is an upper bound on Euclidean distance with error function ε.*

The basic tool for combining distance estimates is the following lemma.

▶ **Lemma 7.** *If d_1 and d_2 are upper bounds on Euclidean distance with error functions $\varepsilon_1, \varepsilon_2$ respectively, then $\min\{d_1,d_2\}$ is an upper bound on Euclidean distance with error $\min\{\varepsilon_1,\varepsilon_2\}$.*

The next lemma shows another way to combine two upper bounds on $\|u-v\|$. We choose a vertex w between u and v and use the triangle inequality, using d_1 to bound $\|u-w\|$ and d_2 to bound $\|w-v\|$. Finally, we minimize over all intermediate vertices w. This hybrid is especially useful when, as with our long-range and short-range estimates, d_1 and d_2 have different ranges of $\|u-v\|$ in which they achieve small error.

▶ **Lemma 8.** *Suppose d_1 and d_2 are upper bounds on Euclidean distance with error functions ε_1 and ε_2 respectively. Define the hybrid distance estimate by*

$$\widehat{d}(u,v) = \min_{w} \big(d_1(u,w) + d_2(w,v)\big)\,. \tag{11}$$

Then $\widehat{d}$ is an upper bound on Euclidean distance with error

$$\widehat{\varepsilon}(u,v) \le \min\nolimits_w \Big[\varepsilon_1(u,w) + \varepsilon_2(w,v) + \|u-w\| + \|w-v\| - \|u-v\|\Big].$$

For an intuition of the proof, see Figure 4. ⌟

The next lemma uses the fact that if a lens is sufficiently large to contain at least one point w with high probability, then this gives an upper bound on the minimum in Lemma 8.

▶ **Lemma 9.** *Let $G \in \mathcal{G}(n,r)$ and suppose that with high probability d_1 and d_2 are upper bounds on Euclidean distance with errors $\varepsilon_1(u,v) = \varepsilon_1(\|u-v\|)$ and $\varepsilon_2(u,v) = \varepsilon_2(\|u-v\|)$. Define $\widehat{d}$ as in Lemma 8. Then there is a constant C such that, with high probability, $\widehat{d}$ is also an upper bound on Euclidean distance, with error $\widehat{\varepsilon}(u,v) = \widehat{\varepsilon}(\|u-v\|)$ where*

$$\widehat{\varepsilon}(\|u-v\|) \le \min_{0<x<\|u-v\|}\ \max_{0\le\delta_1,\delta_2\le\delta(x)} \big[\varepsilon_1(x+\delta_1) + \varepsilon_2(\|u-v\|-x-\delta_2) + \delta(x)\big], \tag{12}$$

with $\delta(x) = C(\log n)^{2/3}\left(\min\{x, \|u-v\|-x\}\right)^{-1/3}$.

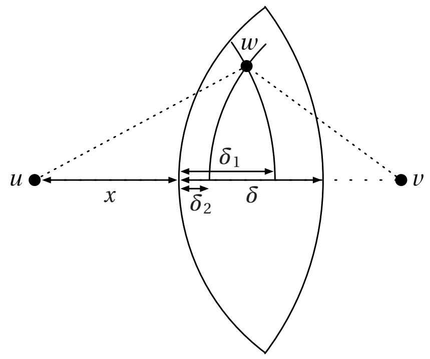

Figure 5 The lens $L(x)$ of Lemma 9. If δ is large enough, this lens is nonempty with high probability, in which case we can use any point w in it as an intermediate point for Lemma 8.

Proof sketch. Fix u, v and consider the lens $L(x) = B(u, x + \delta) \cap B(u, x + \|u - v\| - x)$ of width δ as shown in Fig. 5. By Lemma 3, the area of $L(x)$ is proportional to $C^{3/2} \log n$. Since the probability a region of area A is empty is at most e^{-A}, w.h.p. $L(x)$ contains at least one vertex w. For C sufficiently large, Lemma 8 then yields (12). ◀

Now we use the previous lemma to break the $\Omega(r)$ barrier for the error in estimating Euclidean distances in $G \in \mathcal{G}(n, r)$.

Assume v is deep. First define $d_1 = r d_G(u, v)$, i.e., the upper bound of Corollary 5. Now define d_2 using the precise short-range estimate $\tilde{d}$ from Theorem 2, with a small increment to make it an upper bound on Euclidean distance with high probability. Specifically, for a sufficiently large constant C_2, let

$$d_2(u, v) = \begin{cases} \tilde{d}(u, v) + C_2\sqrt{\log n} & \text{if } d_G(u, v) \leq 2\,, \\ +\infty & \text{otherwise}\,. \end{cases} \tag{13}$$

▶ Remark 10. Given this choice of d_1 and d_2, the hybrid estimate $\widehat{d}(u, v)$ is the graph distance from u to v in a weighted graph G_v where each edge (w, v) with $d_G(w, v) \leq 2$ has weight $\tilde{d}(w, v) + C_2\sqrt{\log n}$ and all other edges have weight r. Thus, for any fixed v, we can compute $\widehat{d}(u, v)$ for all u in with an application of Dijkstra's algorithm.

Now let us bound the error functions ε_1 and ε_2 of d_1 and d_2. As discussed above, for most values of $\|u - v\|$ we have $\varepsilon_1(\|u - v\|) = \Theta(r)$. However, we will choose the lens in Lemma 9 so that $\|u - w\|$ is almost a multiple of r, in which case Corollary 5 shows that $\varepsilon_1(\|u - w\|)$ is much smaller.

To bound ε_2, Theorem 2 implies that, for some absolute constant C_4, w.h.p.

$$\varepsilon_2(\|u - v\|) \leq \begin{cases} C_4\sqrt{\log n} & \text{if } \|u - v\| \leq 2r - C_4 r^{-1/3} \log n, \\ +\infty & \text{otherwise.} \end{cases} \tag{14}$$

Having gathered these facts, we will apply Lemma 9 to d_1 and d_2 with a judicious choice of lens $L(x)$. First note that, since $d_2(w, v) = +\infty$ if $d_G(w, v) > 2$, we can write

$$\widehat{d}(u, v) = \min_{w : d_G(w, v) \leq 2} \{d_1(u, w) + d_2(w, v)\}. \tag{15}$$

▶ **Theorem 11.** *Let $r = n^\alpha$ for a constant $0 < \alpha < 1/2$. For all pairs u, v where v is deep, define $\widehat{d}(u,v)$ as in eq. 15. Then w.h.p., $\widehat{d}$ is an upper bound on the Euclidean distance $\|u - v\|$ with error*

$$\widehat{\varepsilon}(u,v) \le C' \begin{cases} n^{\frac{1}{2}-\frac{4}{3}\alpha} & \alpha < 3/8, \\ \sqrt{\log n} & 3/8 \le \alpha < 1/2, \end{cases} \tag{16}$$

for some absolute constant C'. That is, $\widehat{d}(u,v) - \widehat{\varepsilon}(u,v) \le \|u - v\| \le \widehat{d}(u,v)$.

Proof sketch. We choose x and the lens $L(x)$ in Lemma 3 such that $\|u - w\|$ is almost an integer multiple of r. We use this choice of x to upper bound (12), bounding the two terms inside the minimum separately. Using the definition of κ in Theorem 4, Corollary 5 tells us that $d_1(u,w)$ has error at most $\varepsilon_1 \le \kappa + \delta$. This implies that for all $0 \le \delta_1 \le \delta$, the first term of (12) is at most $\varepsilon_1(x + \delta_1) \le \kappa + \delta$.

To bound the second term of (12) we first prove that w.h.p. $d_G(w,v) \le 2$. We then use (14) to get $\varepsilon_2(w,v) \le C_4\sqrt{\log n}$, which implies $\varepsilon_2(\|u - v\| - x - \delta_2) \le C_4\sqrt{\log n}$. ◀

4 The Reconstruction Algorithm

In this section we use our distance estimates to reconstruct the positions of the points up to a symmetry of the square. Our global strategy is similar to [11]: we first fix a small number of "landmark" vertices v whose positions can be estimated accurately up to a symmetry of the plane. Then for each vertex u we use the estimated distances $\widehat{d}(u,v)$ to reconstruct u's position by triangulation. In [11], the landmarks are vertices close to the corners of the square. Here they will instead be a set of three deep vertices that are far from collinear, forming a triangle which is acute and sufficiently large.

▶ **Definition 12.** *We say a triple of deep vertices x, y, z is* good *if they form an acute triangle with all three side lengths at least $0.1\sqrt{n}$.*

▶ Remark 13. The bounds of Theorem 4 imply that if x, y, z are deep and have pairwise graph distances in the interval $[0.1\sqrt{n}/r, 0.14\sqrt{n}/r]$, then they are a good triple; the triangle is acute since $0.14 < 0.1\sqrt{2}$.

Once we have found a good triple, we perform triangulation using the following lemma.

▶ **Lemma 14.** *Let x, y, z, u be four points in the plane. Suppose x, y, z form an acute triangle with minimum side length at least ℓ. Then, if we know the positions of x, y, z with error at most η, and we have upper bounds $\widehat{d}(u,v)$ on the Euclidean distances $\|u - v\|$ for all $v \in \{x, y, z\}$ with error $\widehat{\varepsilon}$, and all of these distances are at most D, we can determine the position of u relative to x, y, z with error at most*

$$C_5 \frac{D(\widehat{\varepsilon} + \eta)}{\ell}, \tag{17}$$

for an absolute constant C_5.

▶ Remark 15. In our application, (x, y, z) is a good triple, so $\ell = \Omega(\sqrt{n})$. Since we also have $D \le \sqrt{2n}$ and $\eta = O(\widehat{\varepsilon})$, we can reconstruct u's position relative to x, y, z with error $O(\widehat{\varepsilon})$.

Proof. First, let us assume fixed positions for x, y, z within η of their estimated positions (which we can always do so that they form an acute triangle). By the triangle inequality, this changes the distances $\|u - v\|$ for $v \in \{x, y, z\}$ by at most $\pm\eta$. Thus u is in the intersection U of three annuli,

$$U = \bigcap_{v \in \{x,y,z\}} B(v, \widehat{d}(u, v) + \eta) \setminus B(v, \widehat{d}(u, v) - \eta - \widehat{\varepsilon}) . \tag{18}$$

Any point u' in U gives an approximation of u's position with error at most the Euclidean diameter of U, namely $\max_{u,u' \in U} \|u - u'\|$. We will show this diameter is bounded by (17).

We use some basic vector algebra. Let $\varepsilon' = \widehat{\varepsilon} + 2\eta \le 2(\widehat{\varepsilon} + \eta)$. For any $u, u' \in U$ we have, for all $v \in \{x, y, z\}$,

$$-\varepsilon' \le \|u - v\| - \|u' - v\| \le \varepsilon' .$$

Since the triangle x, y, z is acute, at least one of its sides makes an angle φ with the vector $u - u'$ where $0 \le \varphi \le \pi/4$. Taking this side to be (x, y) we have, without loss of generality,

$$(y - x) \cdot (u - u') = \|y - x\| \, \|u - u'\| \cos\varphi \ge \|y - x\| \, \|u - u'\| \sqrt{\frac{1}{2}}.$$

Next, we rewrite this dot product as follows,

$$\begin{aligned} 2(y - x) \cdot (u - u') &= \|x - u\|^2 - \|x - u'\|^2 - \|y - u\|^2 + \|y - u'\|^2 \\ &= (\|x - u\| - \|x - u'\|)(\|x - u\| + \|x - u'\|) \\ &\quad - (\|y - u\| - \|y - u'\|)(\|y - u\| + \|y - u'\|) \\ &\le \varepsilon'(\|x - u\| + \|x - u'\| + \|y - u\| + \|y - u'\|) , \end{aligned}$$

where the first line is a classical polarization identity. Putting these together, we have

$$\|u - u'\| \le \frac{\sqrt{2}}{\|y - x\|} \, \varepsilon' \big(\|x - u\| + \|x - u'\| + \|y - u\| + \|y - u'\|\big) \le \frac{4\sqrt{2} R \varepsilon'}{\ell} ,$$

completing the proof with $C_5 = 8\sqrt{2}$. ◀

Finally we state our main theorem.

▶ **Theorem 16.** *Let $r = n^\alpha$ for a constant $0 < \alpha < 1/2$. There is an algorithm with running time $O(n^2)$ that w.h.p. reconstructs the vertex positions of a random geometric graph $G \in \mathcal{G}(n, r)$, modulo symmetries of the square, with distortion d^* an absolute constant times times $\widehat{\varepsilon}$ as defined in* (16)*. That is, for some constant C'',*

$$d^* = C'' \begin{cases} n^{\frac{1}{2} - \frac{4}{3}\alpha} & \text{if } \alpha < 3/8 , \\ \sqrt{\log n} & \text{if } 3/8 \le \alpha < 1/2 . \end{cases}$$

Proof sketch. We use the fact, proved in [11], that w.h.p. the true positions of the lowest-degree vertices are within $\sqrt{\log n}$ of the corners of the square. Call these vertices a, b, c, d.

1. Find a good triple x, y, z. One way to do this is to find a vertex x near the center of the square, for instance one such that $d_G(x, t) \ge 0.65\sqrt{n}/r$ for all $t \in \{a, b, c, d\}$. Then find a y with $d_G(x, y) \in [0.1\sqrt{n}/r, 0.14\sqrt{n}/r]$, and then find a z such that $d_G(x, z), d_G(y, z) \in [0.1\sqrt{n}/r, 0.14\sqrt{n}/r]$. At each stage of this process, such a vertex exists with high probability, and all three are deep.

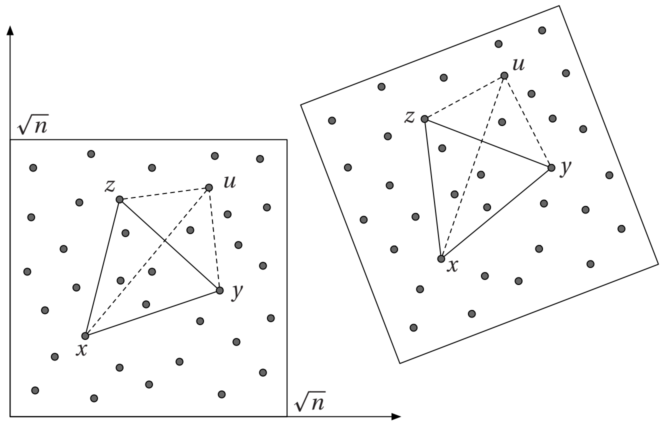

Figure 6 Our reconstruction is built around a triangle x, y, z of deep vertices. It may be translated, rotated, or reflected in $\mathbb{R}^2$ by an isometry, but it can then be shifted to the square $[0, \sqrt{n}]^2$. Then it will be a good reconstruction up to a rotation or reflection of the square.

2. Construct a triangle $x, y, z \in \mathbb{R}^2$ which is congruent to the true positions of the vertices x, y, z within error $\eta = O(\widehat{\varepsilon})$.
3. For each $v \in \{x, y, z\}$, compute the hybrid distance estimate $\widehat{d}(u, v)$ for all u as follows. First, for each w such that $d_G(w, v) \leq 2$, compute $|N(w) \cap N(v)|$ and thus the short-range distance estimates $\tilde{d}(w, v)$. Then compute $\widehat{d}(u, v)$ for all u using Dijkstra's algorithm on the weighted graph G_v described in Remark 10.
4. Use Lemma 14 to reconstruct the position of each vertex u relative to triangle x, y, z with error $O(\widehat{\varepsilon})$. This gives us a reconstruction up to an isometry of $\mathbb{R}^2$ as shown in Figure 6.
5. Finally, rotate and translate this reconstruction to the square $[0, \sqrt{n}]^2$. We choose a mapping of a, b, c, d to the corners of the square arbitrarily, using distance estimates to deduce which pairs are diagonally opposite, and then translate and rotate them as close as possible to $\{0, \sqrt{n}\}^2$. Since our definition of distortion allows rotations and reflections of the square, this gives a reconstruction with distortion $d^* = O(\widehat{\varepsilon} + \sqrt{\log n}) = O(\widehat{\varepsilon})$.

Step 1 can be done by breadth-first search, first from a, b, c, d and then from x and y, and thus takes $O(n)$ time. Steps 2, 3, 4, and 5 require $O(n)$ calculations of finite precision using standard functions, for which $O(\log n)$ bits of accuracy suffices. Thus the running time is dominated by the three uses of Dijkstra's algorithm, one for each $v \in \{x, y, z\}$, giving a running time of $O(n^2)$. ◀

▶ Remark 17. Since the typical degree in the graph is $\pi r^2 = O(n^{2\alpha})$ where $\alpha < 1/2$, and since Dijkstra's algorithm in a graph with n vertices and m edges runs in time $O(m + n \log n)$, the running time is w.h.p. $O(n^{2\alpha} + 1) = o(n^2)$.

▶ Remark 18. Once we reconstruct the positions of all vertices, we can get a good estimate of $||u - v||$ by direct computation from their approximate coordinates for all pairs u, v, including those where neither u nor v is deep.

5 Extensions to Other Domains

Our results can be generalized from the square to a number of alternative domains for random geometric graphs, including higher-dimensional Euclidean spaces and some curved manifolds. Here we sketch extensions of our algorithm to the m-dimensional hypercube and to the sphere, solving an open problem posed in [11].

5.1 Reconstruction in higher-dimensional Euclidean space

The simplest generalization is where the underlying domain is $[0, n^{1/m}]^m \subset \mathbb{R}^m$, i.e., an m-dimensional hypercube with volume n. We assume that m is a constant that does not vary with n. As before, n points are scattered uniformly in the hypercube, pairs u, v are adjacent if they are within Euclidean distance r, and our goal is to reconstruct the points' positions based on the adjacency matrix of the graph.

The following lemma generalizes Lemma 3 to $\mathbb{R}^m$, giving the m-dimensional volume of a lens-shaped intersection of two balls.

▶ **Lemma 19.** *Let $B_1(x, r_1)$ and $B_2(y, r_2)$ be two overlapping balls in $\mathbb{R}^m$ with $r_1 \le r_2$. Consider the lens $L = B_1 \cap B_2$. Let δ be the width of L, i.e., $\delta = \min\{r_1 + r_2 - d, 2r_1\}$ where $d = \|x - y\|$. Then the volume V of L satisfies $V = \Theta\left(\delta^{\frac{m+1}{2}} r_1^{\frac{m-1}{2}}\right)$, where the constant in Θ depends only on m.*

Given this relation between the width and volume of the lens, analogously to Section 3, we can compute both short- and long-range estimates of the distance, and combine them into a hybrid estimate. The error in the hybrid estimate is given by the following theorem.

▶ **Theorem 20.** *Let $r = n^\alpha$ for a constant $0 < \alpha < 1/m$. For all pairs u, v where v is deep, define $\widehat{d}(u, v)$ be the hybrid estimate of the distance. Then with high probability, $\widehat{d}$ is an upper bound on the Euclidean distance $\|u - v\|$ with error*

$$\widehat{\varepsilon}(u, v) \le C_m \begin{cases} n^{\frac{1}{m} - \frac{2m}{m+1}\alpha} & \alpha < \frac{m+1}{2m^2}, \\ \sqrt{\log n} & \frac{m+1}{2m^2} \le \alpha < \frac{1}{m}, \end{cases} \tag{19}$$

for some dimension-dependent constant C_m.

We omit the details of the proof since it closely follows the steps in Section 3.

In order to use the hybrid estimates for reconstruction, we need to find an appropriate number of deep landmarks. Using linear algebra, it suffices to have $m + 1$ landmarks that form a non-degenerate simplex. As in Theorem 16, we find an approximately equilateral m-simplex, namely a set of $m + 1$ points whose graph distances are all roughly the same constant times the diameter $n^{1/m}/r$. We again triangulate the positions of the other points based on their distance estimates, giving a reconstruction up to an isometry of $\mathbb{R}^m$. It is then easy to compute an isometry that shifts the reconstructed hypercube to $[0, n^{1/m}]^m$ by identifying low-degree vertices with the 2^m corners.

Putting this all together gives the following reconstruction theorem for random geometric graphs in $[0, n^{1/m}]^m$. We omit further details of the proof.

▶ **Theorem 21.** *Let $r = n^\alpha$ for a constant $0 < \alpha < 1/m$. There is an algorithm with running time $O(n^2)$ that w.h.p. reconstructs the vertex positions of a random geometric graph, modulo symmetries of the hypercube, with distortion*

$$d^* \le C_m \begin{cases} n^{\frac{1}{m} - \frac{2m}{m+1}\alpha} & \text{for } \alpha < \frac{m+1}{2m^2}, \\ \sqrt{\log n} & \text{for } \frac{m+1}{2m^2} \le \alpha < \frac{1}{m}, \end{cases}$$

for some dimension-dependent constant C_m.

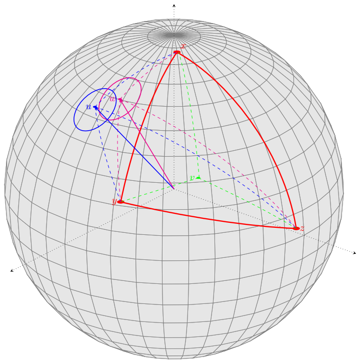

Figure 7 Using three landmarks x, y, z on the sphere to triangulate to other points in Theorem 22.

5.2 Reconstruction on the sphere

Finally, we argue that our algorithm also works on some curved manifolds and submanifolds where the geometric graph is defined in terms of geodesic distance. In particular we claim this for the m-dimensional spherical (hyper)surface S_m of a ball in $\mathbb{R}^{m+1}$. Here we sketch the proof for the two-dimensional surface of a sphere in $\mathbb{R}^3$. Note that the distortion is now defined by minimizing over the sphere's continuous symmetry group, i.e., over all rotations and reflections of the sphere.

In previous work, the authors of [10] gave a procedure to distinguish random geometric graphs on S^m from Erdős-Rényi random graphs. In addition, [3] gave a spectral method for reconstructing random graphs generated by a sparsified graphon model on the sphere, but since this model connects distant pairs of vertices with nonzero probability, it does not include the geodesic disk model we study here.

To define random geometric graphs on the sphere we scale the sphere so that its surface area is n, setting its radius to $R = \sqrt{n/(4\pi)}$. We scatter n points uniformly at random on it, or generate them with a Poisson point process with intensity 1, so that the expected number of points in a region is equal to its surface area. We define the graph as $(u, v) \in E$ if and only if $\|u - v\|_g \leq r$ where $\|u - v\|_g$ is the geodesic distance, i.e., the length of the shorter arc of a great circle that connects u and v. If we associate each point u with a unit vector $\vec{u} \in \mathbb{R}^3$ that points toward it from the center of the sphere, $\|u - v\|_g$ is R times the angle between $\vec{u}$ and $\vec{v}$.

▶ **Theorem 22.** *Let $r = n^\alpha$ for a constant $0 < \alpha < 1/2$. There is an algorithm with running time $O(n^2)$ that with high probability reconstructs the vertex positions of a random geometric graph, modulo a rotation or reflection of the sphere, with distortion an absolute constant times $n^{\frac{1}{2}-\frac{4}{3}\alpha}$ if $\alpha < 3/8$ and $\sqrt{\log n}$ if $\alpha \geq 3/8$.*

The algorithm is similar to that described in Theorem 16. The main difference is that our initial landmarks consist of three points x, y, z which approximately form a right spherical triangle, i.e., such that the vectors $\vec{x}, \vec{y}, \vec{z}$ have angles of about $\pi/2$ between them: see Fig 7.

6 Conclusion and Future Work

We have shown how a combination of geometric ideas can be used to reconstruct random geometric graphs with lower distortion than in previous work [11], achieving a distortion of $o(r)$ whenever $r = n^\alpha$ for $\alpha > 3/14$. Here we pose several questions for further work.

First, let us call a reconstruction ϕ *consistent* if its distances are consistent with the graph: that is, if $(u,v) \in E$ if and only if $\|\phi(u) - \phi(v)\| \leq r$. Even if ϕ has small distortion d^*, it might not be consistent: some edges $(u,v) \in E$ might have $\|\phi(u) - \phi(v)\|$ between r and $r + 2d^*$, and similarly some non-neighboring pairs might have $\|\phi(u) - \phi(v)\|$ between $r - 2d^*$ and r. To the best of our knowledge, even finding a single consistent embedding for random geometric graphs is an open question. It might be possible to refine our embedding to make it consistent, by using "forces" to move neighbors slightly closer together, and push non-neighbors farther away.

Second, a natural question is whether we can prove a significant lower bound on the distortion. An information-theoretic approach to this question would be to show that even the Bayesian algorithm, which chooses from the uniform measure on all consistent embeddings, has a typical distortion. We have been unable to prove this. However, here we sketch an argument that there *exist* consistent embeddings with a certain distortion by applying a continuous function f to the square $[0, \sqrt{n}]^2$ that "warps" the true embedding. If f's derivatives are at most δ in absolute value, then for each v, points close to the edge of v's neighborhood may move $O(\delta r)$ closer or farther away. However, a typical v has some $\varepsilon = O(1/r)$ for which there are no points whose distance is between $r - \varepsilon$ and $r + \varepsilon$, since the area of the corresponding annulus is $O(1)$. This suggests that if $\delta = O(\varepsilon/r) = O(1/r^2)$, the warped embedding is still consistent (except for a few vertices where we need to be more careful). On other other hand, even if f does not change the distance between nearby vertices very much, it can still move some vertices $\delta\sqrt{n}$ from their true positions, giving a distortion $d^* = \Omega(\sqrt{n}/r^2)$. If $r = n^\alpha$ this gives $\Omega(n^{1/2-2\alpha})$.

Even if this lower bound can be made rigorous, and even if it applies to typical consistent embeddings rather than just a few, there is a large gap between it and our upper bounds. Thus it is tempting to think that our algorithm can be improved, reducing the distortion still further. One approach would be to try to extend the geometry of overlapping disks in Theorem 2 to larger graph distances. Another would be to combine them with the spectral ideas of e.g. [3].

Finally, we would like to see how far these techniques can be extended to curved manifolds and submanifolds with boundary. In Theorem 22 we took advantage of the fact that the 2-sphere has a convenient embedding in $\mathbb{R}^3$. A more general approach, which we claim applies to any compact Riemannian submanifold with bounded curvature, would be to work entirely within the manifold itself, building a sufficiently dense mesh of landmarks and then triangulating within mesh cells. In particular, in the popular model of hyperbolic embeddings (e.g. [6, 16, 17]) where the submanifold is a ball of radius ℓ in a negatively curved space with radius of curvature R, we believe similar algorithms will work as long as $\ell/R = O(1)$. We leave this for future work.

References

1 Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018. URL: `http://jmlr.org/papers/v18/16-480.html`.

2 Ittai Abraham, Shiri Chechik, David Kempe, and Aleksandrs Slivkins. Low-distortion inference of latent similarities from a multiplex social network. *SIAM J. Comput.*, 44(3):617–668, 2015.

3 Ernesto Araya Valdivia and De Castro Yohann. Latent distance estimation for random geometric graphs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019. URL: `https://proceedings.neurips.cc/paper/2019/file/c4414e538a5475ec0244673b7f2f7dbb-Paper.pdf`.

4 Ery Arias-Castro, Antoine Channarond, Bruno Pelletier, and Nicolas Verzelen. On the estimation of latent distances using graph distances. *Electronic Journal of Statistics*, 15(1):722–747, 2021.

5 J. Aspnes, D.K. Goldenberg, and Y.R. Yang. On the computational complexity of sensor network localization. In *Proc. ALGOSENSORS-04*, pages 32–44, 2004.

6 Thomas Bläsius, Tobias Friedrich, Anton Krohmer, and Sören Laue. Efficient embedding of scale-free graphs in the hyperbolic plane. *IEEE/ACM Transactions on Networking*, 26(2):920–933, 2018.

7 M. Bradonjic, T. Elsasser, T. Friedrich, T. Sauerwald, and A. Stauffer. Efficient broadcast on random geometric graphs. In *21st. ACM-SIAM SODA*, pages 1412–1421, 2010.

8 H. Breu and D.G. Kirkpatrick. Unit disk graph recognition in np-hard. *Compt. Geometry Theory Appl.*, 9(1-2):3–24, 1998.

9 J. Bruck, J. Gao, and J. Jiang. On the computational complexity of sensor network localization. In *ACM MOBIHOC-05*, pages 181–192, 2005.

10 J. Bubeck, J. Ding, R. Eldan, and M. Racz. Testing for high-dimensional geometry in random graphs. *Random Structures and Algorithms*, 49(3):503–532, 2016.

11 Josep Diaz, Colin McDiarmid, and Dieter Mitsche. Learning random points from geometric graphs or orderings. *Random Structures & Algorithms*, 2019.

12 Josep Diaz, Dieter Mitsche, Guillem Perarnau, and Xavier Pérez-Giménez. On the relation between graph distance and Euclidean distance in random geometric graphs. *Advances in Applied Probability*, 48(3):848–864, 2016.

13 R. Ellis, J.L. Martin, and C. Yan. Random geometric graph diameter in the unit ball. *Algorithmica*, 47(4):421–438, 2007.

14 Edward. E. Gilbert. Random plane networks. *J. Soc. Industrial Applied Mathematics*, 9(5):533–543, 1961.

15 Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090+, December 2002.

16 Maksim Kitsak, Ivan Voitalov, and Dmitri Krioukov. Link prediction with hyperbolic geometry. *Phys. Rev. Research*, 2:043113, 2020.

17 Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, September 2010. doi:10.1103/PhysRevE.82.036106.

18 Shane Lubold, Arun G. Chandrasekhar, and Tyler H. McCormick. Identifying the latent space geometry of network models through analysis of curvature, 2021. arXiv:2012.10559.

19 Cristopher Moore. The computer science and physics of community detection: Landscapes, phase transitions, and hardness. *Bull. EATCS*, 121, 2017. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/480.

20 Elchanan Mossel, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461, 2015.

21 M. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties,. In *16th. ACM-SIAM SODA*, pages 989–998, 2005.

22 S. Parthasarathy, D. Sivakoff, M. Tian, and Y. Wang. A quest to unravel the metric structure behind perturbed networks. In *Proc of the 33rd SoCG*, pages 53:1–53:16, 2017.

23 Mathew Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.

24 Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W Moore. Theoretical justification of popular link prediction heuristics. In *Twenty-Second International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 2722–2727, 2011.

25 Cosma Rohilla Shalizi and Dena Asta. Consistency of maximum likelihood for continuous-space network models, 2019. arXiv:1711.02123.

26 M. Walters. Random geometric graphs. *Surveys in Combinatorics*, pages 365–402, 2011.

27 Yi-Jiao Zhang, Kai-Cheng Yang, and F. Radicchi. Systematic comparison of graph embedding methods in practical tasks, 2021. arXiv:2106.10198.