Low-distortion Inference of Latent Similarities from a Multiplex Social Network*

Ittai Abraham † Shiri Chechik † David Kempe ‡ Aleksandrs Slivkins †

Abstract

Much of social network analysis is — implicitly or explicitly — predicated on the assumption that individuals tend to be more similar to their friends than to strangers. Thus, an observed social network provides a noisy signal about the latent underlying "social space:" the way in which individuals are similar or dissimilar. Many research questions frequently addressed via social network analysis are in reality questions about this social space, raising the question of inverting the process: Given a social network, how accurately can we reconstruct the social structure of similarities and dissimilarities?

We begin to address this problem formally. Observed social networks are usually multiplex, in the sense that they reflect (dis)similarities in several different "categories," such as geographical proximity, kinship, or similarity of professions/hobbies. We assume that each such category is characterized by a latent metric capturing (dis)similarities in this category. Each category gives rise to a separate social network: a random graph parameterized by this metric. For a concrete model, we consider Kleinberg's small world model and some variations thereof. The observed social network is the unlabeled union of these graphs, i.e., the presence or absence of edges can be observed, but not their origins. Our main result is a near-linear time algorithm which reconstructs each metric with provably low distortion.

1 Introduction

Much of social network analysis is, implicitly or explicitly, predicated on the assumption that people tend to be more similar to their friends than to strangers. While many tasks — such as analyzing power and centrality, trading and exchange, or understanding and influencing the diffusion of viruses or information — rely crucially on the precise network structure, many others — such as link prediction, identification of communities, or marketing to friends of past buyers — use network structure as a noisy signal about an underlying social similarity space. To illustrate this insight differently, consider altering a social network data set by removing links between "dissimilar" pairs of individuals, and inserting instead links between "similar" (but previously unconnected) pairs. If this change makes the analysis task easier, rather than impossible, then the analysis task is really about the "social structure" — the latent similarities and dissimilarities between individuals — rather than about the actual network structure.

Given the abundance of important problems naturally phrased in terms of social structure (discussed in more detail below), it is a natural goal to explicitly reconstruct social structures from a given social network. Knowing the social structure may also be of independent interest, as it sheds light on the forces governing social link formation.

The task of inferring social structure in this sense is made non-trivial by the following two obstacles. First, despite a general tendency for friends to be more similar than strangers, many friends are still sufficiently different from each other to look essentially random. Second, and perhaps more fundamentally, social networks are *multiplex* [15, 39, 51]: they tend to be the union of multiple often independent relations among the same actors. For instance, friendships could result from physical proximity, similarity of occupation, kinship, similarities of hobbies, etc. If individuals are very similar in even one such attribute, they are more likely to be connected.

The main contribution of this paper is a near-linear time algorithm for reconstructing the latent social structure with provably low distortion. The model explicitly produces a union of graphs, one for each category, and an important feature of the algorithm is that it separates the different graphs from each other. We also provide two extensions which, respectively, further improve the distortion, and partially address the issue of data scarcity (i.e., very small node degrees). The algorithms in this paper are based on, and significant extensions of, a natural idea that is widely used in practice: nodes are likely to be close if they share many common neighbors.

An overview of the model. We posit a latent space model (described in detail in Section 2) for the generation of social networks akin to models widely used

*The full version of this paper [1] is available on `arxiv.org`.

†Microsoft Research Silicon Valley, Mountain View CA, USA. Email: {ittaia,schechik,slivkins}@microsoft.com.

‡Dept. of Computer Science, University of Southern California, Los Angeles CA, USA. Email: dkempe@usc.edu. Research done in part while visiting Microsoft Research, Silicon Valley.

in the mathematical sociology, statistics, and computer science communities [12, 20, 23, 24, 25, 27, 31, 41, 43, 44, 47] (see also the survey [50, pages 15–21]).

The model is based on two widely accepted tenets about social networks (e.g., [8, 37]). First, people are more likely to have ties with those who are similar to them, but also have many ties to others who are dissimilar.[1] Second, multiple social dimensions (such as geography, occupation, kinship, hobbies, etc.) can independently lead to interactions and the formation of ties.

We call the social dimensions along which people can be (dis)similar *(social) categories*, to avoid confusion with the geometric dimensions of individual metric spaces. Each category is given by a metric space $\mathcal{D}_i, i = 1, \ldots, K$; together, the $\mathcal{D}_i$ define the *social distances* between the individuals. Each of the n individuals occupies a point in each of the categories. For concreteness, and in accordance with much of the preceding literature, we assume that each category is a Euclidean space of known dimensionality [23, 24, 27, 31, 41, 43], and that the density of the points corresponding to individuals is nearly uniform [24, 27, 43]. Furthermore, we assume that the categories have small local correlation. The "local correlation" of two categories is the maximal overlap between any two small balls in those categories. (See Equation (2.1) in Section 2.)

Each category independently gives rise to a social network $\mathcal{G}_i$, modeled as a random graph whose edge distribution is parameterized by the corresponding metric space $\mathcal{D}_i$. Specifically, we use a slight variation of Kleinberg's small-world model [27], in which edge probabilities decrease polynomially in $\mathcal{D}_i(u, v)$. For our purposes, the key feature of the model is that the probability of shorter links is much higher, but long-range links also appear with a significant probability; this captures the first tenet. The algorithm observes the *union* $\mathcal{G} = \bigcup_i \mathcal{G}_i$ of the individual networks $\mathcal{G}_i$ (on the same node set), but does not learn which *particular* network(s) $\mathcal{G}_i$ an edge belonged to. This captures the second tenet; only the existence, but not the social "origins," of ties can be observed.[2] *The algorithm's goal is to use $\mathcal{G}$ to reconstruct the individual metrics $\mathcal{D}_i$ with small distortion, with high probability (over the random network generation process).*

Importantly, social similarity spaces in general tend not to be metrics (see, e.g., [10]), in the sense that the triangle inequality fails to hold. The main reason is the presence of multiple social categories. For example, one's co-worker and one's relative could be very dissimilar to one another, even though the individual is similar to both. The inclusion of a union or minimum in the model is crucial to capture this.

Algorithms and results. Our main contribution is a near-linear time algorithm, called the *Amoeba algorithm*, which infers all individual categories with provably low distortion, with high probability. The following theorem captures the result slightly informally.

THEOREM 1.1. (INFORMAL) *If the K metric spaces $\mathcal{D}_i$ are locally sufficiently different, and the average node degrees are at least $\Omega(K^3 \log^2 n)$, then with high probability, the Amoeba algorithm, in near-linear time, reconstructs metrics $\mathcal{D}'_i$ such that $\mathcal{D}'_i$ approximates $\mathcal{D}_i$ with constant multiplicative distortion (and at most polylogarithmic additive error).*

That this approximate reconstruction should be possible at all — regardless of the running time — is somewhat surprising. One might think a priori that after combining two social networks, there would simply be no way to tease them apart.

In other words, a priori, the challenge appears to be information-theoretical (does the network contain enough information for distance reconstruction with any provable guarantees?) as much as computational. We also remark that even the single-category version was raised by Kleinberg [29] as an open question; we answer the reconstruction question in the positive even for multiple categories.

The Amoeba algorithm, we well as all other algorithms in this paper, is broadly based on a heuristic widely used in practice (e.g., in Facebook, or see [2, 33, 43, 46]): edges (u, v) are more likely to be between friends in a category if they are "supported" by many common neighbors of u and v in that category. However, to deal with multiple categories, low node degrees, or to sharpen the distance estimates, the basic idea of counting common neighbors needs to be extended significantly.

The Amoeba algorithm, presented and analyzed in detail in Section 3, consists of two stages. In a first stage, individual edges are pruned if they do not have enough common neighbors, a direct implementation of the common neighbors heuristic.[3] In the second stage,

[1] The model is agnostic about whether this similarity is caused more by *homophily* [32, 38] (the tendency to form ties with those who are similar) or by *social influence* [36, 42] (the tendency to *become* similar to one's associates).

[2] Our model does not include any information such as demographics, location, wall posts, or communications which would frequently be available to social networking sites [5]. Our goal here is to understand at a fundamental level how much information on social structures can be inferred algorithmically from the observed social network alone.

[3] Sarkar et al. [43] showed that under a model similar to

which we call *the Amoeba stage*, basic estimates of the individual categories are constructed one by one. Each iteration starts with a polylog-sized clique in the graph computed by the first stage, which is then expanded one edge at a time: an edge (u, v) is added to a category only when enough of u's neighbors lie in a small ball around v according to the current estimate of the category. The basic idea is that any sufficiently large clique must be sufficiently close in one category. The clique then bootstraps further iterations, in that a node u with many edges to a small ball around v must itself be close to v. While this intuition is straightforward, each iteration loses accuracy, so it takes a delicate proof to show that this refined version of the common neighbors heuristic guarantees low distortion.

We improve the main result in the following two directions. The first direction (Sections 4.1 and 4.2) focuses on improving the distortion using long-range links, which are now treated as an additional data source rather than an obstacle to be pruned. We improve the distortion from a multiplicative constant to a factor $1 + o(1)$, using a post-processing phase (run after the Amoeba algorithm) which we call *Two-Ball Algorithm.* This is a variation of the common neighbors heuristic where instead of common neighbors of two nodes (u, v), the algorithm counts links between two node sets. The node sets are low-radius balls around u and v according to the initial distance estimates. This result requires a stronger notion of low correlation between categories. Under a stronger uniform density conditions, the Two-Ball Algorithm can be applied recursively, yielding *unit* distortion (with at most polylogarithmic additive error).

Second (in Section 4.3), we deal with the issue of data scarcity, which in our setting translates to low node degrees. In the low (constant) node degree regime, the common neighbors heuristic is uninformative, and it instead becomes necessary to count disjoint constant-length paths for a suitably chosen constant. Combining the new initial pruning phase with a subsequent Two-Ball Algorithm requires a much more careful analysis, which shows that all sufficiently long edges can be treated as mutually independent given the pruned graph. We recover (essentially) all our results for the single-category case; extending the results to multiple categories remains a direction for future work.

These additional results are briefly described in Section 4; detailed proofs can be found in the full version of this paper [1].

Our algorithms are modular: a pre-processing step (counting common neighbors, or the low-degree algorithm of Section 4.3) prunes away very long edges. The Amoeba step separates different metrics and constructs initial distance estimates (though we have not adapted the algorithm and analysis to low node degrees). Finally, the Two-Ball Algorithm and its recursive version can be used to further improve the distortion in individual categories.

Discussion of the model. Our modeling goal is not to define a model of social networks capturing all of their features; this would be a formidable/impossible task for which there is much research but not much consensus. Instead, we aim for generally accepted modeling choices which capture in a clean way the main algorithmic challenges inherent in rigorous distance reconstruction. In particular, our main goal was to capture the two conceptual obstacles to distance reconstruction: links between dissimilar individuals, and multiple social categories. Nevertheless, we discuss some particular modeling choices in more detail.

1. In Kleinberg's small-world model [27, 26, 29, 18], a version of which we adapt as a generative model for individual categories, the probability for an edge between two nodes to exist decreases polynomially in the nodes' distance. Naturally, many other distributions lead to distance-based random graphs [7].

Much of the past work in the statistics community [23, 24, 31, 41, 43] assumed that the edge probabilities were logit-linear in the distance, i.e., that $\log(\frac{p}{1-p})$ is linear in $\mathcal{D}(u, v)$. Since long-range links are thus exponentially unlikely ($p = \frac{e^{-\alpha\mathcal{D}(u,v)}}{1+e^{-\alpha\mathcal{D}(u,v)}}$), the reconstruction task becomes much easier. More importantly, to the extent that precise distributions have been empirically tested, remarkable fits have been found [3, 5, 34] with Kleinberg's inverse polynomial distribution [27, 28].[4] Furthermore, our main constant-distortion result holds for a much more general class of distributions, including logit-linear distributions.

2. The choice of Euclidean spaces with near-uniform density. Both choices (Euclidean and near-uniform) are ubiquitous in past work[5] [20, 23, 24, 25, 27, 31, 41, 43], and are made mostly for technical convenience; they allow us to separate the conceptual difficulty of teasing apart different metrics and inferring distances with low distortion from the technical difficulty of dealing with arbitrary metric spaces. We believe that future work

ours (but using edge probabilities that decrease exponentially with distance), counting common neighbors leads to an accurate distance estimate for a single-category social network.

[4] However, links that appear long could plausibly be short in another metric; whether inverse polynomial distributions remain prevalent when multiple metrics are considered is an interesting — although difficult — direction for future empirical work.

[5] In many respects, our kind of latent space models deteriorate if node densities can be highly non-uniform [19].

will achieve similar results for more general metric spaces or related structures, in particular, ultrametrics [12, 28, 47], which are another popular choice of latent metric spaces.

3. The choice of a union or minimum to combine individual metrics. This choice is clearly a simplification of reality: individuals are more likely to form ties if they share similarities in multiple dimensions, e.g., they work in the same field *and* live in the same town. Our model is supposed to capture in the cleanest way the difficulty of separating edges originating from different categories, and is certainly a better approximation to reality than widely used models treating the social structure as one metric space.

Our model is closely related to (and a slight generalization of) a notion of social distance proposed by Watts, Dodds, and Newman [53], which treats the social distance as the minimum of distances in multiple metrics. To the extent that past work explicitly discussed models of multiple categories, it was also based on the minimum [23, pp. 337, 348], [47, p. 335]. A generalization to more realistic models is a natural direction for future work.

4. We capture a notion of "independence" between categories by requiring that small balls in different categories have small overlap. Even without restrictions on computational resources, some assumption about "independence" is clearly necessary: if categories could be extremely similar, then no low-distortion reconstruction seems possible. It is an interesting direction for future work whether a few isolated violations of the condition permit low-distortion reconstruction in all but the affected areas of the metric spaces.

Our condition is significantly weaker than requiring probabilistic independence. Several past papers (using a single metric space) assumed that nodes were placed independently and uniformly at random over some space [24, 43]; such a model of individual categories would imply our "small intersection" condition with high probability. In fact, we show (see the full paper for details) that with high probability, the "small intersection" condition holds even when nodes are placed adversarially, and their names are permuted randomly. We also remark that while in reality, we will frequently observe high correlation between "categories" (such as work and hobbies), this could be construed as a sign that the categories should be chosen differently, in order to represent the latent traits that manifest themselves in choices of both occupations and hobbies.

Applications. Our work provides two natural reconstruction abilities: separating edges by categories, and reconstructing individual categories with low distortion. Both of them have multiple useful applications.

Important industrial applications for social network information include improving ad placement (*social advertising*), web search results (*social search*), and product recommendations. These applications are of vital importance for some of the major players on the Web. A key commonality of all three applications is that they use the behavior of friends (clicking, searching, purchasing) to predict the behavior of an individual. Yet, two recent studies [21, 35] undertaking a quantitative evaluation of the predictive power of social links for purchases and click behavior have found at best mixed evidence.

This apparent conundrum is resolved by noticing that many links are long-range, and short-range links may be short in an irrelevant category for the prediction task. Indeed, a recent data-driven study by Tang and Liu [52] has shown that social link-based classifiers perform much better when edges are labeled with categories in which they are short. We conjecture that such classifiers would improve even further if instead of edges, the actual *social distance* between nodes were used.

The ability to separate social categories also enables the automatic detection of circles of friends from different contexts in social networking sites. This automatic detection has been cited as one of the main selling points of Google+, and is at the heart of the startup Katango. In this sense, our work provides some theoretical underpinnings for this fast-growing facet of the social networking market. Separating edges by categories has the additional benefit that one can identify when edges are short in more than one category, which could enable the automatic detection of close friends [54, 55].

Another natural application is the discovery of "social communities" [9, 16, 17, 13, 45]. One might argue that the plethora of different network community detection objectives and heuristics is largely a result of stating the objectives and algorithms in terms of the graph structure, when the goal is really to identify clusters in the metric spaces. Since the social space is rarely explicitly modeled or related to the network, the connection between the objective function and the actual desired object is absent. Explicitly reconstructing the social space would constitute the first step toward a more sound community identification algorithm. The presence of multiple categories in the model will naturally give rise to overlapping communities as well. Indeed, some of the work on reconstructing Euclidean spaces in the statistics community [23, 31] is explicitly motivated by the desire to identify communities, and builds community structure into a Bayesian prior.

Social distances can also be used to predict unobserved or potential social links. Link prediction has been studied in [2, 12, 33, 43, 46]. Unobserved or poten-

tial links are most likely present between node pairs at small distances; hence, once distances are known, missing links can be predicted easily [12, 43]. Indeed, explaining why popular heuristics, such as counting common neighbors, work in practice [2, 33, 46] was the main motivation for Sarkar et al. [43] to study latent Euclidean spaces.

Related work. Due to space constraints, an in-depth discussion of related work is deferred to the full paper.

A lot of recent work [5, 12, 20, 23, 24, 25, 31, 41, 44, 47] uses Bayesian Models or Maximum Likelihood Estimation to reconstruct metric spaces (mostly, but not exclusively, Euclidean). These papers do not model multiple categories, and they do not come with any guarantees on the quality of approximation of the inferred metric; in addition, their inference problems are often not tractable, and heuristics without guarantees even on likelihood or probability are used. The most notable exception is the work of Sarkar, Chakrabarti, and Moore [43], who are motivated by the goal of explaining why simple heuristics for link prediction, such as counting common neighbors, are successful. As part of their analysis, they show that for a single category with logit-linear edge probabilities, counting common neighbors gives accurate distance estimates.

There are conceptual similarities between the present paper and simultaneous independent work by Arora et al. [4] and Balcan et al. [6]. Their goal is to reconstruct overlapping community structure with provable guarantees. They posit latent set-based structures which can be interpreted as 0-1 metrics. Interestingly, they also require a "limited overlap" condition, and some of the algorithmic ideas used are similar. However, the reconstructed objects are different, and there is no analogue in their work to our post-processing steps and the algorithms we design for dealing with low degrees.

2 Definitions and Preliminaries

We define a formal model for the latent social space that gives rise to observed social networks. In general, it will not be a metric space: it naturally possesses multiple social dimensions, and proximity in just one of those dimensions (e.g., geography or occupation) usually means that individuals are "close."

First, we define a basic model of a single social metric space. We then discuss how to extend the concept to multiple metrics; in particular, we formalize a notion of metric spaces being sufficiently "independent."

Throughout, V is a *ground set* of n nodes. For a metric $\mathcal{D}$, we use the standard notion of balls: $B(u,r) = \{v \mid \mathcal{D}(u,v) \leq r\}$. We liberally use $O(\cdot)$ notation to simplify the presentation. In theorem statements, the constants in $O(\cdot)$ can depend on the constants in our setting. Elsewhere, the constants in $O(\cdot)$ are absolute, unless noted otherwise.

Most of our results are with high probability, with respect to the randomness in the graph generation process. By this, we mean that the success probabilities are $1 - n^{-c}$, where the constant $c \geq 1$ is large enough to allow all needed applications of the Union Bound (over polynomially many events). By a slight abuse of notation, we will write *with high probability* for probability $1 - n^{-c}$, without explicitly specifying the constant $c \geq 1$.

A model for one social category. A single category of the latent space is modeled essentially as a d-dimensional Euclidean space. More precisely, V is a subset of the d-dimensional *torus*[6], i.e., the nodes lie in $[0,R]^d$ for some R, and the distance between $x, y \in [0,R]^d$ is $\mathcal{D}(x,y) = (\sum_i (\min(|x_i - y_i|, R - |x_i - y_i|))^p)^{1/p}$. We require that the node density be *nearly uniform*, in the following sense: any unit cube in the torus contains between one and C_{UD} nodes, for some known constant $C_{\text{UD}} \geq 1$. (Since C_{UD} is always a constant, we sometimes hide C_{UD} factors in $O(\cdot)$ notation.) For some of our results, we also want to use the actual lattice structure as a reference: We refer to the graph of integer points from $[0,R]^d$ with edges between all pairs at distance $\mathcal{D}(x,y) \leq 1$ as the *toroidal grid*.

If nodes u, v are at distance $r = \mathcal{D}(u,v)$, then the edge (u,v) is present *independently of other edges*, with probability $f(r) = \min(1, C_{\text{sg}} k_{\text{sg}} r^{-d})$. Here, $C_{\text{sg}} = \Theta(\frac{1}{\log n})$ is a normalization constant chosen to ensure that the expected average node degree is 1 whenever $k_{\text{sg}} = 1$. Then, k_{sg} is a parameter controlling the expected average node degree. When $C_{\text{sg}} k_{\text{sg}} \leq 1$, the expected average degree is exactly k_{sg}; otherwise, the dependence of the node degree on k_{sg} is sublinear and strictly monotone. We call k_{sg} the *target degree*, even though strictly speaking, it does not equal the average degree. Following the literature (e.g., [27, 28]), we focus on the cases $k_{\text{sg}} = O(1)$ and $k_{\text{sg}} = \text{polylog}(n)$. We use E_{sg} to denote the edge set obtained from this distribution, and $\mathcal{G}(V, \mathcal{D}_i)$ for the random graph model, which we call the *single-category social graph*.

When $k_{\text{sg}} \geq 1/C_{\text{sg}}$, all edges of length at most 1 are present in E_{sg} with probability 1. Otherwise, even to ensure connectivity of the social graph, one must insert a suitable "local edge set" separately. (For instance, much of the literature on small-world networks assumes

[6] Prior work deals with a d-dimensional grid, which is somewhat undesirable, as there is an asymmetry between the nodes on the border and on the inside.

that the d-dimensional grid is always part of the graph.) This issue is discussed in more detail in Section 4.3, in the context of low node degrees.

Our main result easily extends to a more general model in which, for a suitably large $R = \text{polylog}(n)$, an edge (u, v) of length $r = \mathcal{D}(u, v)$ is present with probability at least $f(r)$ for all $r < R$, and with probability smaller than $f(r)$ for all $r \geq R$. We omit this generalization for ease of presentation.

Multiple social categories. When multiple social categories give rise to edges independently (such as work-related, geography-related, and hobby-related friends), we model the observed social network as the *union* of the graphs generated by the individual categories. Formally, each social category is a single-category social graph $\mathcal{G}_i = \mathcal{G}(V, \mathcal{D}_i)$ with near-uniform density for $i = 1, \ldots, K$, and the edge sets of the $\mathcal{G}_i$ are mutually independent. K is a (small) constant. Balls with respect to the category-i metric are denoted by $B_i(u, r)$. A *multi-category social graph* is obtained by taking the *union* of all edges, i.e., $E_{\text{sg}} = \bigcup_{i=1}^{K} E_{\text{sg}}^{(i)}$. Taking the union is analogous to defining the social distance as the minimum over the categories; in particular, the social space thus defined is not a metric.

The different categories may have different parameters, such as the target degree or number of dimensions. If the target degrees are vastly different, then one category could be completely "drowned out" by other, denser, categories, which would make it impossible to observe its structure. Therefore, we assume that the target degrees $k_{\text{sg}}^{(i)}$ of the categories are within a known constant factor of one another. We define *the* target degree of the multi-category social graph as the average $k_{\text{sg}} = \frac{1}{K} \cdot \sum_i k_{\text{sg}}^{(i)}$.

Local Disjointness of Categories. In order to be able to distinguish the edges arising from different categories, it is necessary that the underlying metrics of different categories be sufficiently different. We capture this intuition by requiring that any pair of small balls in two different categories be sufficiently different: formally, the *Local Category-Disjointness condition* states that for any two balls $B_i(u, r)$, $B_{i'}(u', r')$ in distinct categories $i \neq i'$, with $r, r' = O(\text{polylog}(n))$,

$$|B_i(u, r) \cap B_{i'}(u', r')| \leq O(\log n). \tag{2.1}$$

This condition suffices for our main result; some of the extensions require a similar but stronger local condition called Scale-R Category-Disjointness, which will be introduced in Section 4.2. The Local Category-Disjointness condition is not overly strong; for instance, we prove (details to be found in the full paper) that both Local Category-Disjointness and Scale-R Category-Disjointness hold with high probability when node identifiers within each category are randomly permuted.

Input and output. Since our model has several parameters, we need to be precise about what is known to the algorithm. Most importantly, in terms of the social network, only the union E_{sg} of all social network edges is revealed to the algorithm; the division into individual categories $E_{\text{sg}}^{(i)}$ is not given.

We assume that the algorithm knows how many embeddings it needs to construct, and into what spaces. More formally, this means that K (the number of categories), d_i (the number of dimensions), and R_i (the sizes of the tori) are known to the algorithm. The average target degree k_{sg} can be estimated from the expected degree, and by Chernoff Bounds, such an estimate will be within $1 \pm O(n^{-1/2})$ of the correct value with high probability. According to the model, the individual target degrees $k_{\text{sg}}^{(i)}$ lie within a constant factor of k_{sg}, and we assume that this constant factor is also known to the algorithm. To simplify presentation, we assume that the target degrees $k_{\text{sg}}^{(i)}$ and the dimensions d_i are the same for all categories i, and that k_{sg} is known.

We also assume that the upper bound C_{UD} on the number of points in any unit cube is known to the algorithm. Knowing C_{UD} and the other model parameters, the normalization constant $C_{\text{sg}} = \Theta(\frac{1}{\log n})$ can also be computed to within a constant factor.

The goal of the algorithm is to output metrics $\mathcal{D}_i'$ that approximate the original $\mathcal{D}_i$. If the output satisfies

$$\sigma \mathcal{D}(u, v) \;\leq\; \mathcal{D}'(u, v) \;\leq\; \delta \mathcal{D}(u, v) + \Delta$$

for all node pairs u, v, then we say that $\mathcal{D}_i'$ estimates $\mathcal{D}_i$ with *contraction* σ, *expansion* δ and *additive error* Δ. The *multiplicative distortion* of $\mathcal{D}_i'$ is then δ/σ. If we mention no multiplicative distortion (or contraction), then we implicitly refer to the case of distortion (contraction) 1. We do not require that $\mathcal{D}_i'$ itself be a d_i-dimensional Euclidean metric, only that it approximate $\mathcal{D}_i$ with low distortion.

Chernoff bounds. In many places, we bound tail deviations using standard *Chernoff Bounds*. Specifically, we use the following version, which can be found, e.g., in [14, pages 6–8].

THEOREM 2.1. (CHERNOFF BOUNDS) *Let X be the sum of independent random variables distributed in $[0, 1]$, and let $\mu' \geq \mu = E[X]$. Then the following hold:*

$$\text{Prob}\,[\,|X - \mu| > \delta\mu\,] \leq e^{-\mu\,\delta^2/3} \qquad (\forall\, \delta > 0) \tag{2.2}$$

$$\text{Prob}\,[\,X > (1 + \delta)\mu'\,] \leq e^{-\mu'\,\delta^2/3} \quad (\forall\, \delta \in (0, 1)). \tag{2.3}$$

3 The main result

In this section, we present our main result, an algorithm for distance reconstruction for multiple categories with constant distortion.

THEOREM 3.1. *Consider a multi-category social graph with $C_{sg}k_{sg} = \Omega(\log n)$, near-uniform density and Local Category-Disjointness. There is an algorithm that with high probability reconstructs distances in each category with constant expansion, no contraction, and* $\mathrm{polylog}(n)$ *additive error. Moreover, such distance estimates (as spanner graphs or as distance labels) can be computed in time $O(n\,\mathrm{polylog}(n))$.*

3.1 Overview and intuition

We begin with a high-level overview of the algorithm and the intuition for the proof, before discussing the different stages in detail in individual subsections. Recall that the algorithm's input is the set $E_{\mathrm{sg}} = \bigcup_i E_{\mathrm{sg}}^{(i)}$ of edges from all categories. For the entire section, we assume that the average node degree is high enough: $C_{\mathrm{sg}}k_{\mathrm{sg}} = \Omega(16^d K^3 \log n)$ for a sufficiently large constant in $\Omega(\cdot)$. Let $r_{\mathrm{loc}} = \Theta((C_{\mathrm{sg}}k_{\mathrm{sg}})^{1/d})$ be the *local radius*: by definition of the generative model, all edges between node pairs (u, v) at distance $\mathcal{D}(u, v) \le r_{\mathrm{loc}}$ are in E_{sg} with probability 1. We define the *pruning radius* to be $r_{\mathrm{pru}} = \Theta(r_{\mathrm{loc}}K^{2/d})$.

The algorithm proceeds in multiple stages. Each of these stages makes use of the (random) long-range edges. To avoid stochastic dependencies between the stages, we can randomly partition the edges of E_{sg} into a constant number of sets. Each stage then makes use of its own set. Since the nodes' degrees are high enough, this does not affect the high-probability guarantees. For ease of notation, we will not explicitly talk about the partitions for the remainder of this section. All results in this section hold with high probability.

In the first stage, called the *Two-Hop Test,* the algorithm produces a *pruned set* E_{pru} (which need *not* be a subset of E_{sg}), with the following guarantee for all node pairs (u, u'):

- If u, u' are at distance at most r_{loc} in (at least) one category i, then $(u, u') \in E_{\mathrm{pru}}$.
- If u, u' are at distance at least r_{pru} in all categories i, then $(u, u') \notin E_{\mathrm{pru}}$.

Thus, the guarantee is that all short edges are present, and all sufficiently long edges are absent. The algorithm makes no guarantees for node pairs in the intermediate distance range.

To achieve this pruning, the Two-Hop Test counts the number of 2-hop paths (common neighbors) between (u, u'), and compares it to a carefully chosen threshold. Similar to what Sarkar et al. [43] showed for the single-category case and the logit-linear edge probabilities, our analysis shows that this simple heuristic can provide provable distortion guarantees under the small-world model, even in the more difficult case of multiple categories.

In the second stage, called *Amoeba stage*, the algorithm covers E_{pru} with individual edge sets $E_{\mathrm{amb}}^{(i)}$ (which need not be disjoint); the set $E_{\mathrm{amb}}^{(i)}$ corresponds to category i. The key property we prove is that whenever u, v are at distance at most r_{loc} in category i, then $(u, v) \in E_{\mathrm{amb}}^{(i)}$, whereas $(u, v) \notin E_{\mathrm{amb}}^{(i)}$ whenever u and v are at distance at least $r_{\mathrm{amb}} = \Theta(r_{\mathrm{pru}}K^{3/d}) = \Theta(r_{\mathrm{loc}}K^{5/d})$. Again, for the intermediate range, the algorithm makes no guarantees about the presence or absence of edges. This guarantee implies that the shortest-path metric of $E_{\mathrm{amb}}^{(i)}$ gives an embedding of $\mathcal{D}_i$ with constant multiplicative distortion $O(K^{5/d})$ for all node pairs at distance at least r_{loc}, and poly-logarithmic additive distortion for all node pairs at distance at most r_{loc}.

The algorithm constructs the edge sets $E_{\mathrm{amb}}^{(i)}$ one by one. For each i, it begins by finding a poly-logarithmically large clique in E_{pru} that is sufficiently spread out in all previously constructed $E_{\mathrm{amb}}^{(j)}$. (We show using the Local Category-Disjointness condition that the node set of this clique will have diameter at most $4r_{\mathrm{pru}}$ in some category i). Starting from this clique, as long as possible, it adds edges (u, v) that are "supported" by enough edges (in E_{sg}) between v's neighborhood in $E_{\mathrm{amb}}^{(i)}$ and u. The key part of our analysis is to show that this process will indeed add all sufficiently short edges (and in particular end up having added all nodes), while excluding all edges that are long in category i.

Throughout this section, we frequently count the number of edges in E_{sg} between two node sets (one of which may be a single node). We usually calculate the expectation, and then invoke Chernoff Bounds to guarantee that the number of edges is within the desired range. The expectation or desired number of edges will be (at least) logarithmic, allowing the application of Chernoff Bounds.

3.2 Pruning stage: the Two-Hop Test

For a node pair u, v, let $M_\Lambda(u, v)$ be the number of two-hop u-v paths in E_{sg}, i.e., the number of common neighbors of u and v in E_{sg}. The Two-Hop Test is as follows:

$$\text{for each pair } (u, u'), \text{ accept if } M_\Lambda(u, u') \ge M_\Lambda, \text{ reject otherwise.} \tag{3.4}$$

We define the threshold as $M_\Lambda = \Theta(k_{\text{sg}} C_{\text{sg}})$, where the constant in $\Theta(\cdot)$ can be calculated explicitly from the known parameters. Henceforth, let E_{pru} be the set of all accepted node pairs.

LEMMA 3.1. *With high probability, the Two-Hop Test accepts all node pairs of distance at most r_{loc} in some category, and rejects all node pairs whose distance is at least r_{pru} in all categories.*

Proof. The proof is based on a careful decomposition of the metric space into intersections of rings around u and u', allowing a sufficiently accurate estimate of the number of their common neighbors.

We begin by proving the positive (acceptance) part. If u, u' are at distance $\mathcal{D}_i(u,u') \leq r_{\text{loc}}$, then they are close enough such that the balls $B_i(u, r_{\text{loc}})$ and $B_i(u', r_{\text{loc}})$ overlap in a (dimension-dependent) constant fraction of their nodes. Counting the size of this overlap, and using that $r_{\text{loc}} = \Theta((k_{\text{sg}} C_{\text{sg}})^{1/d})$, we get that

$$\begin{aligned} |B_i(u, r_{\text{loc}}) \cap B_i(u', r_{\text{loc}})| &\geq \Omega(2^{-d} |B_i(u, r_{\text{loc}})|) \\ &\geq \Omega(2^{-d} \Theta((k_{\text{sg}} C_{\text{sg}})^{1/d})^d) \\ &\geq \Omega(k_{\text{sg}} C_{\text{sg}}), \end{aligned}$$

for a sufficiently large constant in the definition of r_{loc}. In the original model, each edge between u or u' and a node in $B_i(u, r_{\text{loc}}) \cap B_i(u', r_{\text{loc}})$ is present with probability 1. Even if the edge set is randomly partitioned into a constant number of edge sets for the different stages of the algorithm, both u and u' will have edges to each node in $B_i(u, r_{\text{loc}}) \cap B_i(u', r_{\text{loc}})$ independently with constant probability. An application of the Chernoff Bound therefore guarantees that $M_\Lambda(u,u') > \Omega(k_{\text{sg}} C_{\text{sg}})$ with high probability, and $M_\Lambda = \Omega(k_{\text{sg}} C_{\text{sg}})$ for a suitably chosen constant.

For the second part of the lemma (rejection), fix two nodes u, u' such that $\mathcal{D}_i(u,u') > r_{\text{pru}}$ for all categories i. Consider two categories i, i' ($i = i'$ is possible), and let $S_{i,i'}$ be the set of all nodes v such that $(u,v) \in E_{\text{sg}}^{(i)}$ and $(u', v) \in E_{\text{sg}}^{(i')}$. We prove a high-probability bound of $O(C_{\text{sg}} k_{\text{sg}} / K^2)$ on $|S_{i,i'}|$ for a suitably small (absolute) constant in the $O(\cdot)$. A union bound over all K^2 pairs i, i' then implies the claim.

We define a sequence of concentric rings of exponentially increasing radius around u, as follows:

$$\begin{aligned} R_0 &= B_i(u, r_{\text{pru}}/2) \\ R_j &= B_i(u, 2^{j/d} \cdot r_{\text{pru}}/2) \setminus B_i(u, 2^{(j-1)/d} \cdot r_{\text{pru}}/2) \\ &= \{v \mid \mathcal{D}_i(u,v) \in (2^{(j-1)/d} \cdot r_{\text{pru}}/2,\ 2^{j/d} \cdot r_{\text{pru}}/2)\}, \\ &\quad \text{for each } j \geq 1. \end{aligned}$$

So R_j is the set of nodes at distance roughly $2^{j/d} \cdot r_{\text{pru}}/2$ from u in category i. Likewise, we define the concentric rings around u', with respect to category i':

$$\begin{aligned} R_0 &= B_{i'}(u', r_{\text{pru}}/2) \\ R_j &= B_{i'}(u', 2^{j/d} \cdot r_{\text{pru}}/2) \setminus B_{i'}(u', 2^{(j-1)/d} \cdot r_{\text{pru}}/2) \\ &\quad \text{for each } j \geq 1. \end{aligned}$$

The rings $\{R_j\}_{j \geq 0}$ form a disjoint cover of V, as do the rings $\{R'_j\}_{j \geq 0}$. To bound the size of $S_{i,i'}$, we bound $S_{i,i'} \cap R_j \cap R'_{j'}$ for all $j, j' \geq 0$.

First consider the case $j = j' = 0$. For $i = i'$, R_0 and R'_0 are disjoint by definition, and for $i \neq i'$, the Local Category-Disjointness condition ensures that $|R_0 \cap R'_0| = O(\log n)$.

Next, we consider the case $j \geq j',\ j \geq 1$. (The case $j' \geq j,\ j' \geq 1$ is symmetric.) We write $r = 2^{j/d} \cdot r_{\text{pru}}/2$ and $r' = 2^{j'/d} \cdot r_{\text{pru}}/2$. By definition of the edge generation model, the probability that $v \in R_j$ has an edge to u in $E_{\text{sg}}^{(i)}$ is at most $C_{\text{sg}} k_{\text{sg}} (r/2^{1/d})^{-d} = 2\, C_{\text{sg}} k_{\text{sg}} r^{-d}$, while the probability that $v \in R'_{j'}$ has an edge to u' in $E_{\text{sg}}^{(i')}$ is at most $2\, C_{\text{sg}} k_{\text{sg}} (r')^{-d}$, or at most 1 if $j' = 0$. The presence of these edges is independent of one another. Because $R_j \cap R'_{j'}$ is contained in $B_{i'}(u', r')$, it can contain at most $C_{\text{UD}}(r')^d = O((r')^d)$ nodes.[7] Thus, both for the case $j' = 0$ and $j' > 0$, we obtain that

$$\begin{aligned} \mathbb{E}\left[\, |S_{i,i'} \cap R_j \cap R'_{j'}| \,\right] &\leq O\left((C_{\text{sg}} k_{\text{sg}})^2\, r^{-d} (r')^{-d} (r')^d\right) \\ &\leq O\left((C_{\text{sg}} k_{\text{sg}})^2\, (2^{j/d} \cdot r_{\text{pru}}/2)^{-d}\right) \\ &\leq O\left((C_{\text{sg}} k_{\text{sg}})^2\, 2^d\, r_{\text{pru}}^{-d} \cdot 2^{-j}\right). \end{aligned}$$

We now first sum over all $j \geq j'$ (using that $\sum_{j \geq j'} 2^{-j} = O(2^{-j'})$), and then over all j', to obtain that

$$\begin{aligned} \sum_{j,j' :\, j + j' > 0} \mathbb{E}\left[\, |S_{i,i'} \cap R_j \cap R'_{j'}| \,\right] \\ \leq O((C_{\text{sg}} k_{\text{sg}})^2\, 2^d\, r_{\text{pru}}^{-d}). \end{aligned}$$

By choosing $r_{\text{pru}} = \Theta(r_{\text{loc}} K^{2/d})$ with a suitably large (absolute) constant, we can cancel out the 2^d term and obtain an arbitrarily small absolute constant γ in the $O(\cdot)$ term. Recalling that $r_{\text{loc}} = \Theta((C_{\text{sg}} k_{\text{sg}})^{1/d})$ and adding the at most $O(\log n)$ nodes (with some absolute constant) in $S_{i,i'} \cap R_0 \cap R'_0$, we see that

$$\mathbb{E}\left[|S_{i,i'}|\right] \leq O(\gamma C_{\text{sg}} k_{\text{sg}} / K^2) + O(\log n).$$

Applying Chernoff Bounds, we obtain that with high probability, $|S_{i,i'}| = O(\gamma\, C_{\text{sg}} k_{\text{sg}} / K^2 + \log n)$, and

[7] Recall that we include C_{UD} terms in $O(\cdot)$.

a union bound over all i, i' now shows that with high probability we have

$$M_\Lambda(u,v) = O(\gamma\, C_{\rm sg} k_{\rm sg} + K^2 \log n) < M_\Lambda$$

(when $C_{\rm sg}k_{\rm sg}$ is large enough and γ small enough), which means that (u,v) will be rejected. ∎

For the remainder of this section, we condition on the high probability event of Lemma 3.1, i.e., we assume that $E_{\rm pru}$ contains all edges of length at most $r_{\rm loc}$ (in at least one category) and no edges whose length would exceed $r_{\rm pru}$ in all categories.

Notice that in the single-category case ($K = 1$), the result of Lemma 3.1 by itself already gives an expansion of $r_{\rm pru}/r_{\rm loc} = \Theta(1)$, no contraction, and additive error polylog(n). We simply estimate $\mathcal{D}(u,v)$ by the length of the shortest u-v path in the pruned graph, multiplied by $r_{\rm pru}$. Lemma 3.2 analyzes the distortion for a single category, and will also be used for the multi-category case. The lemma requires the unit-disk graph to be a good approximation of the metric space, a property that is obvious for near-uniform density sets in $\mathbb{R}^d$.

LEMMA 3.2. *Let $(V,\mathcal{D})$ be a metric space. Let G be a graph on V that includes all node pairs at distance at most r and no node pairs at distance more than r', for some $r' > r \geq 1$. Let $\mathcal{D}_G$ be the shortest-paths metric of G. Let $\mathcal{D}^{\rm sp}$ be the shortest-paths metric of the unit disk graph on (V,D), and assume that $\mathcal{D}^{\rm sp}(u,v) \leq c\,\mathcal{D}(u,v)$ for all node pairs (u,v), for some constant c. Then*

$$\mathcal{D}(u,v) \;\leq\; r' \cdot \mathcal{D}_G(u,v) \;\leq\; \tfrac{cr'}{r} \cdot \mathcal{D}(u,v) + r'.$$

In words, $r' \cdot \mathcal{D}_G$ reconstructs $\mathcal{D}$ with expansion $\frac{cr'}{r}$, no contraction, and additive error r'.

Proof. Fix a node pair (u,v), and let ρ be a shortest u-v path in G. By the triangle inequality, $\mathcal{D}(u,v)$ is a lower bound on the total metric length of ρ, which in turn is at most $r'\,\mathcal{D}_G(u,v)$, because each hop in G has length at most r'. So $\mathcal{D}(u,v) \leq r'\,\mathcal{D}_G(u,v)$. Now, let P be a shortest u-v path in $\mathcal{D}^{\rm sp}$. Any two nodes on P that are within r hops from one another are connected by an edge in G. Therefore, G contains a u-v path of at most $\lceil \frac{|P|}{r} \rceil$ hops, which implies that $\mathcal{D}_G(u,v) \leq \lceil \frac{\mathcal{D}^{\rm sp}(u,v)}{r} \rceil \leq 1 + \frac{c\,\mathcal{D}(u,v)}{r}$. ∎

3.3 Amoeba stage: map edges to categories

We now define the Amoeba stage of the algorithm. The Amoeba stage consists of K iterations $i = 1, \ldots, K$: in each successive iteration i, a new category is identified (and re-numbered as category i), and some edges in $E_{\rm pru}$ are mapped to this category. These edges constitute the edge set $E^{(i)}_{\rm amb}$. Eventually, each edge $e \in E_{\rm pru}$ is mapped to at least one category.

The Amoeba stage is summarized in Algorithm 1. Each iteration i consists of an *initialization phase*, in which we find a suitable clique in $E_{\rm pru}$, and a *growth phase*, in which we grow $E^{(i)}_{\rm amb}$ one edge at a time. We think of this process as *growing the amoeba*.

In Algorithm 1 and the subsequent analysis thereof, we use the following notation. For a subset $S \subseteq V$, let $\mathrm{diam}_j(S)$ be its diameter in $E^{(j)}_{\rm amb}$. Let $\Gamma(v,E)$ denote the (1-hop) neighborhood of node v in the edge set E. We call the clique C from iteration i the *seed clique* for category i. The condition (3.5) is called the *Amoeba Test:* more precisely, edge (u,v) passes the test if and only if (3.5) is satisfied.

Algorithm 1 The Amoeba algorithm.

Output. Estimated social distance $\mathcal{D}'_i$, for each category $i = 1, \ldots, K$.

Parameters. Numbers $(M_\Lambda, M_{\rm amb}, N_{\rm amb}, r_{\rm amb})$.

Pruning Stage. Let $M_\Lambda(u,u')$ be the number of common neighbors of u and u' in $E_{\rm sg}$.

$$E_{\rm pru} \leftarrow \{(u,u') \in V \times V :\; M_\Lambda(u,u') \geq M_\Lambda\}.$$

Amoeba Stage. For each iteration $i = 1, \ldots, K$,

1. *Initialization phase.* Find any clique $C \subseteq V$ in $E_{\rm pru}$ such that $|C| \geq N_{\rm amb}$, and $\mathrm{diam}_j(C) \geq \log^2(n)$ for each category $j = 1, \ldots, i-1$.
   Initialize $E_{\rm amb} = C \times C$.

2. *Growth phase.* While there exists an edge $(u,v) \in E_{\rm pru} \setminus E_{\rm amb}$ such that

   (3.5) $E_{\rm sg}$ contains at least $M_{\rm amb}$ edges between u and $\Gamma(v, E_{\rm amb})$,

   pick any such edge and insert it into $E_{\rm amb}$.

3. Set $E^{(i)}_{\rm amb} = E_{\rm amb}$. Let $\mathcal{D}'_i$ be the shortest-paths metric of $E^{(i)}_{\rm amb}$, multiplied by $r_{\rm amb}$.

Notation. Recall that $\mathrm{diam}_j(S)$ is the diameter of a subset $S \subseteq V$ in $E^{(j)}_{\rm amb}$, and $\Gamma(v,E)$ denotes the (1-hop) neighborhood of node v in the edge set E. Condition (3.5) is called the *Amoeba Test*.

The Amoeba stage is parameterized by numbers $(M_{\rm amb}, N_{\rm amb}, r_{\rm amb})$. We set $N_{\rm amb} = \Theta((r_{\rm loc}/2)^d)$ and $M_{\rm amb} = \Theta(N_{\rm amb}/(8^d K^2))$ for suitable constants in $\Theta(\cdot)$. We define $r_{\rm amb} = \gamma_{\rm amb} \cdot K^{3/d} \cdot r_{\rm pru}$ for a sufficiently large

absolute constant γ_{amb}, and call it the *amoeba radius*.[8]

3.4 Analysis of the Amoeba stage

An edge $(u,v) \in E_{\text{pru}}$ is called *i-long* if $\mathcal{D}_i(u,v) > r_{\text{amb}}$, and *$i$-short* if $\mathcal{D}_i(u,v) \leq r_{\text{loc}}$. An edge set $E_{\text{amb}} \subseteq E_{\text{pru}}$ is an *i-amoeba* iff (V, E_{amb}) contains no i-long edges, and it contains a clique of at least N_{amb} nodes whose category-i diameter is at most $4r_{\text{pru}}$.

The high-level outline of the correctness proof for Amoeba is as follows. We will prove by induction on i that each edge set $E^{(i)}_{\text{amb}}$ captures (at least) all i-short edges (renumbering the categories appropriately), and does not include any i-long edges.

The induction step requires that the algorithm be able to reconstruct another category i while there is an uncovered edge. Thereto, we show that E_{amb} remains an i-amoeba throughout the algorithm. We break the induction step into multiple lemmas capturing the following four key points:

- The seed clique C of size N_{amb} exists in E_{pru}.
- All edges in C have sufficiently small length.
- No i-long edge passes the Amoeba Test.
- If there is an i-short edge not yet added to E_{amb}, at least one such edge passes the Amoeba Test.

LEMMA 3.3. *If there is an edge e not included in any $E^{(j)}_{amb}$, then E_{pru} contains a clique of at least N_{amb} nodes whose diameter in $E^{(j)}_{amb}$ is at least $\log^2(n)$ for all $j < i$.*

Proof. Let $e \in E_{\text{pru}}$ be an edge not included in $E^{(j)}_{\text{amb}}$ for all $j < i$, and let i be a category it belongs to. For an arbitrary node u, consider $B = B_i(u, r_{\text{loc}}/2)$. Because $\mathcal{D}_i(v,v') \leq r_{\text{loc}}$ for all $v, v' \in B$, the set B forms a clique in E_{pru}. Furthermore, because of the near-uniform density of category i, B has $\Theta((r_{\text{loc}}/2)^d) = \Theta(C_{\text{sg}} k_{\text{sg}}) = \Omega(K^3 \log n)$ nodes, for a sufficiently large constant in the $\Omega(\cdot)$.

For any $j < i$, the Local Category-Disjointness condition condition implies that $|B_j(u, r_{\text{amb}} \cdot \log^2(n)) \cap B| \leq O(\log n)$. Thus, there is at least one node $v \in B \setminus B_j(u, r_{\text{amb}} \cdot \log^2(n))$. Because each edge in $E^{(j)}_{\text{amb}}$ has length at most r_{amb} in category j, this means that $\mathcal{D}_j(u,v) > \log^2(n)$; in particular, B cannot have diameter less than $\log^2(n)$ in $E^{(j)}_{\text{amb}}$. Since this holds for all j, B is a candidate for seed clique i, and the algorithm thus guarantees progress. ∎

[8] Recall that $k_{\text{sg}} C_{\text{sg}} = \Omega(16^d K^3 \log n)$ with a sufficiently large constant. In particular, if $k_{\text{sg}} C_{\text{sg}} = \Theta(16^d K^3 \log n)$, then the parameters are $N_{\text{amb}} = \Theta(8^d K^3 \log n)$, $M_{\text{amb}} = \Theta(K \log n)$ and $r_{\text{amb}} = \Theta(K^8 \log n)^{1/d}$.

LEMMA 3.4. *Let C be a clique in E_{pru} of size $|C| > \Omega(K^3 \log n)$, for a sufficiently large constant in $\Omega(\cdot)$. Then, there exists a category i such that $\mathcal{D}_i(u,v) \leq 4r_{pru}$ for all $u, v \in C$.*

Proof. Fix an arbitrary $w \in C$. Because each edge $(u,v) \in E_{\text{pru}}$ satisfies $\mathcal{D}_i(u,v) \leq r_{\text{pru}}$ for some category i, there is a category i such that for at least $|C|/K$ nodes $v \in C$, we have $\mathcal{D}_i(w,v) \leq r_{\text{pru}}$. Fix such a category i, and let S be the set of all $v \in C$ with $\mathcal{D}_i(w,v) \leq r_{\text{pru}}$. If $S = C$, then we are done.

Otherwise, consider a node $u \in C \setminus S$. For each node $v \in S$, there is a category i' with $\mathcal{D}_{i'}(u,v) \leq r_{\text{pru}}$. In particular, there must be a category i' such that $\mathcal{D}_{i'}(u,v) \leq r_{\text{pru}}$ for at least $|C|/K^2 > \Omega(\log n)$ nodes $v \in S$, with a large enough constant in $\Omega(\cdot)$. Fix such a category i', and let S' be the set of nodes $v \in S$ with $\mathcal{D}_{i'}(u,v) \leq r_{\text{pru}}$. Because $S' \subseteq B_i(w, r_{\text{pru}}) \cap B_{i'}(u, r_{\text{pru}})$, the assumption $i' \neq i$ would contradict the Local Category-Disjointness condition. Hence $i' = i$, and u is at distance at most $2r_{\text{pru}}$ from w in category i. Since this argument holds for every $u \in C \setminus S$, we have proved that C has diameter at most $4r_{\text{pru}}$ in category i. ∎

LEMMA 3.5. *Assume that $E_{amb} \subseteq E_{pru}$ contains no i-long edge, and let u, v be nodes with $(u,v) \in E_{pru}$ and $\mathcal{D}_i(u,v) > r_{amb}$. Then, with high probability, (u,v) does not pass the Amoeba Test.*

Proof. We bound the number of edges between u and $\Gamma(v, E_{\text{amb}})$ in two parts: by the number of edges between u and $B_i(v, r_{\text{pru}})$, and the number of edges between u and $\Gamma(v, E_{\text{amb}}) \setminus B_i(v, r_{\text{pru}})$.

First, $|\Gamma(v, E_{\text{amb}}) \setminus B_i(v, r_{\text{pru}})| \leq O(K \log n)$. The reason is that any node $w \in \Gamma(v, E_{\text{amb}}) \setminus B_i(v, r_{\text{pru}})$ must be at distance at most r_{pru} from v in some category $j \neq i$ (because $(v,w) \in E_{\text{pru}}$), so $w \in B_j(v, r_{\text{pru}}) \cap B_i(v, r_{\text{amb}})$. Now, the Local Category-Disjointness condition implies that there can be at most $O(\log n)$ such nodes w for any fixed j, and thus at most $O(K \log n)$ total.

Next, we consider nodes $w \in B_i(v, r_{\text{pru}})$. By the Local Category-Disjointness condition for $B_i(v, r_{\text{pru}}) \cap B_j(u, r_{\text{amb}})$, there can be at most $O(\log n)$ such nodes w at distance at most r_{amb} from u in category j, for a total of $O(K \log n)$ nodes.

All other nodes $w \in B_i(v, r_{\text{pru}})$ are at distance at least r_{amb} from u in all categories $j \neq i$, and at distance at least $r_{\text{amb}} - r_{\text{pru}} \geq r_{\text{amb}}/2$ from u in category i. Thus, the probability for the edge (u,w) to exist in any one category j is at most $q = O(C_{\text{sg}} k_{\text{sg}} r^{-d}_{\text{amb}}) = O(C_{\text{sg}} k_{\text{sg}} / (\gamma^d_{\text{amb}} K^3) \cdot r^{-d}_{\text{pru}})$. Summing over all $w \in B_i(v, r_{\text{pru}})$ and all categories gives us at most $qK|B_i(v, r_{\text{pru}})| = O(C_{\text{sg}} k_{\text{sg}} / (\gamma^d_{\text{amb}} K^2))$ edges in expectation, and Chernoff Bounds prove concentration.

Adding the at most $O(K \log n)$ edges of the first two types, and recalling that γ_{amb} is a suitably large constant and $C_{\text{sg}} k_{\text{sg}} = \Omega(K^3 \log n)$ with a large constant, we see that with high probability, the total number of edges between u and $\Gamma(v, E_{\text{amb}})$ is less than M_{amb}, so the edge (u, v) does not pass the Amoeba Test. ∎

LEMMA 3.6. *Let E_{amb} be an i-amoeba that does not include all i-short edges. Then, w.h.p., there exists an edge $(u, v) \in E_{pru}$ that is accepted by the Amoeba Test.*

Proof. First notice that because the Amoeba Test only counts edges from u to a neighborhood of v, it is monotone in the following sense: if the edge e passes for some current edge set E_{amb}, then it also passes for any $E'_{\text{amb}} \supseteq E_{\text{amb}}$. We will define an ordering $e_1, e_2, \ldots$ of all edges in category i such that with high probability, e_ℓ will pass the Amoeba Test whenever $C \cup \{e_1, \ldots, e_{\ell-1}\} \subseteq E_{\text{amb}}$. Thus, Amoeba, starting from C, can always make progress when considering the lowest-numbered edge e_ℓ not yet included. (Notice that this does not require the algorithm to actually know the ordering.)

Let C be the clique in (V, E_{amb}) of size at least N_{amb} whose existence is guaranteed by the definition of an i-amoeba. $C \subseteq B_i(w, 2r_{\text{pru}})$ for some w, and $B_i(w, 2r_{\text{pru}})$ can be covered by $O((r_{\text{pru}}/r_{\text{loc}})^d) = O(K^2)$ balls of radius $r_{\text{loc}}/2$, at least one of which must therefore contain a sub-clique $C' \subseteq C$ of at least N_{amb}/K^2 nodes. Let v_0 be the center of such a ball $B_i(v', r_{\text{loc}}/2)$.

First, all edges between $u \in B_i(v_0, r_{\text{loc}}/2)$ and $v \in C'$ will pass the Amoeba Test, because (u, w) is i-short for all $w \in C' \subseteq \Gamma(v, E_{\text{amb}})$ (implying that the edge (u, w) is in E_{pru}), and $|C'| \geq N_{\text{amb}}/K^2 \geq M_{\text{amb}}$.

Second, because each $v \in B_i(v_0, r_{\text{loc}}/2)$ is now connected to all of C' in E_{amb}, the exact same argument applies to all node pairs $u, v \in B_i(v_0, r_{\text{loc}}/2)$.

Third, we use induction on r, showing that once all edges in $B_i(v_0, r)$ have been included, all edges in $B_i(v_0, r+1)$ will be included next in some order. For the base case, we use $r = r_{\text{loc}}/2$. Let u be any node in $B_i(v_0, r+1) \setminus B_i(v_0, r)$, and w a node "close to u on the line from v_0 to u." More formally, w is a node with $\mathcal{D}_i(v_0, w) \leq r - r_{\text{loc}}/4$ and $\mathcal{D}_i(u, w) \leq r_{\text{loc}}/4 + O(1)$. The existence of w follows by the near-uniform density assumption.

By near-uniform density, $B' = B_i(w, r_{\text{loc}}/4)$ contains at least $\Omega(2^{-d} N_{\text{amb}})$ nodes, and by induction hypothesis, all nodes of B' are neighbors of v. Furthermore, E_{sg} contains edges between u and all w with constant probability, so using Chernoff Bounds, with high probability, the pair (u, v) will pass the Amoeba Test for all $v \in B'$, inserting all these edges. Once all i-short edges between $u \in B_i(v_0, r+1)$ and $v \in B_i(v_0, r)$ have been inserted, the i-short edges between the remaining pairs $u, v \in B_i(v_0, r+1)$ will be inserted by the following argument. Node u has i-short edges to all nodes in B' (which are already in E_{amb}), so $\mathcal{D}_i(v, w) \leq 2r_{\text{loc}}$ for all $w \in B'$. Thus, each edge from v to $w \in B'$ is included with probability at least $p = \Omega(C_{\text{sg}} k_{\text{sg}} 2^{-d} r_{\text{loc}}^{-d})$, and there are at least $|B'| \geq \Omega(4^{-d} r_{\text{loc}}^d)$ such nodes, implying that the expected number of edges between v and the neighborhood of u is at least $\Omega(8^{-d} C_{\text{sg}} k_{\text{sg}})$. By Chernoff Bounds, we obtain concentration results, and because $M_{\text{amb}} \leq \Theta(8^{-d} C_{\text{sg}} k_{\text{sg}})$, the edge (u, v) will be included with high probability. ∎

The algorithm will thus terminate with i-amoebae $E^{(i)}_{\text{amb}}, i = 1, \ldots, K$. The distance $\mathcal{D}_i(u, v)$ is now estimated as the shortest-path distance between u and v in $E^{(i)}_{\text{amb}}$, multiplied by r_{amb}. By Lemma 3.2, this gives constant expansion $r_{\text{amb}}/r_{\text{loc}} = \Theta(K^{5/d})$, no contraction, and additive error r_{amb}.

3.5 Efficient implementation

We outline how to implement the Amoeba algorithm in near-linear time. The first (and perhaps most surprising) step is quickly finding the seed clique. Then, we need to execute each Amoeba step in (amortized) polylogarithmic time. The resulting algorithm computes the graph $E^{(i)}_{\text{amb}}$ for each category i in near-linear time. Recall that $E^{(i)}_{\text{amb}}$ is a constant-distortion *spanner* for $\mathcal{D}_i$, in the sense that its shortest-path metric approximates $\mathcal{D}_i$. Once we have a spanner, we can compute succinct distance labels by adapting a hierarchical beaconing technique from prior work on distance labeling and routing schemes (e.g. [22, 11, 48, 49]). We next describe each of these steps in more detail.

Finding the seed clique. By suitably adjusting the threshold M_Λ, the Two-Hop Test can be modified to accept all node pairs that are within distance $r'_{\text{loc}} = 3\, r_{\text{pru}}$ in some category, and to reject all node pairs that are at distance at least $r'_{\text{pru}} = \Theta(K^{2/d}\, r'_{\text{loc}})$ in all categories. We run the Amoeba algorithm on the pruned graph E'_{pru} obtained by this modified Two-Hop Test. Let r'_{amb} be the corresponding Amoeba radius. To produce the seed cliques for E'_{pru}, we use the original Two-Hop Test in the way described below.

Consider the original Two-Hop Test, and let E_{pru} be the corresponding pruned graph. Let $N(u)$ denote the 1-hop neighborhood of node u in E_{pru}, including u itself. For a node set S, define $N(S)$ to be the *intersection* $N(S) \triangleq \bigcap_{u \in S} N(u)$. We focus on such intersections for node sets $S \subseteq N(u)$ of size $|S| = K$.

LEMMA 3.7. *For any node u and category i, there exists a set $S \subseteq N(u)$ of size K such that the intersection $N(S)$ contains at least N_{amb} nodes, has diameter at most $3\,r_{pru}$ in category i, and diameter at least $R = r'_{amb}\log^2(n)$ in all other categories.*

Proof. Let $B = B_i(u, r_{\text{loc}}/2)$. We show that there exists a candidate set $S \subseteq B$. Recall that B induces a clique in the pruned graph E_{pru}, so for any subset $S \subseteq B$, we have $B \subseteq N(S)$. Since B contains at least N_{amb} nodes and has diameter at least R in each category $j \neq i$, $N(S)$ inherits these properties. Thus, it remains to ensure that $N(S)$ has low diameter in category i.

We claim that Local Category-Disjointness implies the existence of a subset $S \subseteq B$ of size K, such that any two nodes in S are at distance at least $2\,r_{\text{pru}}$ in each category $j \neq i$. Consider (for the proof only) the following simple algorithm. The algorithm works with two set-valued variables, S and U, initialized to $S = \emptyset$ and $U = B$. It runs the following loop K times: pick any node $v \in U$, add this node to S, and remove from U all balls $B_j(v, 2\,r_{\text{pru}}), j \neq i$. Clearly, the following invariant is maintained after each iteration: any two nodes $v \in S, w \in S \cup U$ are at distance at least $2\,r_{\text{pru}}$ in any category $j \neq i$. Therefore, the algorithm finds the desired set S unless U were to become empty prematurely. This cannot happen because by Local Category-Disjointness, B and any $B_j(v, 2\,r_{\text{pru}}), j \neq i$ overlap in at most $O(\log n)$ nodes, so the cardinality of U decreases by at most $O(K \log n)$ in each iteration.

Now fix the subset S guaranteed by the previous paragraph. Consider some node $w \in N(S)$. For any category $j \neq i$, there can be at most one node in S within category-j distance r_{pru} from w. (If there were two such nodes $v, v' \in S$ then $\mathcal{D}_j(v, v') \leq r_{\text{pru}}$, a contradiction.) It follows that at least one node $v \in S$ is at distance more than r_{pru} from w in *each* category $j \neq i$. Since the pruned graph E_{pru} contains the edge (v, w), v and w must be close in some category, and we have proved that they can only be close in category i. Therefore $\mathcal{D}_i(v, w) \leq r_{\text{pru}}$. Since $S \subseteq B$, it follows that $\mathcal{D}_i(u, w) \leq r_{\text{pru}} + r_{\text{loc}}/2$. Therefore, any two nodes in $N(S)$ are at category-i distance at most $2\,r_{\text{pru}} + r_{\text{loc}}$ from one another. ∎

For each iteration i of the Amoeba Stage, we need to find a seed clique C for E'_{pru} such that $|C| \geq N_{\text{amb}}$ and $\text{diam}_j(C) \geq \log^2(n)$, for each category $j < i$. By Lemma 3.7, one such clique is given by $N(S)$, for any given node u and some subset $S \subseteq N(u)$ of size K. Therefore, we can run the original Two-Hop Test to obtain the pruned graph E_{pru}, pick any node u, and iterate through all K-node subsets $S \subseteq N(u)$ until we find a set S such that $N(S)$ is a clique in E'_{pru}. It is easy to see that this approach results in running time $n\,\text{polylog}(n)$. In fact, one only needs the initial pruning step to be local to node u, so the list of all candidate subsets $N(S)$ can be obtained in $\text{polylog}(n)$ time.

Efficient implementation of the Amoeba step. To implement the Amoeba step efficiently, we use a queue which initially contains all edges. In each Amoeba step, edges are popped from the queue until one is found that satisfies Condition (3.5). Once an edge (u, v) satisfies this condition, it is added to the amoeba, while all its adjacent edges are (re-)enqueued. Any one edge is adjacent to at most polylogarithmically many other edges, and can therefore be enqueued at most polylogarithmically many times. Thus, the entire growth phase of the Amoeba algorithm is implemented in $n\,\text{polylog}(n)$ running time. The following argument shows the correctness of this queue policy: If an edge (u, v) is checked and does not satisfy Condition (3.5), then it can satisfy this condition at some later point only if another edge incident to u or v has been added to the Amoeba, i.e., only if (u, v) is re-enqueued.

From a spanner to succinct distance labels. Fix a category i. For the remainder of this section, all "balls" and "distances" refer to category i. We use the spanner $E_{\text{amb}} = E^{(i)}_{\text{amb}}$ produced by the Amoeba algorithm to produce *distance labels* for $\mathcal{D}_i$ of polylogarithmic size, so that for any two nodes u, v the distance $\mathcal{D}_i(u, v)$ can be estimated with constant distortion from their labels alone (in polylogarithmic time).

Consider exponentially increasing distance scales r. For each distance scale r, pick k_r *scale-r beacon nodes* independently and uniformly at random; k_r is chosen so that with high probability, each ball of radius r contains $\Theta(\log n)$ scale-r beacon nodes; For each scale-r beacon b, run a breadth-first search in E_{amb} for $\Theta(r)$ steps, to compute distance estimates between b and all nodes within distance $\Theta(r)$ from b. Simple accounting shows that computing the estimates for all scales and all beacons takes $n\,\text{polylog}(n)$ time.

Thus, for every given node u, we have computed estimates for distances between u and some subset S_u of beacons. S_u includes all scale-r beacons within distance $\Theta(r)$ from u, for each scale r. Together, these distance estimates constitute u's distance label. Given the distance labels of two nodes u and v, one can reconstruct the distance estimate for the pair (u, v) by picking the beacon $b \in S_u \cap S_v$ closest to node u, and using the distance estimate for the pair (b, v) as an estimate for (u, v).

4 Overview of the additional results

This section contains a *succinct* account of the extensions outlined in the introduction. The details, including all proofs, can be found in the full version [1].

We improve the main result in two directions: improving the distortion from a multiplicative constant to a factor $1 + o(1)$, and handling the case of low (constant) degree.

4.1 Improving the distortion: single category

In trying to improve the distortion beyond a multiplicative constant, we face an immediate obstacle: as discussed in Section 2, an algorithm can estimate the normalization constant $C_{\rm sg}$ and the target degree $k_{\rm sg}$ only up to a constant factor. However, for further improvements of the distortion, more accurate estimates of $C_{\rm sg}$ and $k_{\rm sg}$ appear to be necessary. In order to side-step this technical obstacle, we define *normalized distances*

$$\mathcal{N}(u,v) = \mathcal{D}(u,v)/(C_{\rm sg}\, k_{\rm sg})^{1/d}$$

, and we focus on $\mathcal{N}$ instead of actual distances as the quantities to be inferred.

Note that Theorem 3.1 can also be interpreted to yield an estimate $\mathcal{N}^*$ for $\mathcal{N}$ which with high probability has no contraction, constant expansion and polylog(n) additive error. We improve this bound to unit distortion with sub-linear additive error.

THEOREM 4.1. *Consider a single-category social graph of dimension d, with $C_{sg}k_{sg} = \Omega(\log n)$ and near-uniform density. There is a polynomial-time algorithm that w.h.p. reconstructs each normalized distance $\mathcal{N}(u,v)$ with additive error $\pm\mathcal{N}^{\gamma}\log^{O(1)} n$, where $\gamma = \frac{d+2}{2d+2}$. The algorithm runs in polynomial time.*

The high-level idea is to augment the Two-Hop Test from Section 3 with a post-processing step we call *Two-Ball Algorithm.* This is a variation of the common neighbors heuristic where instead of common neighbors, the algorithm counts 3-hop paths whose first and last hops are sufficiently short according to the initial estimates. More precisely, to estimate $\mathcal{N}(s,t)$, the algorithm counts edges between two node sets $\tilde{B}^*_s$ and $\tilde{B}^*_t$ that are small balls (centered at s and t, respectively) with respect to the initial estimates $\mathcal{N}^*$.

The Two-Ball Algorithm proceeds as follows. The input consists of $\mathcal{N}^*$ and the original edge set $E_{\rm sg}$. For every two nodes s and t, the normalized distance $\mathcal{N}(s,t)$ is estimated as follows. Let $\tilde{B}_u(\kappa;\mathcal{N}^*)$ be the set of the κ closest nodes to node u according to $\mathcal{N}^*$, breaking ties arbitrarily; note that this set is — up to tie-breaking — a ball with respect to $\mathcal{N}^*$. Consider balls $\tilde{B}^*_s = \tilde{B}_s(\kappa;\mathcal{N}^*)$ and $\tilde{B}^*_t = \tilde{B}_t(\kappa;\mathcal{N}^*)$, for some cardinality κ to be specified later. Count the number of edges in $E_{\rm sg}$ between $\tilde{B}^*_s$ and $\tilde{B}^*_t$, and let $\tilde{M}_{s,t}$ be that number. The new estimate is

$$\mathcal{N}'(s,t) = \left(\kappa^2/\tilde{M}_{s,t}\right)^{1/d}.$$

We take $\kappa = r_x^d$, where $r_x \triangleq x^{(d+2)/(2d+2)}$ and $x = \mathcal{N}^*(s,t)$. See Algorithm 2 for the pseudocode.

Algorithm 2 The Two-Ball Algorithm.

Inputs. Original edge set $E_{\rm sg}$ and initial estimates $\mathcal{N}^*$ from Theorem 3.1.

Output. Improved distance estimates $\mathcal{N}'$.

For each node pair (s,t):

1. $\tilde{B}^*_s = \tilde{B}_s(\kappa;\mathcal{N}^*)$ and $\tilde{B}^*_t = \tilde{B}_t(\kappa;\mathcal{N}^*)$, where $\kappa = x^{d(d+2)/(2d+2)}$ and $x = \mathcal{N}^*(s,t)$.
2. $\tilde{M}_{s,t}$ is the number of edges in $E_{\rm sg}$ between $\tilde{B}^*_s$ and $\tilde{B}^*_t$.
3. $\mathcal{N}'(s,t) = (\kappa^2/\tilde{M}_{s,t})^{1/d}$.

Notation. $\tilde{B}_u(\kappa;\mathcal{N}^*)$ is the set of the κ closest nodes to u according to $\mathcal{N}^*$, breaking ties arbitrarily.

The idea is that $\mathbb{E}\left[\tilde{M}_{s,t}\right] \approx \kappa^2\,\mathcal{N}^{-d}(s,t)$, and our estimate inverts this relation. We pick κ to optimize the trade-off between the "spatial uncertainty" (pairwise distances between nodes in $\tilde{B}^*_s$ and $\tilde{B}^*_t$ are not exactly $\mathcal{N}(s,t)$) and "sampling uncertainty" (deviations of the number of edges from the expectation). The former increases with κ, and the latter decreases with κ.

Recursive two-ball algorithm. Given that the Two-Ball Algorithm produces improved estimates of (normalized) distances, it seems natural to run the algorithm again, using the improved estimates as a starting point for defining the balls $\tilde{B}^*_s$ and $\tilde{B}^*_t$ more accurately. This suggests a recursive approach: to estimate $\mathcal{D}(s,t)$, the algorithm can use the previously computed estimates for smaller distance scales to define $\tilde{B}^*_s$ and $\tilde{B}^*_t$. We call the resulting algorithm (with carefully optimized distance scales) the *Recursive Two-Ball Algorithm.* The technical goal is to improve the additive error in Theorem 4.1.

The analysis of this algorithm is significantly more delicate. In particular, in order to take advantage of the improved estimates, a stronger uniformity condition is needed on the metric: we say that the metric space has *perfectly uniform density* iff each ball of radius r contains $C_{\rm PD}\, r^d \pm O(r^{d-1})$ points, where $C_{\rm PD}$ is a known constant. Then we can improve the additive error to polylog(n).

THEOREM 4.2. *Consider a single-category social graph with $C_{sg}k_{sg} = \Omega(\log n)$ and perfectly uniform density. Assume that the social distance is defined by the ℓ_2^d norm, with $d > 2$. Then, the Recursive Two-Ball Algorithm w.h.p. reconstructs all normalized distances with unit distortion and additive error* polylog(n).

Remark. The algorithm uses a constant c_d that captures, up to the first-order term, how the expected number of edges between two radius-r balls depends on r and the distance between centers. Specifically, in the setting of Theorem 4.2, consider two radius-r balls whose centers are at distance $x > 4r$. The expected number of edges between these two balls is $(c_d\, r^2/x)^d$, up to a multiplicative factor $1 + O(r^{-2})$. Here, c_d is a constant that depends only on the dimension d and the constant C_{PD} in the definition of perfectly uniform density. We assume that c_d is known to the algorithm.

The restriction to the ℓ_2 norm is essential to define c_d: under ℓ_p, $p \neq 2$, the expected number of edges between the two balls significantly depends on the alignment of the s-t line relative to the coordinate axes.

Remark. For $d = 2$, a similar (but slightly more complicated) algorithm and analysis yield additive error $2^{O(\sqrt{\log x})}$ for node pairs at normalized distance x; we omit the details.

We next define the algorithm. Let us first set up the notation. Let $\mathcal{N}^*$ be the normalized distance estimates guaranteed by Theorem 3.1. We will compute refined estimates $\mathcal{N}'$, which are initialized to $\mathcal{N}^*$. Let $\tilde{B}_u(\kappa; \mathcal{N}')$ be the set of the κ closest nodes to u according to $\mathcal{N}'$, breaking ties arbitrarily.

The Recursive Two-Ball Algorithm proceeds as follows. The input consists of $\mathcal{N}^*$ and the original edge set E_{sg}. The algorithm considers node pairs (s,t) such that $\mathcal{N}^*(s,t) > \text{polylog}(n)$, in order of increasing $\mathcal{N}^*$. For each such node pair, we define balls around s and t whose radius is roughly $\hat{r}_x$, where $x = \mathcal{N}^*(s,t)$ and $\hat{r}_x = x^{1/2+1/d}$. Formally, we define balls $\tilde{B}'_s = \tilde{B}_s(\kappa; \mathcal{N}')$ and $\tilde{B}'_t = \tilde{B}_t(\kappa; \mathcal{N}')$, where $\kappa = C_{\text{PD}}\, \hat{r}_x^d$. Note that these balls are defined with respect to the improved estimates $\mathcal{N}'$. Let $\tilde{M}_{s,t}$ be the number of edges between $\tilde{B}'_s$ and $\tilde{B}'_t$. The new estimate is $\mathcal{N}'(s,t) = c_d\, \hat{r}_x^2\, \tilde{M}_{s,t}^{-1/d}$. The pseudocode is shown in Algorithm 3. Note that the algorithm is quite simple; the only complication is how to pick κ as a function of $x = \mathcal{N}^*(s,t)$.

Algorithm 3 The Recursive Two-Ball Algorithm.

Inputs. Original edge set E_{sg} and initial estimates $\mathcal{N}^*$ from Theorem 3.1.

Output. Improved distance estimates $\mathcal{N}'$.

$\mathcal{N}' \leftarrow \mathcal{N}^*$.

For each node pair (s,t) such that $\mathcal{N}^*(s,t) > \text{polylog}(n)$, in order of increasing $\mathcal{N}^*$:

1. $\kappa = C_{\text{PD}}\, \hat{r}_x^d$, where $x = \mathcal{N}^*(s,t)$ and $\hat{r}_x = x^{1/2+1/d}$.
2. $\tilde{B}'_s = \tilde{B}_s(\kappa; \mathcal{N}')$ and $\tilde{B}'_t = \tilde{B}_t(\kappa; \mathcal{N}')$.
3. $\tilde{M}_{s,t}$ is the number of edges in E_{sg} between $\tilde{B}'_s$ and $\tilde{B}'_t$.
4. $\mathcal{N}'(s,t) = c_d\, \hat{r}_x^2\, \tilde{M}_{s,t}^{-1/d}$.

Notation. $\tilde{B}_u(\kappa; \mathcal{N}')$ is the set of the κ closest nodes to node u according to $\mathcal{N}'$, breaking ties arbitrarily. c_d is the constant from the remark after Theorem 4.2.

Overview of the analysis. Let $a(x)$ be the maximum additive error for node pairs at normalized distance at most x. As in the Two-Hop Test, the error comes from two sources: spatial uncertainty and sampling uncertainty. We show that the spatial uncertainty can contribute at most $O(a(\hat{r}_x))$ to the overall additive error; interestingly, this holds for any choice of $\hat{r}_x$. We use Chernoff Bounds to bound the contribution of sampling uncertainty by $O(a(\hat{r}_x))$ as well; this is where the particular exponent in $\hat{r}_x$ is used. It follows that $a(x) = O(a(\hat{r}_x))$. Finally, the distance estimates for a given node pair implicitly rely on recursion from distance scale x to distance scale $\hat{r}_x$. Let $\rho(x)$ be the depth of this recursion: the number of steps until the distance scale goes below polylog(n). It is easy to see that $a(x) = 2^{O(\rho(x))}$ and that $\rho(x) = O(\log \log n)$.

4.2 Improving distortion: multiple categories

In order to improve the estimates for multiple categories, we employ the two algorithms from Section 4.1. The main difference with the single-category case is that when we count the number of edges between the balls in the original multi-category social graph graph for some category i, some of these edges may come from other categories, which might affect the estimation. We would like to claim that the number of edges from other categories between the two balls is small compared to the number of edges from category i. Unfortunately, such a claim does not follow from the Local Category-Disjointness condition, which prompts the following stronger condition.

The stronger condition, called Scale-R Category-Disjointness, states that at all scales up to R, categories look essentially "random" with respect to one another. More specifically, given a pair of balls B, B' in some category i, we count the number of node pairs (u,u'), $u \in B$, $u' \in B'$ such that u and u' are close in some

other category j:

$$\#\texttt{pairs}_j(B, B', r) \triangleq |\{(u, u') \mid u \in B,\, u' \in B',\, \mathcal{D}_j(u, u') < r\}|. \tag{4.6}$$

If the node identifiers within each category are permuted randomly, then the expected number of such node pairs is $\Theta(r^d/n) \cdot |B|\,|B'|$, and with high probability, the deviations are bounded by:

$$\#\texttt{pairs}_j(B, B', r) \leq O(\tfrac{r^d}{n})\,|B|\,|B'| + O(\log^2 n). \tag{4.7}$$

Scale-R Category-Disjointness asserts that (4.7) holds "locally:" at all distance scales up to R.

DEFINITION 4.1. *The* Scale-R Category-Disjointness *condition states that* (4.7) *holds for any two categories $i \neq j$, any two disjoint category-i balls B, B' with $|B| \cdot |B'| \leq R^d$, and any $r \in (0, R]$.*

Remark. Equation (4.7) for randomly permuted categories is derived in the full version of the paper [1]. The expectation is relatively easy to derive, whereas the high-probability guarantee requires a more careful analysis. We obtain (a slightly weaker version of) Local Category-Disjointness as a special case if $R = \text{polylog}(n)$ and B is restricted to be a single node.

We will improve over the constant distortion under the condition above. We present two results: an extension of the Two-Ball Algorithm and an analysis of the Recursive Two-Ball Algorithm for multiple categories.

Like in the single-category case, we focus on normalized distances. For each category i, let $C_{\text{sg}}^{(i)}$ and $k_{\text{sg}}^{(i)}$ be the normalization constant and the target degree, respectively. The *normalized* category-i distance between nodes $u, v \in V$ is $\mathcal{N}_i(u, v) \triangleq \mathcal{D}_i(u, v)/(C_{\text{sg}}^{(i)}\, k_{\text{sg}}^{(i)})^{1/d}$.

The Extended Two-Ball Algorithm. The Scale-R Category-Disjointness condition does not apply to distance scales beyond R, and even for $R = \infty$, the guarantee of Equation (4.7) is quite weak at very large scales. Accordingly, we find that the Two-Ball Algorithm becomes problematic at large distance scales. To deal with these issues, we apply the Two-Ball Algorithm only to distance scales small enough to provide strong guarantees. The improved distance estimates define edge lengths, and a post-processing step computes shortest paths with respect to these edge lengths. The resulting algorithm, called *Extended Two-Ball Algorithm,* satisfies the following theorem.

THEOREM 4.3. *Assume the setting of Theorem 3.1 with Scale-$R^{1+1/(d+1)}$ Category-Disjointness, where $R \geq \text{polylog}(n)$ for a sufficiently large $\text{polylog}(n)$. Then, the Extended Two-Ball Algorithm runs in polynomial time, and with high probability produces distance estimates $\mathcal{N}_i'$ with the following guarantee:*

For any pair (s, t) at normalized distance $x = \mathcal{N}_i(s, t)$, the estimate $\mathcal{N}_i'(s, t)$ has multiplicative distortion $1 \pm \Big[(\min(x, R, \hat{R}))^{-d/(2d+2)} \cdot O(\log^2 n)\Big]$, where $\hat{R} = \left(\frac{n}{\log n}\right)^{(2d+2)/(2d^2+3d)}$.

Remark. The distortion in Theorem 4.3 can be interpreted as $1 \pm O\left(\ell^{-d/(2d+2)} \cdot \log^2 n\right)$, where $\ell = \min(x, R, \hat{R})$ is the "effective distance scale".

We begin by defining the Extended Two-Ball Algorithm precisely. The input consists of the multi-category social graph and the distance estimates $\mathcal{N}^* = \mathcal{N}_i^*$ for a given category i, as guaranteed by Theorem 3.1. Recall that these are non-contracting estimates with constant expansion δ and $\text{polylog}(n)$ additive error; we assume that (an upper bound on) δ is known to the algorithm. Apart from δ, the algorithm is parameterized by the distance scale R from Theorem 4.3.

The algorithm proceeds as follows. It focuses on the edge set $H = \{(u, v) \mid \mathcal{N}^*(u, v) \leq R\}$. For each edge $(u, v) \in H$, it applies the Two-Ball Algorithm with respect to distances $\mathcal{N}^*$ to obtain improved distance estimates $\mathcal{N}_H(u, v)$. These improved estimates are treated as edge lengths for H. For each node pair (s, t), we distinguish two cases. If the edge (s, t) is in H, we simply set the final estimate $\mathcal{N}_i'(s, t) = \mathcal{N}_H(s, t)$. Otherwise, the final distance estimate $\mathcal{N}_i'(s, t)$ is the length of the shortest s-t path using the edge set

$$H_t = \{(u, v) \in H \mid \mathcal{N}^*(u, v) \geq \tfrac{R}{2\delta} \text{ or } v = t\}. \tag{4.8}$$

In other words, the distance is estimated by the length of the shortest path using only "sufficiently long" edges, except for possibly the last edge, which may be short.

The Recursive Two-Ball Algorithm. We show that the Recursive Two-Ball Algorithm from Section 4.1 can be applied verbatim in the case of multiple categories with Scale-∞ Category-Disjointness, yielding poly-logarithmic additive error. The analysis only needs to be modified slightly to deal with edges from other categories. However, our guarantees only apply to node pairs at distances $x \leq n^{1/(d+1)} = D^{d/(d+1)}$, where $D = n^{1/d}$ is the diameter of the metric space.

THEOREM 4.4. *Consider a multi-category social graph with $C_{sg}k_{sg} = \Omega(\log n)$, with Scale-$\infty$ Category-Disjointness and perfectly uniform density for each category. Assume that the social distance in each category*

is defined by the ℓ_2^d norm, with $d > 2$. Then, the Recursive Two-Ball Algorithm runs in polynomial time, and produces distance estimates $\mathcal{N}_i'$ satisfying the following guarantee with high probability:

> *For every pair (s,t) of nodes at normalized distance $\mathcal{N}_i(s,t) \leq n^{1/(d+1)}$, we have that*
>
> $$|\mathcal{N}_i'(s,t) - \mathcal{N}_i(s,t)| \leq \text{polylog}(n).$$

For normalized distances larger than $n^{1/(d+1)}$, even under actual randomly permuted categories, the number of edges from other categories grows prohibitively large for large distances; it seems unlikely that this obstacle could be easily overcome.

However, we can use the improved estimates from Theorem 4.4 with the post-processing step from the Extended Two-Ball Algorithm (with $R = n^{1/(d+1)}$). The resulting algorithm estimates normalized distances $x > R$ with additive error (x/R) polylog(n).

4.3 Constant target degree

The analysis so far has relied heavily on the fact that the target degree k_{sg} (essentially the expected average node degree) was at least logarithmic. Indeed, as discussed in Section 2, the first obvious problem with constant expected degree is that with non-negligible probability, the social graph E_{sg} is disconnected. To circumvent this problem, much of the past literature (e.g., [18, 27, 28, 40]) assumes that in addition to the random edges, the network also contains a set E_{loc} of *local edges* deterministically.[9] In the literature, E_{loc} is frequently the d-dimensional grid. We adopt a more general model in which E_{loc} can be essentially any set of short edges. A constant target degree poses two additional challenges beyond mere connectivity:

- There are insufficiently many long-range links to support pruning via counting common neighbors. Even for short distances, the number of common neighbors is only constant, and high-probability guarantees can therefore not be obtained.[10] Therefore, in order to identify short edges as such, we need to rely on the structure of E_{loc}.
- To avoid stochastic dependence between multiple stages (such as the Two-Hop Test and Two-Ball Algorithm), we had previously partitioned E_{sg} randomly into separate sets to be used in the stages. With constant node degrees, this may risk leaving the Two-Hop Test with only half of the local edges E_{loc}. Hence, partitioning the edges may not be viable any more. On the other hand, if the same edges are used in multiple stages, subtle stochastic dependencies between the stages are created; our analysis needs to carefully account for these dependencies.

Here we explore the changes (in modeling, algorithms and analysis) necessary to deal with constant target degrees. We focus on the single-category case for the remainder of the subsection.

Our results apply so long as the set of local edges is "rich enough" in local connectivity.

DEFINITION 4.2. ("RICHNESS" OF LOCAL EDGES)
1. *An edge set E is a* (σ,δ)-spanner *if its shortest-path distance $\mathcal{D}^{\text{sp}}$ satisfies, for all node pairs (u,v):*

$$\sigma \cdot \mathcal{D}(u,v) \;\leq\; \mathcal{D}^{\text{sp}}(u,v) \;\leq\; \delta \cdot \mathcal{D}(u,v)$$

2. *A set E of edges is* (b,h)-connected *if for every edge $(u,v) \in E$, E contains b edge-disjoint u-v paths of at most h edges each.*

3. *E_{loc} is* (b,h)-rich *with distortion (σ,δ) if it is a (σ,δ)-spanner and contains a (b,h)-connected (σ,δ)-spanner $E \subseteq E_{loc}$ (called its* connectivity witness*).*

As an example, we show in the full version [1] that the d-dimensional toroidal grid is $(2d-1,3)$-rich and (for $d \geq 2$) $(2d,7)$-rich, both with distortion $(1,O(1))$.

Next we present a solution which relies on knowing parameters (b,h) of the local structure's richness. In other words, the pruning algorithm needs to know how rich a local structure to expect. Later, we show how to make the pruning algorithm adapt to the available richness under fairly mild assumptions.

Basic Approach: Edge-Disjoint Paths. Our solution is based on a more careful design of the pruning stage, where instead of counting common neighbors, the algorithm counts edge-disjoint paths of bounded length. The pruning stage is very simple: The algorithm starts with an edge set $E = E_{\text{sg}}$. It prunes each edge $(u,v) \in E$ such that E does not contain b edge-disjoint u-v paths of at most h hops each. This is repeated until no more edges can be pruned. We call this algorithm the (b,h)-*EDP Pruning Algorithm;* here, *EDP* stands for Edge-Disjoint Paths.

The idea is that this algorithm keeps a sufficiently rich subset of local edges, and prunes all edges in E_{sg}

[9] Without loss of generality, E_{loc} can also include all edges which would be included by the basic small-world model with probability 1.

[10] See, e.g., the difficulties faced by [20]. The authors of [20] consider a small-world model with one random neighbor for each node. They can only make guarantees about pruning away all but a poly-logarithmic number of long-range edges. The main reason is that even distant nodes will choose the same random neighbor with probability $\Omega(1/n)$, and high-probability bounds therefore only guarantee at most poly-logarithmically many long random edges to remain.

whose length exceeds some threshold r_{EDP} (defined in Equation (4.9)). (We call such edges *long edges*.) For edges of intermediate length, the algorithm makes no guarantees about whether they are pruned. Crucially, the pruned graph does not depend on the long edges, in the following sense: Let $E_{\text{sg}}, \hat{E}_{\text{sg}}$ be two edge sets generated according to the same distribution, such that the random choices for non-long edges are the same, and the random choices for long edges are independent. Then, with high probability (over the random process generating all edges of E_{sg} and $\hat{E}_{\text{sg}}$), the remaining set of edges after pruning is the same for both E_{sg} and $\hat{E}_{\text{sg}}$. The advantage of this guarantee is that we do not need to worry about dependencies on the pruned graph, so long as the post-processing stage only uses long edges. Therefore, we can use the pruned graph to define the initial estimates $\mathcal{N}^*$ for normalized distances and then use a suitably modified and optimized version of the (Recursive) Two-Ball Algorithm which only considers node pairs (s,t) for which $\mathcal{N}^*(s,t)$ is sufficiently large.

We start the analysis of the (b,h)-EDP Pruning Algorithm with several observations. First, notice that the pruned graph $T(E)$ is the maximal (b,h)-connected subset of E, i.e., the union of all such subsets. It follows that $T(E)$ does not depend on the order in which the edges are pruned. Second, because $T(E)$ is the maximal (b,h)-connected subset of E, the pruned graph $T(E)$ does not depend on the presence or absence of the pruned edges $e \in E \setminus T(E)$. Formally, $T(E) = T(E')$ whenever $T(E) \subseteq E' \subseteq E$.

To ensure correctness, we can use the (b,h)-EDP Pruning Algorithm only if the local structure is (b,h)-rich. The performance depends on the parameters (b,h): we get better estimates for larger b and smaller h. We summarize our results as follows. In a slight abuse of notation, here, the (Recursive) Two-Ball Algorithm refers to the suitably modified version that works with the (b,h)-EDP Pruning Algorithm.

THEOREM 4.5. *Consider a single-category social graph of near-uniform density. Suppose that the local edge set E_{loc} is (b,h)-rich with distortion (σ,δ). Let $D = \Theta(n^{1/d})$ be the diameter of the metric space. For any constant $\alpha > 0$ (not known to the algorithm), let*

$$r_{\text{EDP}}(\alpha) = D^{(2+\alpha)/b} \cdot h \cdot (O(k_{sg} + \log^{1+\alpha} n))^{2h/d} = D^{(2+\alpha)/b} \cdot (O(\log n))^{O(h)}. \tag{4.9}$$

Let E' be the edge set retained by the (b,h)-EDP Pruning Algorithm. Then, with probability at least $1 - O(n^{-\alpha})$, the following hold.

(a) *E' contains the connectivity witness E'_{loc} of E_{loc} and no edges whose length exceeds $r_{\text{EDP}}(\alpha)$. The algorithm makes no guarantees for other edges.*

(b) *Let $\mathcal{D}^{\text{sp}}$ be the shortest-path distance on E'. Then, for all node pairs (u,v), we have that*

$$\mathcal{D}(u,v) \le \beta\, \mathcal{D}^{\text{sp}}(u,v) \le \delta \cdot \beta\, \mathcal{D}(u,v), \quad \text{where } \beta = \max(\tfrac{1}{\sigma}, r_{\text{EDP}}(\alpha)).$$

In words, the shortest paths distance in E', scaled up by β, gives no contraction, and expansion at most $\delta\,\beta$.

(c) *The Two-Ball Algorithm reconstructs all normalized distances $\mathcal{N}(u,v)$ with unit distortion and additive error $r_{\text{EDP}}(\alpha)(\mathcal{N}^{\gamma}(u,v) + r_{\text{EDP}}(\alpha))$, where $\gamma = \frac{d+2}{2d+2}$.*

(d) *Assume that the metric has perfectly uniform density, and the social distance is the ℓ_2^d norm for $d \ge 3$ dimensions. Then the Recursive Two-Ball Algorithm reconstructs all normalized distances with unit distortion and additive error $r_{\text{EDP}}(\alpha) \cdot \text{polylog}(n)$.*

Running times in Theorem 4.5. While the main thrust in this paper is information-theoretic, the algorithms in Theorem 4.5 are actually polynomial. Let us discuss how to improve the running times to near-linear, an important feature for the sizes of networks we are envisioning.

The naïve implementation of the (b,h)-EDP Pruning Algorithm checks every remaining edge at each iteration, which gives a running time of $\tilde{O}(n^2)$. However, we show the following:

LEMMA 4.1. *The (b,h)-EDP Pruning Algorithm can be implemented in $\tilde{O}(n)$ time for constant b and h.*

We also comment on the running time of the Two-Ball Algorithm. Applying this algorithm to a given node pair (u,v) can be computationally expensive when $\mathcal{D}(u,v)$ is large (and consequently, the algorithm needs to consider large balls around u and v). Thus, the Two-Ball Algorithm for a given node pair can be viewed as a precise but costly distance measurement. Instead of applying it to *every* node pair, we could instead use the beacon-based triangulation technique from [30]: here, one selects $O((\frac{1}{\epsilon})(\frac{1}{\delta})^d)$ "beacon nodes" uniformly at random, and measures the distance from each node only to each beacon. This technique achieves distortion $(1+\delta)C$ for all but an ϵ-fraction of node pairs, where C is the distortion of the Two-Ball test.

Adapting to the "optimal" richness. Theorem 4.5 assumes that the (b,h)-richness of the local edge set E_{loc} is known to the algorithm. In reality, it is desirable to adapt to the "optimal" richness without knowing it in advance. Here, the "optimal" richness means the (b,h) pair that minimizes $r_{\text{EDP}}(\alpha)$ in Equation (4.9), subject to the constraint that E_{loc} is (b,h)-rich with small distortion.

Our algorithm, called *Adaptive EDP algorithm,* proceeds as follows: for a given set H of candidate hop counts, we try all (b, h) pairs, $h \in H$, in order of increasing $r_{\text{EDP}}(\alpha)$ until the pruned graph is connected, and focus on the last pair. Without loss of generality, we can start with b equal to the smallest node degree in E_{sg}. We can use binary search over the (b, h) pairs (in the same order) to reduce the number of pairs that we need to consider.

While the above algorithm is very simple, the challenge is to argue when and if it works. Let $T_{b,h}(E)$ denote the pruned graph if the (b, h)-EDP Pruning Algorithm is applied to the edge set E. We rely on the following crucial observation:

LEMMA 4.2. *Consider a single-category social graph with near-uniform density. Suppose that the local structure E_{loc} is a $(\cdot, \delta)$-spanner, and moreover, $T_{b,h}(E_{loc})$ contains at least ϵn isolated nodes, for some parameters b, h, ϵ, δ such that*

$$(2\delta h)^d \, C_{UD}^2 \, C_{sg} \, k_{sg} \leq \tfrac{1}{6}. \tag{4.10}$$

Then $T_{b,h}(E_{sg})$ is disconnected with high probability.

Since $C_{\text{sg}} = \Theta(1/\log n)$ and $C_{\text{UD}} = \Theta(1)$, condition (4.10) holds, for large enough n, whenever k_{sg}, δ and h are constants. The lemma motivates the following definition of "robustness" of a graph.

DEFINITION 4.3. *A connected graph $G = (V, E)$ is called* (ϵ, h)-robust *with distortion (σ, δ), for some $\epsilon \in (0, 1]$, if the following holds for every b: either G is (b, h)-rich with distortion (σ, δ), or $T_{b,h}(E)$ contains at least ϵn isolated nodes.*[11]

In the first case of this definition, we can use the (b, h)-EDP Pruning Algorithm safely, while in the second case, we will show that $T_{b,h}(E_{\text{sg}})$ is disconnected with high probability. Notice that the toroidal grid is $(1, h)$-robust for any h. We give more examples of robust graphs in the full version [1].

THEOREM 4.6. *Consider a single-category social graph with near-uniform density and local structure E_{loc}. Suppose that for all $h \in H$, E_{loc} is (ϵ, h)-robust with distortion (σ, δ) and (4.10) holds. Then, when the Adaptive EDP algorithm is run with the candidate set H, it will obtain the guarantees of Theorem 4.5 for the optimum pair (b, h) among all $h \in H$.*

[11] Any graph G in Definition 4.3 is a (σ, δ)-spanner. This is because for $b = 1$, no edges are pruned, and so G must be $(1, h)$-rich with distortion (σ, δ), so it is a (σ, δ)-spanner.

5 Conclusions

We have shown that, under standard assumptions about generative models for social networks, it is possible to reconstruct social spaces with small distortion from a multiplex social network; indeed, it is possible to do so in near-linear time. The edges do not need to be labeled with their "origin," so long as the categories are "locally sufficiently uncorrelated." Under increasingly stronger assumptions, the distortion can be improved from constant, to $1 + o(1)$, to poly-logarithmic additive error. While these results rely on having poly-logarithmic node degree, we also show that small polynomial distortion can be obtained in the constant-degree regime, so long as the social network contains a sufficiently rich local structure. This is possible even if the algorithm only possesses very rudimentary knowledge about the local structure.

While our results can be interpreted as a proof of concept — it is possible in principle to efficiently separate the different dimensions of social interactions and identify similarities between individuals — they set the stage for a number of possible extensions.

1. There are several specific technical open questions within our model, the most immediate of which is extending the multi-category results to the constant-degree regime.

2. We assumed that the algorithm had knowledge of various input parameters (the number of categories, the number of dimensions, etc.), whereas ideally, the algorithm should be able to learn these parameters from input data as well.

3. For our multi-category algorithms to work, we required a "category disjointness" condition, essentially stating that locally, metrics look uncorrelated with respect to each other. It seems unlikely that one could reconstruct metrics if categories were extremely similar, but it is an interesting open question how much our current condition could be weakened while still allowing for provable reconstruction. In particular, we conjecture that future work will be able to deal with a few localized violations of the category disjointness condition, so that they lead to incorrect distance estimates only for the affected nodes, without propagating to other parts of the metric space.

4. Our model so far also assumes that the node degrees are essentially uniform across nodes, which will usually not hold in practice. A corresponding extension for the single-category case might not be too difficult, but inferring the individual node degrees for multiple categories appears more difficult.

5. Finally, and perhaps most importantly, one may want to consider "host spaces" other than Euclidean spaces with near uniform density, such as ultrametrics, more general "group structures" (e.g., [28]), or point sets with significantly non-uniform density. It would be particularly interesting if an algorithm did not need to know the structure of the host space in advance, and instead could infer it from the data.

In practice, there will usually be additional information available beyond the edges. This may include information about nodes' locations, interests, or demographics (as collected by social networking sites); partial interaction statistics along the edges; or perhaps a social network that has been previously embedded in a social distance space, but is now being extended by a few new nodes. In either case, it is an interesting question how to formalize the benefits that can be obtained with such side information. In particular, time stamps on edges introduce a temporal dimension into the problem: now, instead of fixed node locations in the social space, one could ask about nodes' trajectories over time.

Acknowledgments We would like to thank Christian Borgs, Jennifer Chayes, Moises Goldszmidt, Bobby Kleinberg, Jon Kleinberg, Peter Monge, Satish Rao and Ken Wilbur for useful discussions and pointers.

References

[1] Ittai Abraham, Shiri Chechik, David Kempe, and Aleksandrs Slivkins. Low-distortion inference of latent simlarities from a multiplex social network. Eprint arXiv:1202.0922, 2012.

[2] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2003.

[3] Lada A. Adamic and Eytan Adar. How to search a social network. *Social Networks*, 27(3):187–203, 2005.

[4] Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: Toward a rigorous approach. In *Proc. 14th ACM Conf. on Electronic Commerce*, pages 37–54, 2012. arXiv: 1112.1831.

[5] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *19th Intl. World Wide Web Conference*, pages 61–70, 2010.

[6] Maria-Florina Balcan, Christian Borgs, Mark Braverman, Jennifer Chayes, and Shang-Hua Teng. Finding endogenously formed communities. In *Proc. 24th ACM-SIAM Symp. on Discrete Algorithms*, 2013.

[7] Marc Barthélemy. Spatial Networks. *Physics Reports*, 499:1–101, 2011.

[8] Peter M. Blau. *Inequality and Heterogeneity: A Primitive Theory of Social Structure*. Free Press, 1977.

[9] Ulrik Brandes and Thomas Erlebach, editors. *Network Analysis*. Springer, 2005.

[10] Carter T. Butts. Predictability of large-scale spatially embedded networks. In *Dynamic Social Network Modeling and Analysis: Workshop summary and papers*, pages 313–323, 2003.

[11] Hubert T-H. Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. In *16th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 762–771, 2005. Full and updated version available as a Carnegie Mellon University ETR CMU-PDL-04-106.

[12] Aaron Clauset, Cris Moore, and Mark Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[13] Leon Danon, Jordi Duch, Albert Diaz-Guilera, and Alex Arenas. Comparing community structure identification. *J. Stat. Mech.*, 2005.

[14] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[15] Stephen E. Fienberg, Michael M. Meyer, and Stanley S. Wasserman. Statistical analysis of multiple sociometric relations. *J. American Statistical Association*, 80(389):51–67, 1985.

[16] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010. Eprint arXiv: 0906.0612.

[17] Santo Fortunato and Claudio Castellano. Community structure in graphs. In R. Meyers, editor, *Encyclopedia of Complexity and System Science*. Springer, 2009. Eprint arXiv:0712.2716.

[18] Pierre Fraigniaud. Small worlds as navigable augmented networks: Model, analysis, and validation. In *Proc. 15th European Symp. on Algorithms*, pages 2–11, 2007.

[19] Pierre Fraigniaud, Emmanuelle Lebhar, and Zvi Lotker. A doubling dimension threshold $\Theta(\log \log n)$ for augmented graph navigability. In *Proc. 14th European Symp. on Algorithms*, pages 376–386, 2006.

[20] Pierre Fraigniaud, Emmanuelle Lebhar, and Zvi Lotker. Recovering the long-range links in augmented graphs. *Theoretical Computer Science*, 411(14–15):1613–1625, 2010.

[21] Sharad Goel and Daniel G. Goldstein. Predicting behavior with social networks. Unpublished Manuscript, 2010.

[22] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *44thIEEE Symp. on Foundations of Computer Science*, pages 534–543, 2003.

[23] Mark S. Handcock, Adrian E. Raftery, and Jeremy M. Tantrum. Model-based clustering for social networks (with discussion). *Journal of the Royal Statistical Society, Series A*, 170:301–354, 2007.

[24] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*,

97:1090–1098, 2002.

[25] Anne-Marie Kermarrec, Vincent Leroy, and Gilles Trédan. Distributed social graph embedding. Technical Report RR-7327, INRIA, 2010.

[26] Jon Kleinberg. Navigation in a small world. *Nature*, 406:485, 2000.

[27] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. 32nd ACM Symp. on Theory of Computing*, pages 163–170, 2000.

[28] Jon Kleinberg. Small-world phenomena and the dynamics of information. In *Proc. 13th Advances in Neural Information Processing Systems*, pages 431–438, 2001.

[29] Jon Kleinberg. Complex networks and decentralized search algorithms. In *Proc. Intl. Congress of Mathematicians (ICM)*, 2006.

[30] Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and Embedding Using Small Sets of Beacons. *Journal of the ACM*, 56(6), September 2009. Preliminary version in Proc. 45th IEEE Symp. on Foundations of Computer Science.

[31] Pavel N. Krivitsky, Mark S. Handcock, Adrian E. Raftery, and Peter D. Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31:204–213, 2009.

[32] Paul Lazarsfeld and Robert K. Merton. Friendship as a social process: A substantive and methodological analysis. In Morroe Berger, Theodore Abel, and Charles H. Page, editors, *Freedom and Control in Modern Society*, pages 18–66. Van Nostrand, 1954.

[33] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[34] David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic routing in social networks. *Proc. Natl. Acad. Sci. USA*, 102:11623–11628, 2005.

[35] Kun Liu and Lei Tang. Large scale behavioral targeting with a social twist. In *Proc. 20th ACM Conf. on Information and Knowledge Management (CIKM)*, 2011.

[36] Peter Marsden and Noah E. Friedkin. Network studies of social influnce. *Sociological Measures and Research*, 22(1):127–151, 1993.

[37] David D. McFarland and Daniel J. Brown. Social distance as a metric: A systematic introduction to smallest space analysis. In Edward O. Laumann, editor, *Bonds of Pluralism: The Form and Substance of Urban Social Networks*, pages 213–253. Wiley, 1973.

[38] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[39] Michael J. Minor. New directions in multiplexity analysis. In Ronald S. Burt and Michael J. Minor, editors, *Applied Network Analysis*, pages 223–244. Sage Publications, 1983.

[40] Van Nguyen and Charles U. Martel. Analyzing and characterizing small-world graphs. In *Proc. 16th ACM-SIAM Symp. on Discrete Algorithms*, pages 311–320, 2005.

[41] Adrian E. Raftery, Xiaoyue Niu, Peter D. Hoff, and Ka Yee Yeung. Fast inference for the latent space network model using a case-control approximate likelihood. Technical Report 572, Department of Statistics, University of Washington, 2010.

[42] Everett Rogers. *Diffusion of innovations*. Free Press, 4th edition, 1995.

[43] Purnamrita Sarkar, Deepayan Chakrabarti, and Andrew W. Moore. Theoretical justification of popular link prediction heuristics. In *Proc. 23rd Conference on Learning Theory*, pages 295–307, 2010.

[44] Purnamrita Sarkar and Anrew W. Moore. Dynamic social network analysis using latent space models. In *Proc. 17th Advances in Neural Information Processing Systems*, 2005.

[45] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[46] Michael F. Schwartz and David C. M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, 1993.

[47] Michael Schweinberger and Tom A. B. Snijders. Settings in social networks: A measurement model. *Sociological Methodology*, 33:307–341, 2003.

[48] Aleksandrs Slivkins. Distance estimation and object location via rings of neighbors. *Distributed Computing*, 19(4):313–333, March 2007. Special issue for *24th ACM PODC*, 2005.

[49] Aleksandrs Slivkins. Towards Fast Decentralized Construction of Locality-Aware Overlay Networks. In *26thACM Symp. on Principles of Distributed Computing*, pages 89–98, 2007.

[50] Tom A. B. Snijders. Statistical models for social networks. *Annual Review of Sociology*, 37:129–151, 2011.

[51] Micheal Szell, Renaud Lambiotte, and Stefan Thurner. Multi-relational organization of large-scale social networks in an online world. *Proc. Natl. Acad. Sci. USA*, 107:13636–13641, 2010.

[52] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proc. 15th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 817–826, 2009.

[53] Duncan J. Watts, Peter S. Dodds, and Mark E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

[54] Barry Wellman. Which types of ties and networks give what kinds of social support? *Advances in Group Processes*, 9:207–235, 1992.

[55] Barry Wellman and Scot Wortley. Different strokes from different folks: Community ties and social support. *American Journal of Sociology*, 96:558–588, 1990.