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

Ittai Abraham [*] Shiri Chechik [†] David Kempe [‡] Aleksandrs Slivkins [*]

February 2012
This revision: April 2012

Abstract

What can a social network tell us about the underlying latent "social structure," the way in which individuals are similar or dissimilar? 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. Having explicit access to similarity information instead of merely the noisy signal provided by the presence or absence of edges could improve analysis significantly. We study the following natural question: Given a social network — reflecting the underlying social distances between its nodes — how accurately can we reconstruct the social structure?

It is tempting to model similarities and dissimilarities as distances, so that the social structure is a metric space. However, observed social networks are usually multiplex, in the sense that different edges originate from similarity in one or more among a number of different categories, such as geographical proximity, professional interactions, kinship, or hobbies. Since close proximity in even one category makes the presence of edges much more likely, an observed social network is more accurately modeled as a *union* of separate networks. In general, it is a priori not known which network a given edge comes from. While generative models based on a single metric space have been analyzed previously, a union of several networks individually generated from metrics is structurally very different from (and more complex than) networks generated from just one metric.

In this paper, we begin to address this reconstruction problem formally. The latent "social structure" consists of several metric spaces. Each metric space gives rise to a "distance-based random graph," in which edges are created according to a distribution that depends on the underlying metric space and makes long-range edges less likely than short ones. For a concrete model, we consider Kleinberg's small-world model and some variations thereof. The observed social network is the union of these graphs. All edges are unlabeled, in the sense that the existence of an edge does not reveal which random graph it comes from. Our main result is a near-linear time algorithm which reconstructs from this unlabeled union each of the individual metrics with provably low distortion.

[*] Microsoft Research Silicon Valley, Mountain View CA, USA. Email: {ittaia,slivkins}@microsoft.com.

[†] Dept. of Computer Science, Weizmann Institute of Science, Rehovot, Israel. Email: shiri.chechik@weizmann.ac.il. Research done in part while at Microsoft Research, Silicon Valley.

[‡] 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.

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* [19, 58, 73]: 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.

1.1 An overview of the model

We posit a latent space model (described in detail in Section 3) for the generation of social networks akin to models widely used in the mathematical sociology, statistics, and computer science communities [14, 29, 33, 34, 38, 40, 45, 62, 64, 65, 68] (see also the survey [71, pages 15–21]).

The model is based on two widely accepted tenets about social networks (e.g., [9, 56]). 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.

[1] The model is agnostic about whether this similarity is caused more by *homophily* [47, 57] (the tendency to form ties with those who are similar) or by *social influence* [55, 63] (the tendency to *become* similar to one's associates).

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 [33, 34, 40, 45, 62, 64], and that the density of the points corresponding to individuals is nearly uniform [34, 40, 64]. 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 (1) in Section 3).

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 [40], 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., [11]), 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.

1.2 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 [42] 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 [1, 51, 64, 67]): edges (u, v) are more likely to be between friends in a category if they are "supported" by many common neighbors

[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.

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 4, 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, 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 5 and 6) 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 long-range 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 7), 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.

For both extensions, more detailed descriptions of challenges, results, and high-level approaches are deferred to the introductory portions of the corresponding sections.

Our algorithms are modular: a pre-processing step (counting common neighbors, or the low-degree algorithm of Section 7) 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.

[3] Sarkar et al. [64] showed that under a model similar to 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.

1.3 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 [40, 39, 41, 42, 22], 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 [8].

   Much of the past work in the statistics community [33, 34, 45, 62, 64] 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 [2, 5, 52] with Kleinberg's inverse polynomial distribution [40, 41].[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] [29, 33, 34, 38, 40, 45, 62, 64], 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 will achieve similar results for more general metric spaces or related structures, in particular, ultrametrics [14, 41, 68], 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 [76], 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 [33, pp. 337, 348], [68, p. 335]. A generalization to more realistic models is a natural direction for future work.

[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 [28].

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 [34, 64]; such a model of individual categories would imply our “small intersection” condition with high probability. In fact, we show in Section 8 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.

1.4 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 [31, 53] 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 [75] 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 [78, 79].

Another natural application is the discovery of “social communities” [10, 20, 21, 16, 66]. 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 [33, 45] 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 [1, 14, 51, 64, 67]. Unobserved or potential links are most likely present between node pairs at small distances; hence, once distances are known, missing links can be predicted easily [14, 64].

2 Related work

Our work is related to work in a large number of communities: latent space reconstruction in statistics and mathematical sociology, community discovery, small-world networks, network localization, and metric space embeddings. We discuss the different areas in their separate sections.

2.1 Latent Space Reconstruction

Several recent papers [5, 14, 29, 33, 34, 38, 45, 62, 64, 65, 68] aim to reconstruct latent metrics from an observed social network. The precise models differ across these papers: most assume Euclidean spaces [5, 29, 33, 34, 38, 45, 62, 64, 65], while a few consider ultrametrics to model hierarchical communities [14, 68]. Among the papers considering Euclidean spaces, there are different assumptions about link distributions: most assume a logit-linear model [33, 34, 45, 62, 64], while a few consider inverse polynomial "small-world" distributions [5, 29, 38].[6] There are many other modeling dimensions along which these papers (and ours) differ, including: variance in node degrees, additional information about nodes (such as locations of some nodes [5]), uniform or clustered priors for node locations, whether algorithms are supposed to be centralized or distributed [38], etc.[7]

Two main differences stand out between our work and the majority of these papers (in addition to the more minor modeling differences). First, we model multiple categories, which is extremely realistic, but makes the model, algorithms, and analysis significantly more complex. Second, the majority of the work cited above [5, 14, 33, 34, 38, 45, 62, 68] estimates the underlying space either using Maximum Likelihood Estimates (MLE), or by imposing a Bayesian Prior and maximizing the probability of the chosen locations. Both appear to be very complex problems, and indeed, all of the papers employ heuristics (based on Gibbs Sampling, Metropolis-Hastings, Simulated Annealing, etc.) without guarantees on the likelihood or probability of the solution returned. More fundamentally, even if it were possible to obtain the MLE or highest-probability solution, it is not clear that it would come with any guarantees on the worst-case (or even average) distortion; the

[6] We remark that several recent studies [2, 5, 52] show that the frequency of friendships as a function of (2-dimensional) geographic distance, when corrected for non-uniform densities, appears to decrease as $\Theta(r^{-2})$. This gives some tentative empirical evidence in favor of "small-world" distributions.

[7] Much of the recent work in the mathematical sociology community has focused on exponential random graph models, which in a sense "hard-wire" desired distributions of certain features. These models are generally of a very different nature from latent-space models. A recent paper by Butts [12] combines features of both location-based and exponential random graph models; like the other papers listed above, it is not clear whether inference of model parameters would be tractable, and whether it would lead to any guarantees on distortion.

objective function does not explicitly model distortion, and in particular may be sacrificing the distortion of some edges in order to optimize the more global objective.

Two notable exceptions to the MLE/Bayesian approach are the works of Fraigniaud, Lebhar, and Lotker [29] and Sarkar, Chakrabarti, and Moore [64]. Fraigniaud et al. [29] aim to reconstruct a single-category small-world model in order to use the distance estimates for greedy routing. They propose a heuristic based on an MLE intuition; interestingly, this heuristic leads to essentially counting common neighbors. Their algorithm may retain a small number of long-range edges, and hence does not come with provable guarantees on the distortion of the reconstructed metric space. They prove that this does not stand in the way of greedy routing: despite the lack of distortion guarantees, the distances they construct provably enable greedy routing along poly-logarithmic length paths.

Sarkar et al. [64] begin from the goal of explaining why simple heuristics for link prediction, such as counting common neighbors, are successful. They show that such heuristics can be understood as identifying close pairs of nodes in a latent Euclidean space, and use this insight to give provable guarantees on the performance of several heuristics for link prediction. (They also suggest additional heuristics). In the process, they show how a metric space is implicitly reconstructed by counting common neighbors. There are a few key differences between their work and ours. First, their distributions are logit-linear, implying that long-range edges are extremely unlikely. The reconstruction task is still non-trivial, but they do not have to deal with any very long-range edges, of which our model will have many. Second, they only consider a single category; for us, the single-category pruning step is a departure point for the more complex stages of separating the different categories, and using long-range links to improve the distortion.

2.2 Overlapping Communities

There are conceptual similarities between our work and concurrent and independent work by Arora et al. [3] and Balcan et al. [6]. Their goal is more specifically to reconstruct overlapping community structure in graphs; similar to our approach, they also posit that the social network is a noisy signal about some true underlying social structure, and communities are defined with respect to those structures. Recall that the goal of properly identifying communities is also one of the motivations for our work, although we do not explicitly pursue the question of reconstructing communities with provable guarantees.

The major difference between our work and that of [3, 6] is that both Arora et al. and Balcan et al. assume a set-based latent structure (each community is modeled as a set), whereas we assume a latent structure based on a near-uniform-density metric (each social category is modeled as a separate metric space). This difference, in turn, leads to different random graph models and algorithmic ideas. In principle, the set-based structures could be modeled as 0-1 metrics (and thus fit into our framework); however, such metrics would dramatically violate our uniform density assumption, so that our algorithms are not applicable.

Nonetheless, some conceptual similarities between our work and [3, 6] are worth noting. First, a crucial aspect of all three papers is the ability to deal with overlapping latent structures: multiple social categories in the present paper, and multiple communities for [3, 6]. All three papers need some notion of "gap assumption" that limit overlaps in order to handle such structures. Second, a high-level idea present in all three papers is to start with a "seed" and then "grow" it to find the respective latent structures. While the high-level algorithmic ideas are similar, the details differ significantly between our Amoeba algorithm and the algorithms in [3, 6]. The Amoeba algorithm grows the "Amoeba" gradually, and using short disjoint paths, whereas [3, 6] use ideas related to finding hidden cliques. In addition, the goal of reconstructing metrics motivates substantial algorithmic extensions (discussed in Sections 5–7) related to improving the distortion and dealing with small node degrees. These algorithmic questions have no direct analogue in the setting of reconstructing communities.

2.3 Network Localization, Embeddings, and Distance Oracles

Reconstructing (low-dimensional, Euclidean) node distances from distance measurements has been studied both theoretically and practically from a wide variety of angles. In *network localization* for mobile and sensor networks (e.g., [4, 72, 81]), and *network embedding* for peer-to-peer networks and the Internet (e.g., [60, 15, 80, 43]), distances are known fairly accurately, but typically only to a few "beacon" nodes. The challenge is to choose beacons, and combine measurements, to estimate pairwise distance. In our setting, the presence or absence of edges provides much less reliable estimates of distances. However, once we succeed in obtaining basic distance measurements, the techniques from network embedding/localization can lead to further improvements in the estimates without a blowup in the running time, as shown in Section 7.

We measure the quality of our inferred metrics in terms of the distortion of the estimates. Distortion is commonly used as a measure of quality in metric embeddings and distance oracles (see, respectively, [35] and [82] for surveys). In those domains, distances are known precisely, and the challenge is typically to find a compact and faithful representation, for instance in terms of low dimensionality of the target metric or small space of the oracle. In our setting, the true distances (in each category) are not explicitly known, and the estimates are very noisy. Similar to metric space embeddings, our goal is to extract a faithful representation of each category. However, a second fundamental difference is that the space we "embed" in consists of multiple metrics, and thus severely violates the triangle inequality.

Our focus on near-uniform density metrics is motivated by similar notions of low dimensionality in metric embedding, nearest neighbor search, and a number of other problems, e.g. [36, 32, 44, 74, 43]. In particular, near-uniform density has been used along with various modeling assumptions in [37, 43].

2.4 Small-World Networks

A long line of empirical studies confirms that many social ties and interactions correlate strongly with social distance, and particularly geographical distance (see, e.g., [56, 59] for a discussion). For example, Butts [11] gives calculations showing that geographical information alone could reduce the entropy in network prediction by roughly 90% under moderate assumptions. More specifically, several recent studies [2, 5, 52] show that the frequency of friendships as a function of (2-dimensional) geographic distance, when corrected for non-uniform densities, appears to decrease as $\Theta(r^{-2})$.

Small-world models aim to capture the natural tradeoff between a preference for shorter links and the randomness observed in the presence of long-range links. Initial models were due to Watts and Strogatz [77] and Kleinberg [40, 41]. One of the main goals in these papers was to explain why greedy routing — based only on the position of one's neighbors in the metric space — can discover paths of polylogarithmic length. Since the publication of [40, 41], a large number of papers in the theoretical computer science community have expanded the models and results in various ways [7, 18, 23, 24, 25, 27, 26, 28, 30, 46, 49, 48, 54, 61]. The main focus in the community has continued to be the ability of small-world networks to route greedily and efficiently. In particular, the goal has been to find ways to augment graphs with suitable long-range links or (semi-)metrics, provide nodes with additional knowledge or let them perform some local graph exploration, or exploit non-uniformity in node degrees, all in an effort to achieve routing along paths of essentially optimal length. Several good recent surveys summarize the work along these lines [22, 42, 50].

3 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."

We begin with some formalities. Throughout, V is a *ground set* of n nodes. For a metric $\mathcal{D}$, we use the standard notion of balls, i.e., $B(u,r) = \{v \mid \mathcal{D}(u,v) \leq r\}$. 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 $c \geq 1$.

3.1 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*[8], that is, the nodes lie in $[0,R]^d$ for some R, and the distance between points $x, y \in [0,R]^d$ 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 at least one and at most C_{UD} nodes, for some known constant $C_{\text{UD}} \geq 1$. (Since C_{UD} will always be a constant, we will 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., [40, 41]), we focus on the cases $k_{\text{sg}} = O(1)$ and

[8] 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, which gets more pronounced in higher dimensions.

$k_{\mathrm{sg}} = \mathrm{polylog}(n)$. 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 $k_{\mathrm{sg}} \geq 1/C_{\mathrm{sg}}$, 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 that the d-dimensional grid is always part of the graph.) This issue is discussed in more detail in Section 7, in the context of low node degrees.

Our main result easily extends to a more general model in which, for a suitably large $R = \mathrm{polylog}(n)$, 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.

3.2 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. $E_{\mathrm{sg}} = \bigcup_{i=1}^{K} E_{\mathrm{sg}}^{(i)}$. 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 $k_{\mathrm{sg}}^{(i)}$ 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 $k_{\mathrm{sg}} = \frac{1}{K} \cdot \sum_i k_{\mathrm{sg}}^{(i)}$.

3.3 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 $r, r' = O(\mathrm{polylog}(n))$,

$$|B_i(u, r) \cap B_{i'}(u', r')| \leq O(\log n). \tag{1}$$

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 6. The Local Category-Disjointness condition is not overly strong; for instance, we prove (in Section 8) that both Local Category-Disjointness and Scale-R Category-Disjointness hold with high probability when node identifiers within each category are randomly permuted.

3.4 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 $E_{\mathrm{sg}}^{(i)}$ 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 $k_{\mathrm{sg}}^{(i)}$ 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 $k_{\mathrm{sg}}^{(i)}$ 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 $C_{\mathrm{sg}} = \Theta(\frac{1}{\log n})$ 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.

3.5 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 [17, pages 6–8].

Theorem 3.1 (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:*

$$\operatorname{Prob}\left[\,|X-\mu| > \delta\mu\,\right] \leq \exp(-\mu\,\delta^2/3), \qquad \textit{for any } \delta > 0 \tag{2}$$

$$\operatorname{Prob}\left[\,X > (1+\delta)\mu'\,\right] \leq \exp(-\mu'\,\delta^2/3), \qquad \textit{for any } \delta \in (0,1). \tag{3}$$

The bounds in Theorem 3.1 sometimes apply (and are useful) even when the summands are not independent. In particular, our analysis of Local Category-Disjointness and Scale-R Category-Disjointness in Section 8 uses one such result in which the randomness arises from a random permutation. We state and prove the corresponding version of Chernoff Bounds in that section.

4 The main result

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

Theorem 4.1. *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* $\operatorname{polylog}(n)$ *additive error. Moreover, such distance estimates (as spanner graphs or as distance labels) can be computed in time $n \operatorname{polylog}(n)$.*

4.1 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 $E_{\text{sg}} = \bigcup_i E_{\text{sg}}^{(i)}$ of edges from all categories. For the entire section, we assume that the average node degree is high enough: $C_{\text{sg}}k_{\text{sg}} = \Omega(16^d K^3 \log n)$ for a sufficiently large constant in $\Omega(\cdot)$. Let $r_{\text{loc}} = \Theta((C_{\text{sg}}k_{\text{sg}})^{1/d})$ be the *local radius*: by definition of the generative model, all edges between node pairs (u, v) at distance $\mathcal{D}(u, v) \le r_{\text{loc}}$ are in E_{sg} with probability 1. We define the *pruning radius* to be $r_{\text{pru}} = \Theta(r_{\text{loc}}K^{2/d})$.

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

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. [64] 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 $E_{\text{amb}}^{(i)}$ (which need not be disjoint); the set $E_{\text{amb}}^{(i)}$ 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 $(u, v) \in E_{\text{amb}}^{(i)}$, whereas $(u, v) \notin E_{\text{amb}}^{(i)}$ whenever u and v are at distance at least $r_{\text{amb}} = \Theta(r_{\text{pru}}K^{3/d}) = \Theta(r_{\text{loc}}K^{5/d})$. 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 $E_{\text{amb}}^{(i)}$ 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 $E^{(i)}_{\text{amb}}$ 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 $E^{(j)}_{\text{amb}}$. (We show using the Local Category-Disjointness condition that the node set of this clique will have diameter at most $4r_{\text{pru}}$ 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 $E^{(i)}_{\text{amb}}$ 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.

4.2 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:

$$\text{for each node pair } (u, u')\text{, accept if } M_\Lambda(u, u') \geq M_\Lambda\text{, reject otherwise.} \tag{4}$$

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 4.2. *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

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

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 define $S_{i,i'}$ to be the set of all nodes v such that $(u, v) \in E^{(i)}_{\text{sg}}$ and $(u', v) \in E^{(i')}_{\text{sg}}$. 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:

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

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':

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

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.[9] 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

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

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

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

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

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

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

(when $C_{\text{sg}} k_{\text{sg}}$ 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 4.2, i.e., we assume that E_{pru} contains all edges of length at most r_{loc} (in at least one category) and no edges whose length would exceed r_{pru} in all categories.

Notice that in the single-category case ($K = 1$), the result of Lemma 4.2 by itself already gives an expansion of $r_{\text{pru}}/r_{\text{loc}} = \Theta(1)$, 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_{pru}. Lemma 4.3 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 4.3. *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 $\mathcal{D}^{\mathsf{sp}}$ be the shortest-paths metric of the unit disk graph on (V, D), and assume that $\mathcal{D}^{\mathsf{sp}}(u, v) \leq c\,\mathcal{D}(u, v)$ 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 $\mathcal{D}^{\mathsf{sp}}$. 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 $\mathcal{D}_G(u, v) \leq \lceil \frac{\mathcal{D}^{\mathsf{sp}}(u,v)}{r} \rceil \leq 1 + \frac{c\,\mathcal{D}(u,v)}{r}$. □

4.3 Amoeba stage: mapping 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_{pru} are mapped to this category. These edges constitute the edge set $E^{(i)}_{\text{amb}}$. Eventually, each edge $e \in E_{\text{pru}}$ 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_{pru}, and a *growth phase*, in which we grow $E^{(i)}_{\text{amb}}$ 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 $\text{diam}_j(S)$ be its diameter in $E^{(j)}_{\text{amb}}$. 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 (5) is called the *Amoeba Test:* more precisely, edge (u, v) passes the test if and only if (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 $\text{diam}_j(C) \geq \log^2(n)$ 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

   $$E_{\rm sg} \text{ contains at least } M_{\rm amb} \text{ edges between } u \text{ and } \Gamma(v, E_{\rm amb}), \tag{5}$$

   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 $\text{diam}_j(S)$ 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 (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 $\gamma_{\rm amb}$, and call it the *amoeba radius.*[10]

4.4 Analysis of the Amoeba stage

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

The high-level outline of the correctness proof for Amoeba is as follows. We will prove by induction on i that each edge set $E^{(i)}_{\rm amb}$ 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 $E_{\rm amb}$ remains an i-amoeba throughout the algorithm. We break the induction step into multiple lemmas capturing the following four key points:

- The required seed clique C of size $N_{\rm amb}$ exists in $E_{\rm pru}$.
- All edges in the seed clique have sufficiently small length.
- No i-long edge passes the Amoeba Test.

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

- While there is an i-short edge not yet added to E_{amb}, at least one such edge passes the Amoeba Test.

Lemma 4.4. *If there is an edge e not included in any $E_{amb}^{(j)}$, then E_{pru} contains a clique of at least N_{amb} nodes whose diameter in $E_{amb}^{(j)}$ is at least $\log^2(n)$ for all j.*

Proof. Let $e \in E_{\text{pru}}$ be an edge not included in $E_{\text{amb}}^{(j)}$ 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 $v \in B \setminus B_j(u, r_{\text{amb}} \cdot \log^2(n))$. Because each edge in $E_{\text{amb}}^{(j)}$ 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 $E_{\text{amb}}^{(j)}$. Since this holds for all j, B is a candidate for seed clique i, and the algorithm thus guarantees progress. $\square$

Lemma 4.5. *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. $\square$

Remark. Lemma 4.5 can be restated as saying that for any edge-coloring of a sufficiently large clique that is consistent with the Local Category-Disjointness condition[11], there is a color i such that the set of edges of color i has diameter at most 4. Without the Local Category-Disjointness condition, this statement is false in general for $K \geq 3$. For a simple counter-example, consider a clique C whose nodes are partitioned into three sets C_1, C_2, C_3 so that color $i \in \{1, 2, 3\}$ is assigned to all edges with both endpoints in C_i and to all edges with neither endpoint in C_i. Then, the edge set corresponding to any one color i is not even connected. For $K = 2$, there is a simple combinatorial proof that does not involve Local Category-Disjointness.

Lemma 4.6. *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.*

[11] Reformulated in terms of edge colorings, the Local Category-Disjointness states that two balls with respect to edges of colors $i \neq i'$, each of radius polylog(n), overlap in at most $O(\log n)$ nodes.

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, we claim that $|\Gamma(v, E_{\text{amb}}) \setminus B_i(v, r_{\text{pru}})| \leq O(K \log n)$. 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 $q = O(C_{\text{sg}} k_{\text{sg}} r_{\text{amb}}^{-d}) = O(C_{\text{sg}} k_{\text{sg}} / (\gamma_{\text{amb}}^d K^3) \cdot r_{\text{pru}}^{-d})$. Summing over all $w \in B_i(v, r_{\text{pru}})$ and all categories gives us at most $qK|B_i(v, r_{\text{pru}})| = O(C_{\text{sg}} k_{\text{sg}} / (\gamma_{\text{amb}}^d K^2))$ 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 4.7. *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, the ball $B' = B_i(w, r_{\text{loc}}/4)$ 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-amoebas $E_{\text{amb}}^{(i)}, i = 1, \ldots, K$. The distance $\mathcal{D}_i(u,v)$ is now estimated as the shortest-path distance between u and v in $E_{\text{amb}}^{(i)}$, multiplied by r_{amb}. By Lemma 4.3, this gives constant expansion $r_{\text{amb}}/r_{\text{loc}} = \Theta(K^{5/d})$, no contraction, and additive error r_{amb}.

4.5 Efficient Implementation of the Amoeba algorithm

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 $E_{\text{amb}}^{(i)}$ for each category i in near-linear time. Recall that $E_{\text{amb}}^{(i)}$ 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. [32, 13, 69, 70]). We next describe each of these steps in more detail.

4.5.1 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 4.8. *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 4.8, 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.

4.5.2 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 (5) holds. 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 (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.

4.5.3 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 distance estimates 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).

5 Improving the distortion for a single category

Our first improvement is to reduce the distortion from a multiplicative constant to a factor $1+o(1)$. In fact, under stronger assumptions on the uniformity of the metric space, we will be able to reduce the distortion to additively polylogarithmic. We first show the improvement for a single category, and discuss the necessary extensions for multiple categories in Section 6.

In trying to improve the distortion beyond a multiplicative constant, we face an immediate obstacle: as discussed in Section 3, an algorithm can estimate the normalization constant C_{sg} and the target degree k_{sg} only up to a constant factor. However, for further improvements of the distortion, more accurate estimates of C_{sg} and k_{sg} appear to be necessary. In order to side-step this technical obstacle, we define *normalized distances*

$$\mathcal{N}(u, v) = \mathcal{D}(u, v)/(C_{\text{sg}}\, k_{\text{sg}})^{1/d}, \tag{6}$$

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

Note that Theorem 4.1 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. In this section, we improve this bound to unit distortion with sub-linear additive error.

Theorem 5.1. *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 4 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_{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_{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 $E_{\text{sg}}$ and initial estimates $\mathcal{N}^*$ from Theorem 4.1.
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 $E_{\text{sg}}$ 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 $\kappa$ 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" (the 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, while the latter decreases with κ.

Proof of Theorem 5.1. Assume that $\mathcal{N}^*$ satisfies the high-probability property that it is an estimate of $\mathcal{N}$ with constant distortion and polylog(n) additive error. Consider a node pair (s,t), at normalized distance $y = \mathcal{N}(s,t)$.

Assume that $\mathcal{N}(s,t)$ is large enough to ensure that r_y is larger than the polylog(n) additive error. (Otherwise, the additive error guarantee is trivially satisfied.) Then, by near-uniform density, all nodes in $\tilde{B}_s(\kappa; \mathcal{N}^*)$ are at normalized distance at most $c\, r_y$ from s, for some constant c. Likewise, all nodes in $\tilde{B}_t(\kappa; \mathcal{N}^*)$ are at normalized distance at most $c\, r_y$ from t. Therefore

$$\frac{\kappa^2}{(y + 2c\, r_y)^d} \leq \mathbb{E}\left[\tilde{M}_{s,t}\right] \leq \frac{\kappa^2}{(y - 2c\, r_y)^d}. \tag{7}$$

We next apply Chernoff bounds to $\tilde{M}_{s,t}$, and use the bounds that $\frac{1}{1-2\beta} \cdot (1 + 6\beta) \leq \frac{1}{1-8\beta}$ and $\frac{1}{1+2\beta} \cdot (1 - 6\beta) \geq \frac{1}{1+8\beta}$ (with $\beta = c\,\frac{r_y}{y}$) to derive that

$$\text{Prob}\left[\frac{\kappa^2}{(y + 8c\, r_y)^d} \leq \tilde{M}_{s,t} \leq \frac{\kappa^2}{(y - 8c\, r_y)^d}\right] \geq 1 - 1/n^{O(\log n)}.$$

Taking the union bound over all node pairs (s,t), it follows that w.h.p. $|(\kappa^2/\tilde{M}_{s,t})^{1/d} - y| \leq O(r_y)$. □

5.1 The 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 5.1.

The analysis of this algorithm is significantly more delicate and involved. 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 $C_{\text{PD}}\, r^d \pm O(r^{d-1})$ points, where C_{PD} is a known constant. Then we can improve the additive error to $\text{polylog}(n)$.

Theorem 5.2. *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* $\text{polylog}(n)$.

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 5.2, 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 4.1. 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 $x = \mathcal{N}^*(s,t)$.

5.2 Proof of Theorem 5.2

The high-level idea of the analysis is as follows. 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

Algorithm 3 The Recursive Two-Ball Algorithm.

```
Inputs. Original edge set $E_{\text{sg}}$ and initial estimates $\mathcal{N}^*$ from Theorem 4.1.
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 $E_{\text{sg}}$ 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 $\kappa$ closest nodes to node $u$ according to $\mathcal{N}'$, breaking ties arbitrarily.
$c_d$ is the constant from the remark after Theorem 5.2.
```

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 $\rho(x) = O(\log\log n)$.

Consider two nodes s and t whose normalized distance is $x = \mathcal{N}(s,t)$.

Let $\tilde{B}_s = \tilde{B}_u(\kappa;\mathcal{N})$ and $\tilde{B}_t = \tilde{B}_u(\kappa;\mathcal{N})$ be the sets of the κ closest nodes to s and t, respectively, under the (correct) normalized distances $(V,\mathcal{N})$.

We start with a simple lemma showing that this choice implies that the actual *sets* of nodes are very close between $\tilde{B}'_s$ and $\tilde{B}_s$ (and $\tilde{B}'_t$ and $\tilde{B}_t$, respectively).

Lemma 5.3. *For a sufficiently large constant β, we have that*

$$B_{\mathcal{N}}(s, \hat{r}_x - 2a(\hat{r}_x) - \beta) \subseteq \tilde{B}'_s \subseteq B_{\mathcal{N}}(s, \hat{r}_x + 2a(\hat{r}_x) + \beta),$$
$$B_{\mathcal{N}}(t, \hat{r}_x - 2a(\hat{r}_x) - \beta) \subseteq \tilde{B}'_t \subseteq B_{\mathcal{N}}(t, \hat{r}_x + 2a(\hat{r}_x) + \beta).$$

Proof. We first prove the first inclusion. Let $v \in B_{\mathcal{N}}(s, \hat{r}_x - 2a(\hat{r}_x) - \beta)$ be arbitrary. Because $\mathcal{N}(s,v) \leq \hat{r}_x - 2a(\hat{r}_x) - \beta$, the definition of $a(\cdot)$ implies that $\mathcal{N}'(s,v) \leq \mathcal{N}(s,v) + a(\hat{r}_x) \leq \hat{r}_x - a(\hat{r}_x) - \beta$. On the other hand, $\mathcal{N}'(s,u) \geq \hat{r}_x - a(\hat{r}_x) - \beta$ for all nodes u such that $\mathcal{N}(s,u) \geq \hat{r}_x - \beta$. Therefore, there can be at most $C_{\text{PD}}\,(\hat{r}_x - \beta)^d \pm O((\hat{r}_x - \beta)^{d-1})$ nodes u with $\mathcal{N}'(s,u) \leq \mathcal{N}'(s,v)$. This number is less than $C_{\text{PD}}\,\hat{r}_x^d = \kappa$ whenever β is large enough.

Because $\tilde{B}'_s$ contains the κ nodes closest to s under $\mathcal{N}'$ (by its definition), this means that $v \in \tilde{B}'_s$. Since this argument holds for arbitrary v, we have proved the first claim. The second inclusion is proved by an analogous calculation. □

We next show that the number of edges between $\tilde{B}_s$ and $\tilde{B}_t$ is close to the number of edges between $\tilde{B}'_s$ and $\tilde{B}'_t$. To state this claim concisely, let $\#\texttt{edges}(S,S')$ be the number of edges in E_{sg} between node sets S and S'.

Lemma 5.4. *With high probability,* $|\mathbb{E}\left[\#\texttt{edges}(\tilde{B}'_s,\tilde{B}'_t)\right] - \mathbb{E}\left[\#\texttt{edges}(\tilde{B}_s,\tilde{B}_t)\right]| = O(x \cdot a(\hat{r}_x))$.

Proof. We construct a bijection $\phi : (\tilde{B}'_s \cup \tilde{B}'_t) \to (\tilde{B}_s \cup \tilde{B}_t)$ as follows. Partition the domain and the co-domain into four disjoint regions each (using $\oplus$ to denote the disjoint union of sets):

$$(\tilde{B}'_s \cup \tilde{B}'_t) = (\tilde{B}'_s \cap \tilde{B}_s) \oplus (\tilde{B}'_s \setminus \tilde{B}_s) \oplus (\tilde{B}'_t \cap \tilde{B}_t) \oplus (\tilde{B}'_t \setminus \tilde{B}_t),$$
$$(\tilde{B}_s \cup \tilde{B}_t) = (\tilde{B}'_s \cap \tilde{B}_s) \oplus (\tilde{B}_s \setminus \tilde{B}'_s) \oplus (\tilde{B}'_t \cap \tilde{B}_t) \oplus (\tilde{B}_t \setminus \tilde{B}'_t).$$

The regions in each partition are indeed disjoint because $\tilde{B}_s \cap \tilde{B}_t = \tilde{B}'_s \cap \tilde{B}'_t = \emptyset$. We define ϕ separately for each of the four subsets the domain. First, any node in $(\tilde{B}'_s \cap \tilde{B}_s)$ or $(\tilde{B}'_t \cap \tilde{B}_t)$ is mapped to itself. Second, ϕ is an arbitrary bijection $(\tilde{B}'_s \setminus \tilde{B}_s) \to (\tilde{B}_s \setminus \tilde{B}'_s)$ and $(\tilde{B}'_t \setminus \tilde{B}_t) \to (\tilde{B}_t \setminus \tilde{B}'_t)$. This completes the definition. For the second step, note that the respective domains and co-domains have the same size; this is because $|\tilde{B}_s| = |\tilde{B}'_s| = \kappa$ and $|\tilde{B}_t| = |\tilde{B}'_t| = \kappa$.

Nodes $v \in (\tilde{B}'_s \setminus \tilde{B}_s) \cup (\tilde{B}'_t \setminus \tilde{B}_t)$ called *perturbed nodes*. By Lemma 5.3, $\tilde{B}'_s$ and $\tilde{B}'_t$ contain at most $C_{\mathrm{PD}} \cdot (2a(\hat{r}_x) + \beta) \cdot \hat{r}_x^{d-1}$ perturbed nodes each.

By the perfectly uniform density assumption, at least $C_{\mathrm{PD}}\, r^d - O(r^{d-1})$ nodes have distance at most r from s. In particular, setting $r = \hat{r}_x + \beta$ gives us that at least κ nodes satisfy the distance bound, implying that every node $u \in \tilde{B}_s$ satisfies $\mathcal{N}(s,u) \leq \hat{r}_x + \beta$, Furthermore, by the second inclusion of Lemma 5.3, every node $v \in \tilde{B}'_s$ satisfies $\mathcal{N}(s,v) \leq \hat{r}_x + 2a(\hat{r}_x) + \beta$. Similar bounds apply for t. We thus get that $\mathcal{N}(v,\phi(v)) \leq 2\hat{r}_x + 2a(\hat{r}_x) + O(1) < 3\hat{r}_x$ for all v, and of course $\mathcal{N}(v,\phi(v)) = 0$ for unperturbed nodes v.

Now consider a pair $u \in \tilde{B}'_s$ and $v \in \tilde{B}'_t$ such that at least one of u, v is perturbed. (We call such a pair a *perturbed pair*.) By triangle inequality, $|\mathcal{N}(v,u) - \mathcal{N}(\phi(v),\phi(u))| \leq 6\hat{r}_x$, and the number of perturbed pairs is at most $(4a(\hat{r}_x) + 2\beta) \cdot \hat{r}_x^{2d-1}$, by the bound on the number of perturbed nodes.

Next, we bound how much a single perturbed pair $u \in \tilde{B}'_s, v \in \tilde{B}'_t$ affects the expected number of edges between the balls. Because $x + 6\hat{r}_x \geq \mathcal{N}(\phi(u),\phi(v)) \geq x - 6\hat{r}_x$, we get that

$$\frac{\mathcal{N}(u,v)}{\mathcal{N}(\phi(u),\phi(v))} \in 1 \pm O(\hat{r}_x/x).$$

We can now express the difference between the probabilities of the edges $(\phi(u),\phi(v))$ and (u,v) as

$$\begin{aligned}\left|(x \pm O(\hat{r}_x))^{-d} - (x \pm 2\hat{r}_x)^{-d}\right| &= x^{-d} \cdot \left|\left(1 \pm \frac{O(\hat{r}_x)}{x}\right)^{-d} - (1 \pm \frac{2\hat{r}_x}{x})^{-d}\right| \\ &= O\left(x^{-d} \cdot \left(\frac{1}{1 \pm O(\hat{r}_x/x)} - \frac{1}{1 \pm 2\hat{r}_x/x}\right)\right) \\ &= O\left(x^{-d} \cdot \hat{r}_x/x\right).\end{aligned}$$

In the second step, we truncated the Binomial expansion (because $\hat{r}_x/x = o(1/d)$), and the final step again used that $\hat{r}_x/x$ is small. Summing over all perturbed pairs, the total expected difference in the number of edges can be bounded by above as follows:

$$\left|\mathbb{E}\left[\#\mathtt{edges}(\tilde{B}'_s, \tilde{B}'_t) - \#\mathtt{edges}(\tilde{B}_s, \tilde{B}_t)\right]\right| \leq O\left(\frac{\hat{r}_x}{x} \cdot x^{-d} \cdot a(\hat{r}_x) \cdot \hat{r}_x^{2d-1}\right) = O(x\, a(\hat{r}_x)),$$

where the last step was obtained by substituting the definition of $\hat{r}_x$. The concentration now follows from Chernoff Bounds. □

Lemma 5.5. $a(x) = O(a(\hat{r}_x))$.

Proof. Consider two nodes s and t at normalized distance x. Using an analysis very similar to the one in the proof of Lemma 5.4, the expected number of edges between $\tilde{B}_s$ and $\tilde{B}_t$ is $(c_d\,\hat{r}_x^2/x)^d \pm O(x) = c_d^d x^2 \pm O(x)$ (where c_d is the constant from the remark after Theorem 5.2). $\hat{r}_x$ is chosen so that Chernoff Bounds ensure that w.h.p., the actual number of edges between $\tilde{B}_s$ and $\tilde{B}_t$ does not deviate from its expectation by more than $O(x \cdot a(\hat{r}_x))$. Combining this number of edges with the bound from Lemma 5.4, the expected number of edges between $\tilde{B}'_s$ and $\tilde{B}'_t$ is $\tilde{M}_{s,t} = c_d^d x^2 \pm O(x \cdot a(\hat{r}_x))$ with high probability. (The big-O term combines both the misestimates bounded by the Chernoff Bound and the ones from Lemma 5.4.)

Because the algorithm estimates the distance as $c_d\,\hat{r}_x^2 \tilde{M}_{s,t}^{-1/d}$, the additive distortion is at most

$$\begin{aligned}
\left|x - c_d\,\hat{r}_x^2\,\tilde{M}_{s,t}^{-1/d}\right| &= x \cdot \left|1 - \frac{c_d\,x^{2/d}}{(c_d^d\,x^2 \pm O(xa(\hat{r}_x)))^{1/d}}\right| \\
&= x \cdot \left|1 - \left(\frac{c_d^d\,x}{c_d^d\,x \pm O(a(\hat{r}_x))}\right)^{1/d}\right| \\
&= x \cdot \left|1 - \left(1 \pm \frac{O(a(\hat{r}_x))}{c_d^d\,x \pm O(a(\hat{r}_x))}\right)^{1/d}\right| \\
&\leq x \cdot \frac{O(a(\hat{r}_x))}{c_d^d\,x \pm O(a(\hat{r}_x))} \\
&\leq O(a(\hat{r}_x)).
\end{aligned}$$

In the penultimate inequality, we used that $|1 - (1 \pm \delta)^{1/d}| \leq \delta$ for any δ, and the final inequality used that $a(\hat{r}_x) = o(x)$ to simplify the denominator. □

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 $\rho(x) = O(\log\log n)$. This completes the proof of Theorem 5.2.

6 Improving the distortion for multiple categories

In order to improve the estimates for multiple categories, we employ the two algorithms from Section 5. 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:

$$\#\texttt{pairs}_j(B, B', r) \triangleq |\{(u, u') \mid u \in B,\, u' \in B',\, \mathcal{D}_j(u, u') < r\}|. \tag{8}$$

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:

$$\#\texttt{pairs}_j(B, B', r) \leq O(r^d/n) \cdot |B|\,|B'| + O(\log^2 n). \tag{9}$$

Scale-R Category-Disjointness asserts that (9) holds "locally:" at all distance scales up to R.

Definition 6.1. The *Scale-R Category-Disjointness* condition states that (9) 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 (9) for randomly permuted categories is derived in Section 8. 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 (Section 6.1) and an analysis of the Recursive Two-Ball Algorithm for multiple categories (Section 6.2).

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 $\mathcal{N}_i(u,v) \triangleq \mathcal{D}_i(u,v)/(C_{\text{sg}}^{(i)}\, k_{\text{sg}}^{(i)})^{1/d}$.

6.1 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 (9) 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 6.2. *Assume the setting of Theorem 4.1 with Scale-$R^{1+1/(d+1)}$ Category-Disjointness, $R \geq \text{polylog}(n)$ 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 $1 \pm \left[(\min(x, R, \hat{R}))^{-d/(2d+2)} \cdot O(\log^2 n)\right]$, where $\hat{R} = \left(\frac{n}{\log n}\right)^{(2d+2)/(2d^2+3d)}$.*

Remark. The distortion in Theorem 6.2 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, in some sense, 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 4.1. Recall that these are non-contracting estimates with constant expansion δ

and 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 6.2.

The algorithm proceeds as follows. (See Algorithm 4 for the pseudocode). 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

$$H_t = \{(u,v) \in H \mid \mathcal{N}^*(u,v) \geq \tfrac{R}{2\delta} \text{ or } v = t\}. \tag{10}$$

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.

Algorithm 4 The Extended Two-Ball Algorithm (for a given category i).

```
Inputs. Original edge set $E_{\text{sg}}$ and initial estimates $\mathcal{N}^* = \mathcal{N}^*_i$ from Theorem 4.1.
Parameters. Distance scale $R$ and expansion $\delta$ of $\mathcal{N}^*$.
Output. Improved distance estimates $\mathcal{N}'_i$.

$H = \{(u,v) \mid \mathcal{N}^*(u,v) \leq R\}$.

The Two-Ball Algorithm. For each node pair $(s,t) \in H$,
    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 $E_{\text{sg}}$ between $\tilde{B}^*_s$ and $\tilde{B}^*_t$.
    3. $\mathcal{N}_H(s,t) = (\kappa^2/\tilde{M}_{s,t})^{1/d}$.

Post-processing. For each node pair $(s,t)$,
If $(s,t) \in H$, then $\mathcal{N}'_i(s,t) = \mathcal{N}_H(s,t)$; otherwise
    1. $H_t = \{(u,v) \in H \mid \mathcal{N}^*(u,v) \geq \frac{R}{2\delta}$ or $v = t\}$.
    2. $\mathcal{N}'_i(s,t)$ is the length of the shortest $s$-$t$ path in $H_t$ with respect to edge lengths $\mathcal{N}_H$.

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

6.1.1 Analysis: the Two-Ball Algorithm for multiple categories

We begin the analysis by showing that for sufficiently small distances, Scale-R Category-Disjointness ensures that the basic Two-Ball Algorithm gives accurate estimates.

Lemma 6.3. *Assume that the Scale-$R^{1+1/(d+1)}$ Category-Disjointness condition holds, and let (s,t) be a node pair at normalized category-i distance $\mathcal{N}_i(s,t) = x \leq R$. Then, the Two-Ball Algorithm obtains a distance estimate $\mathcal{N}'_i(s,t)$ of $\mathcal{N}_i(s,t)$ with the following guarantee:*

$$\left|\mathcal{N}'_i(s,t) - \mathcal{N}_i(s,t)\right| \leq \left(x^{(d+2)/(2d+2)} + \tfrac{x^{d+1}}{n}\right) \cdot O(\log^2 n).$$

Proof. Recall from the proof of Theorem 5.1 that to estimate $\mathcal{N}_i(s,t)$, the Two-Ball Algorithm considers two balls $\tilde{B}^*_s$, $\tilde{B}^*_t$ around s and t, respectively, and counts edges between them. The balls were chosen so that $|\tilde{B}_s^*| = |\tilde{B}_t^*| = \kappa \triangleq r_x^d$, where $r_x = x^{(d+2)/(2d+2)}$. The improved distance estimate is $\mathcal{N}'(s,t) \triangleq (\kappa^2/\tilde{M}_{s,t})^{1/d}$, where $\tilde{M}_{s,t}$ is the number of edges between $\tilde{B}_s^*$ and $\tilde{B}_t^*$.

If only edges from $E_{\text{sg}}^{(i)}$ were counted, Theorem 5.1 would apply verbatim. However, edges between $\tilde{B}_s^*$ and $\tilde{B}_t^*$ from other categories can be erroneously included in the count. The presence of other categories never *decreases* $\tilde{M}_{s,t}$, so the high-probability lower bound on $\tilde{M}_{s,t}$, and hence the high-probability upper bound on $\mathcal{N}'(s,t)$, carries over from Theorem 5.1.

We need to prove a lower bound on $\mathcal{N}'(s,t)$. Let $\tilde{M}_{s,t}^{(i)}$ be the number of category-i edges between $\tilde{B}_s^*$ and $\tilde{B}_t^*$. In the proof of Theorem 5.1, we showed that with high probability, $\tilde{M}_{s,t}^{(i)} \le \frac{\kappa^2}{(x-8c\,r_x)^d}$, for some constant c. This implies $\tilde{M}_{s,t}^{(i)} \le \frac{\kappa^2}{x^d}(1 + O(c\,r_x/x))^d \le \frac{\kappa^2}{x^d}(1 + O(cd\,r_x/x))$.

We next count edges from other categories between $\tilde{B}_s^*$ and $\tilde{B}_t^*$. Fix some category $j \ne i$, and consider node pairs $(u \in \tilde{B}_s^*, u' \in \tilde{B}_t^*)$. We distinguish between two distance scales for $\mathcal{N}_j(u,u')$.

1. We first consider the case that $\mathcal{N}_j(u,u') > R^{1+1/(d+1)}$. The probability for the edge (u,u') to exist is then at most $O(R^{-(d+1-1/(d+1))})$. The number of candidate pairs (u,u') is at most $\kappa^2 = x^{d+1-1/(d+1)} \le R^{d+1-1/(d+1)}$, so the expected number of such long edges is $O(1)$. Using Chernoff Bounds, with high probability, the number of long edges is at most $O(\log^2 n)$.

2. The other case is $\mathcal{N}_j(u,u') \le R^{1+1/(d+1)}$. We divide the range of possible distances into exponentially increasing buckets of the form $(y, 2y]$. Suppose that $y \le \mathcal{N}_j(u,u') \le 2y$ (for some $y \le R/2$). Then, the pair (u,u') has an edge with probability at most $O(y^{-d})$, and by the Scale-$R^{1+1/(d+1)}$ Category-Disjointness condition, there are at most $O(y^d/n) \cdot |\tilde{B}_s^*|\,|\tilde{B}_t^*| + O(\log^2 n)$ pairs (u,u') at this distance scale. Using linearity of expectations, and summing over all $O(\log n)$ distance scales y, we obtain that the expected number of short category-j edges between $\tilde{B}_s^*$ and $\tilde{B}_t^*$ is at most $O(\frac{|\tilde{B}_s^*|\,|\tilde{B}_t^*|\,\log n}{n} + \log^2 n)$, and Chernoff Bounds establish concentration.

Combining both cases, and substituting that $|\tilde{B}_s^*| = |\tilde{B}_t^*| = \kappa$ gives us that with high probability, the number of category-j edges between $\tilde{B}_s^*$ and $\tilde{B}_t^*$ is at most $O(\frac{\log n}{n} \cdot \kappa^2 + \log^2 n)$. Combining these edges across all categories $j \ne i$ and plugging in the upper bound for $\tilde{M}_{s,t}^{(i)}$, we obtain:

$$\tilde{M}_{s,t} \le \frac{\kappa^2}{x^d}\left(1 + O\left(cd\,\frac{r_x}{x}\right)\right) + O(K)\left(\tfrac{\log n}{n} \cdot \kappa^2 + \log^2 n\right).$$

Adding some $\log n$ factors for simplification, and hiding the constants inside $O(\cdot)$, we can re-write this bound as follows:

$$\tilde{M}_{s,t} \le \frac{\kappa^2}{x^d}\left(1 + O(\log^2 n)\left(x^{-d/(2d+2)} + \tfrac{x^d}{n}\right)\right).$$

Substituting the definition $\mathcal{N}'(s,t) \triangleq (\kappa^2/\tilde{M}_{s,t})^{1/d}$, it follows that

$$\mathcal{N}_i'(s,t) \ge x\,\left(1 - O(\log^2 n)\left(x^{-d/(2d+2)} + \tfrac{x^d}{n}\right)\right) \ge x - O(\log^2 n)\left(x^{(d+2)/(2d+2)} + \tfrac{x^{d+1}}{n}\right). \qquad \square$$

6.1.2 Analysis: the post-processing step

Theorem 6.2 easily follows from Lemma 6.3 and the following Lemma 6.4, which analyzes the post-processing step. The lemma is not specific to the actual estimates produced by the Two-Ball

Algorithm. Instead, it states that if each individual edge's length is estimated with small additive distortion (compared to the true edge length), then the multiplicative distortion of the overall estimates is small. For readability, we continue to omit the subscript i from all metrics.

Lemma 6.4. *Assume the setting of Theorem 4.1, and let δ be the expansion in $\mathcal{N}^*$. Consider running the post-processing step of the Extended Two-Ball Algorithm (parameterized by some R) on distance estimates $\mathcal{N}_H$ satisfying the following for some $\Delta < \frac{R}{4\delta^2}$:*

$$|\mathcal{N}_H(u,v) - \mathcal{N}(u,v)| \leq \Delta \quad \textit{for all } (u,v) \in H. \tag{11}$$

Then, the final estimates $\mathcal{N}'(s,t)$ have multiplicative distortion $1 + O(\delta^2\Delta/R)$ for all node pairs (s,t) not in H.

Proof of Theorem 6.2. Without loss of generality, assume that $R \leq \hat{R}$, where $\hat{R}$ is from the theorem statement. (If $R > \hat{R}$, then we could parameterize the algorithm by $\hat{R}$ instead.) Then the upper bound in Lemma 6.3 becomes $\Delta_x \triangleq x^{(d+2)/(2d+2)} \cdot O(\log^2 n)$.

To complete the proof of Theorem 6.2, notice that all edges $(u,v) \in H$, by definition, satisfy $\mathcal{N}^*(u,v) \leq R$. As $\mathcal{N}^*$ is non-contracting, this also implies that $\mathcal{N}(u,v) \leq R$, so the bound (11) holds with $\Delta = \Delta_R$, according to Lemma 6.3. If $(s,t) \in H$ (which happens when $\mathcal{N}^*(s,t) \leq R$), then we can apply Lemma 6.3 directly to the edge (s,t), obtaining the bound in terms of x. □

Proof of Lemma 6.4. Fix a node pair $(s,t) \notin H$, and let $x = \mathcal{N}(s,t)$. Because $(s,t) \notin H$, and the estimate $\mathcal{N}^*$ has expansion at most δ, we get that $\mathcal{N}(s,t) \geq \frac{1}{\delta}\mathcal{N}^*(s,t) > \frac{R}{\delta}$. Let $H_t \subseteq H$ be the edge set defined in (10), and for any path P, let $\mathcal{N}(P)$ the length of the path P according to the distance function $\mathcal{N}$.

We claim that the edge set H_t contains an s-t path P with $k = \lceil x/(\frac{R}{\delta} - 1) \rceil$ hops and length $\mathcal{N}(P) \leq \mathcal{N}(s,t) + k$. Consider the straight line between s and t in $\mathbb{R}^d$. For each i, let p_i be the point at $\mathcal{N}$-distance $i \cdot (\frac{R}{\delta} - 1)$ from s on the straight line between s and t. The point p_i itself may not be the location of any node in the social network. However, by near-uniform density (which guarantees that every unit cube contains at least one node of the network), each point p_i has a node u_i at distance at most $\mathcal{D}(p_i, u_i) \leq d$. Thus, $\mathcal{N}(p_i, u_i) \leq d/(C_{\text{sg}}\, k_{\text{sg}})^{1/d} \leq \frac{1}{2}$ for large enough n, as $C_{\text{sg}} k_{\text{sg}} = \Omega(\log n)$.

Let P be the path $(s = u_0, u_1, u_2, \ldots, u_{k-1}, t = u_k)$. By triangle inequality, all edges $(u_i, u_{i+1}) \in P$ have $\mathcal{N}$-length within ±1 of the distance $\mathcal{D}(p_i, p_{i+1})$ between the corresponding points p_i. Therefore, $\mathcal{N}(P) \leq \mathcal{N}(s,t) + k$. Moreover, because each edge $(u,v) \in P$ satisfies $\mathcal{N}(u,v) \leq \frac{R}{\delta}$, the fact that $\mathcal{N}^*$ has expansion at most δ implies that $\mathcal{N}^*(u,v) \leq R$. In particular, each edge of P is in H. Furthermore, all edges $(u_i, u_{i+1}) \in P$ except possibly the last one satisfy $\mathcal{N}^*(u_i, u_{i+1}) \geq \mathcal{N}(u_i, u_{i+1}) \geq \frac{R}{\delta} - 2$. By definition of H_t, it follows that the path P is in H_t, completing the proof of the claim.

Next, we upper-bound the estimated distance $\mathcal{N}'(s,t)$. Simply using the path P we just exhibited, we see that

$$\mathcal{N}'(s,t) \;\leq\; \mathcal{N}_H(P) \;\overset{(11)}{\leq}\; \mathcal{N}(P) + k\Delta \;\leq\; \mathcal{N}(s,t) + k(\Delta + 1),$$

where the last inequality used the property that $\mathcal{N}(P) \leq \mathcal{N}(s,t) + k$. An upper bound of $1 + O(k\delta/R)$ on the expansion now follows by substituting $k = O(x\,\frac{\delta}{R})$.

It remains to bound the contraction, by proving that each s-t path P in H_t has $\mathcal{N}_H(P) \geq \mathcal{N}(s,t) - O(x\,\delta^2\Delta/R)$. By the same argument as in the preceding paragraph, this holds whenever

P has at most $4x\,\delta^2/R$ hops. We therefore focus on the case when P has at least $4x\,\delta^2/R$ hops. Each of these hops (u,v), except possibly the last one, has $\mathcal{N}^*(u,v) \geq \frac{R}{2\delta}$ by definition of H. In turn, by the maximum expansion of $\mathcal{N}^*$, the actual length of each hop is at least $\mathcal{N}(u,v) \geq \frac{R}{2\delta^2}$, so that the estimates $\mathcal{N}_H$ satisfy $\mathcal{N}_H(u,v) \geq \mathcal{N}(u,v) - \Delta \geq \frac{R}{4\delta^2}$, because we assumed that $\Delta \leq \frac{R}{4\delta^2}$. Summing over all (at least) $4x\,\delta^2/R$ hops (u,v), we obtain that $\mathcal{N}_H(P) \geq x = \mathcal{N}(s,t)$, so in this case, the estimate has no contraction at all. This completes the proof of the lower bound. □

6.2 The Recursive Two-Ball Algorithm for multiple categories

We show that the Recursive Two-Ball Algorithm from Section 5.1 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 6.5. *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

$$|\mathcal{N}'_i(s,t) - \mathcal{N}_i(s,t)| \leq \operatorname{polylog}(n).$$

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 6.5 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 $(x/R)\operatorname{polylog}(n)$. (This follows from the shortest-path argument encapsulated in Lemma 6.4.)

Proof of Theorem 6.5. The proof of Theorem 5.2 applies almost verbatim. Recall that the Recursive Two-Ball Algorithm counts edges between balls $\tilde{B}'_s$, $\tilde{B}'_t$ around s and t, containing $\kappa = \Theta(\hat{r}_x^d)$ nodes each, where $\hat{r}_x = x^{1/2+1/d}$. These balls are calculated with respect to the distances estimated by the algorithm in earlier stages. The only added difficulty for the analysis in the case of multiple categories is bounding the additional edges between $\tilde{B}'_s$ and $\tilde{B}'_t$ arising from categories $j \neq i$.

Notice that there are $\kappa^2 = O(x^{d+2}) \leq O(x \cdot n)$ pairs of nodes that could have an edge between them. Focus on one category $j \neq i$, and divide node pairs $(u,v), u \in \tilde{B}'_s, v \in \tilde{B}'_t$ into buckets of the form $(y, 2y]$ depending on their distance in category j. By Scale-∞ Category-Disjointness, the bucket $(y, 2y]$ contains at most $O(\frac{y^d}{n} \cdot |\tilde{B}'_s|\,|\tilde{B}'_t| + \log^2 n) = O(y^d \cdot x + \log^2 n)$ node pairs. Each of these node pairs gives rise to an edge with probability at most $O(y^{-d})$, and summing over all $O(\log n)$ buckets $(y, 2y]$ gives us that the expected number of category-j edges between $\tilde{B}'_s$ and $\tilde{B}'_t$ is at most $O(x \log n + \log^2 n) = O(x \log^2 n)$. Using Chernoff Bounds and a union bound over all categories, with high probability, the total number of edges added by categories $j \neq i$ is at most $O(Kx\log^2 n)$.

Because $\log^2 n = O(a(\hat{r}_x))$ for sufficiently large poly-logarithmic x, the $O(Kx\log^2 n) = O(x\,a(\hat{r}_x))$ additional edges are easily subsumed in the error bound of $O(x\,a(\hat{r}_x))$ already present in the proof of Lemma 5.5. For smaller distances x, the only change will be a slightly different poly-logarithmic base case for $a(\hat{r}_x)$. □

7 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 3, 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., [22, 40, 41, 61]) assumes that in addition to the random edges, the network also contains a set E_{loc} of *local edges* deterministically.[12] 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.[13] 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.

In this section, 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 section.

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

Definition 7.1 ("Richness" of local edges).

1. An edge set E is a (σ, δ)-*spanner* if its shortest-path distance $\mathcal{D}^{\mathsf{sp}}$ satisfies the following for all node pairs (u, v):

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

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 $E \subseteq E_{\mathrm{loc}}$ (called its *connectivity witness*).

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

[13] See, e.g., the difficulties faced by [29]. The authors of [29] 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.

Remark. As an example, the d-dimensional toroidal grid is $(2d-1,3)$-rich and (for $d \geq 2$) $(2d,7)$-rich, both with distortion $(1,O(1))$.[14]

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. In Section 7.2, we show how to make the pruning algorithm adapt to the available richness under fairly mild assumptions.

7.1 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. See Algorithm 5 for pseudocode.

Algorithm 5 The (b,h)-EDP Pruning Algorithm.

```
Input. Edge set $E$.
Repeat
    1. Find any $(u,v) \in E$ s.t. $E$ does not contain $b$ edge-disjoint $u$-$v$ paths of at most $h$ hops each.
    2. Prune $(u,v)$ from $E$.
Until no such edges $(u,v)$ remain.
```

The idea is that this algorithm keeps a sufficiently rich subset of local edges, and prunes all edges in E_{sg} whose length exceeds some threshold $r_{\texttt{EDP}}$ (defined in Equation (12)). (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 omit the (easy) modifications of the algorithm and analysis.

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

[14] Fix an edge (u,v). As a base case, for $d=2$, it is easy to construct three paths of lengths $(1,3,3)$, or four paths of lengths $(1,3,5,7)$. For each added dimension, there are two additional disjoint paths of length 3, taking one edge along the new dimension, an edge parallel to (u,v), and another edge in the new dimension. These paths are clearly disjoint.

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 7.2. *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 $\alpha > 0$ (which need not be known to the algorithm), let*

$$r_{\texttt{EDP}}(\alpha) \;=\; D^{(2+\alpha)/b} \cdot h \cdot (O(k_{sg} + \log^{1+\alpha} n))^{2h/d} \;=\; D^{(2+\alpha)/b} \cdot (O(\log n))^{O(h)}. \tag{12}$$

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 $r_{\texttt{EDP}}(\alpha)$. The algorithm makes no guarantees for other edges.

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

$$\mathcal{D}(u, v) \;\leq\; \beta\, \mathcal{D}^{\texttt{sp}}(u, v) \;\leq\; \delta \cdot \beta\, \mathcal{D}(u, v), \;\; \textit{where } \beta = \max(\tfrac{1}{\sigma}, r_{\texttt{EDP}}(\alpha)).$$

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 $r_{\texttt{EDP}}(\alpha)(\mathcal{N}^{\gamma}(u, v) + r_{\texttt{EDP}}(\alpha))$, 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 $d \geq 3$ dimensions. Then the Recursive Two-Ball Algorithm reconstructs all normalized distances with unit distortion and additive error $r_{\texttt{EDP}}(\alpha) \cdot \mathrm{polylog}(n)$.

Proof. Most of the proof will focus on the first part of the theorem, i.e., that with high probability, all edges of length at least $r_{\texttt{EDP}}(\alpha)$ are pruned. The remaining parts then follow analogously to previous proofs. The proof of the second part is virtually identical to the proof of Lemma 4.3. The analysis of the (Recursive) Two-Ball Algorithm is also similar to the high-degree case, as long we we establish the independence between the pruned graph and the long edges: the edges of length exceeding $r_{\texttt{EDP}}(\alpha)$. The reason that this independence is sufficient is that the (Recursive) Two-Ball Algorithm only uses long edges, and its analysis can then omit any conditioning on the pruned graph.

To prove independence formally, let E_{sg} be a random edge set, and E the set of all its non-long edges (of length at most $r_{\texttt{EDP}}(\alpha)$). Let $\hat{E}_{\text{sg}}$ be another random edge set drawn from the same distribution whose non-long edges are also exactly E, while its long edges are generated independently from those of E_{sg}. With high probability, the (b, h)-EDP Pruning Algorithm will prune all long edges from both E_{sg} and $\hat{E}_{\text{sg}}$. By the observation preceding Theorem 7.2, this implies that $T(E_{\text{sg}}) = T(E)$ and $T(\hat{E}_{\text{sg}}) = T(E)$, so that the (b, h)-EDP Pruning Algorithm will produce the same pruned edge set on both graphs.

The remainder of the proof focuses on the first part of the theorem, i.e., the fact that with high probability, all long edges are pruned. The proof involves an intricate Deferred Decisions argument encapsulated in Lemma 7.3 below, which may be of interest in its own right.

Fix parameters (b, h) and a node pair (s, t), and let $r = \mathcal{D}(s, t) > r_{\texttt{EDP}}(\alpha)$. In applying Lemma 7.3, we consider the "universal set" U of all node pairs. Recall that the edge set $E = E_{\text{sg}}$ includes each node pair (u, v) independently with some probability $p_{(u,v)}$. The "feasible subsets" of U ("feasible paths") are all simple s-t paths of at most h hops. Any such path must contain at least one hop of length at least $\frac{r}{h}$; the corresponding edge is present with probability at most $q \triangleq C_{\text{sg}}\, k_{\text{sg}}\, (h/r)^d$. By Lemma 7.3, we obtain that for each $c \in \mathbb{N}$,

$$\pi_{s,t} \triangleq \text{Prob}\left[\, E_{\text{sg}} \text{ contains } b \text{ disjoint feasible paths} \,\right] \leq \text{Prob}\left[\, |E'| > c \,\right] + \tfrac{1}{1-cq}\,(cq)^b, \tag{13}$$

where E' is the set of all node pairs (u, v) such that $E_{\text{sg}} \cup \{(u, v)\}$ contains a feasible path.

The edge (s, t) is retained with probability at most $\pi_{s,t}$. Once we prove that $\pi_{s,t} = O(n^{-(2+\alpha)})$, we can complete the proof by taking the Union Bound over all n^2 node pairs (s, t). So it remains to upper-bound the right-hand side of (13) by $O(n^{-(2+\alpha)})$.

We first bound $\text{Prob}\left[\, |E'| > c \,\right]$ in (13). Let the random variable Δ denote the maximum degree of E_{sg}. Any node pair $(u, v) \in E'$ has the property that E_{sg} contains both an s-u path and a v-t path of length at most h hops each. Therefore, for fixed endpoints (s, t), there are at most Δ^h candidates for u and at most Δ^h candidates for v, and thus at most Δ^{2h} candidates for (u, v). We have thus proved that $|E'| \leq \Delta^{2h}$. Now, using Chernoff Bounds to upper-bound Δ, we have:

$$\text{Prob}\left[\, \Delta \geq \Theta(k_{\text{sg}} + \log \tfrac{n}{\delta}) \,\right] \leq \delta/n^2, \qquad \text{for all } \delta > 0.$$

Therefore $\text{Prob}\left[\, |E'| \geq c \,\right] \leq \delta/n^2$ for $c = (\Theta(k_{\text{sg}} + \log \frac{n}{\delta}))^{2h}$.

Substituting this choice of c into (13) and taking $\delta = n^{-\alpha}$, we obtain:

$$\pi_{s,t} = O(n^{-(2+\alpha)} + (cq)^b).$$

Finally, we show that $\pi_{s,t} = O(n^{-(2+\alpha)})$ by substituting $q = C_{\text{sg}}\, k_{\text{sg}}\, (h/r)^d$ and $r \geq r_{\texttt{EDP}}(\alpha)$. □

Lemma 7.3. *Consider a* universe set *U and a collection $\mathcal{F}$ of non-empty subsets of U called* feasible sets. *A random set $E \subseteq U$ is obtained by including each element $e \in U$ independently with probability p_e. The goal is to bound from above the number of disjoint feasible subsets of E.*

Fix $q \in [0, 1]$ such that each feasible set contains at least one element e with $p_e \leq q$. Let E' be the set of elements $e \in U$ such that $F \subseteq E \cup \{e\}$ for some feasible set F. Then, for each $b \in \mathbb{N}$,

$$\text{Prob}\left[\, E \textit{ contains } b \textit{ disjoint feasible sets} \,\right] \leq \min_{c \in \mathbb{N}} \left[\text{Prob}\left[\, |E'| > c \,\right] + \frac{1}{1 - cq}\,(cq)^b \right]. \tag{14}$$

Proof. An element $e \in U$ with $p_e \leq q$ is called a *witness*. Fix an arbitrary ordering ρ of U in which all non-witnesses precede all witnesses. For each feasible set $F \in \mathcal{F}$, the latest witness in F according to ρ is called a *canonical witness* for F. If furthermore $F \subseteq E$, then w is called *E-important*. Since each feasible set $F \subseteq E$ contains an E-important witness, from here on, we will focus on counting distinct E-important witnesses (rather than disjoint feasible sets $F \subseteq E$).

We reveal one by one whether elements of U are included in E, in the order of ρ. For each witness w, let E_w be the actual subset of E that is revealed *before* w is considered. Let us say that w is *ρ-important* if it is a canonical witness for some feasible set $F \subseteq E_w \cup \{w\}$. Then, w is E-important if and only if $w \in E$ and w is ρ-important. The latter two events, namely $\{w \text{ is } \rho\text{-important}\}$ and $\{w \in E\}$, are independent.

Let $w(t)$ be t^{th} ρ-important witness chosen in the above revelation process, $X_t = \mathbf{1}_{\{w(t) \in E\}}$, and let N be the total number of ρ-important witnesses. Then, $S_N \triangleq \sum_{t=1}^{N} X_t$ is the total number of E-important witnesses. Our goal is to bound S_N from above.

We accomplish this goal via Lemma 7.4 below. The sequence $\{X_t\}$ and the stopping time N satisfy the conditions in Lemma 7.4 (the upper bound). Specifically, we have established that $\mathbb{E}\left[X_t \mid N \geq t\right] = p_{w(t)} \leq q$, and the event $\{X_t = 1\}$ is independent of the past history given that $N \geq t$. By Lemma 7.4, we obtain that for all c,

$$\text{Prob}\left[\, S_N \geq b \,\right] \leq \text{Prob}\left[\, \texttt{Bin}_{c,q} \geq b \,\right] + \text{Prob}\left[\, N > c \,\right], \tag{15}$$

where $\texttt{Bin}_{c,q}$ is a random variable distributed according to the Binomial distribution with c samples and success probability q. We have $\text{Prob}\left[\, N > c \,\right] \leq \text{Prob}\left[\, |E'| > c \,\right]$, since each ρ-important witness is in E'. We complete the proof by noting that

$$\text{Prob}\left[\, \texttt{Bin}_{c,q} \geq b \,\right] \;=\; \sum_{l=b}^{c} \binom{c}{l} q^l (1-q)^{c-l} \;\leq\; \sum_{l=b}^{c} (cq)^l \;\leq\; \tfrac{1}{1-cq}\,(cq)^b. \qquad \square$$

Lemma 7.4 below is a technical lemma for analyzing a certain kind of "revelation process," in which a sequence of history-dependent 0-1 random variables is revealed, and the length of this sequence is also a history-dependent random variable. The lemma shows that whenever the expectation of each individual 0-1 random variable can be bounded, we can also bound the sum: we relate its distribution to the corresponding Binomial distribution. We will also use this lemma in the analysis of the adaptive algorithm in Section 7.2.

Lemma 7.4. *Consider a stochastic process $X_t \in \{0,1\}, t \in \mathbb{N}$ and a stopping time N on a filtration $\{\mathcal{F}_t : t \in \mathbb{N}\}$. Define $S_N \triangleq \sum_{t=1}^{N} X_t$. Assume that for some constants $p \leq q$ we have*

$$\mathbb{E}\left[X_t \mid N \geq t, F\right] \in [p,q] \quad \textit{for all } t \in \mathbb{N}, F \in \mathcal{F}_{t-1}.$$

Our goal is to bound the distribution of S_N in terms of the Binomial distribution.

Let $\texttt{Bin}_{t,p}$ be a random variable distributed according to the Binomial distribution with t samples and success probability p. Then, for all $x, t \in \mathbb{N}$, we have that

$$\text{Prob}\left[\, \texttt{Bin}_{t,p} \geq x \,\right] - \text{Prob}\left[\, N < t \,\right] \;\leq\; \text{Prob}\left[\, S_N \geq x \,\right] \;\leq\; \text{Prob}\left[\, \texttt{Bin}_{t,q} \geq x \,\right] + \text{Prob}\left[\, N < t \,\right]. \tag{16}$$

Proof. It suffices to prove the lower bound in (16); the upper bound is then derived from the lower bound applied to the stochastic process $\{1 - X_t \mid t \in \mathbb{N}\}$. Let $\{Y_t \mid t \in \mathbb{N}\}$ be a family of mutually independent 0-1 random variables with expectation p, and define

$$X_t^* = \begin{cases} X_t, & N \geq t \\ Y_t, & \text{otherwise.} \end{cases}$$

For each t, let $S_t = \sum_{s=1}^{t} X_s$, $S_t^* = \sum_{s=1}^{t} X_s^*$, and $\mathcal{F}_t^* = \sigma(X_1^*, \ldots, X_t^*)$. For each event $F \in \mathcal{F}_{t-1}^*$, we have that

$$\begin{aligned} &\mathbb{E}\left[X_t^* \mid F, N \geq t\right] = \mathbb{E}\left[X_t \mid F, N \geq t\right] \geq p, \\ &\mathbb{E}\left[X_t^* \mid F, N < t\right] = \mathbb{E}\left[Y_t \mid F\right] = p, \end{aligned}$$

which implies that

$$\mathbb{E}\left[X_t^* \mid F\right] \;=\; \mathbb{E}\left[X_t^* \mid F, N \geq t\right] \cdot \text{Prob}\left[\, N \geq t \mid F \,\right] + \mathbb{E}\left[X_t^* \mid F, N < t\right] \cdot \text{Prob}\left[\, N < t \mid F \,\right] \;\geq\; p.$$

By induction on t, it follows that $\text{Prob}\left[\, S_t^* \geq x \,\right] \geq \text{Prob}\left[\, \texttt{Bin}_{t,p} \geq x \,\right]$ for all $x, t \in \mathbb{N}$. Noting that $S_N \geq S_t = S_t^*$ whenever $N \geq t$, we obtain that

$$\begin{aligned} \text{Prob}\left[\, S_N \geq x \,\right] &\geq \text{Prob}\left[\, S_N \geq x \mid N \geq t \,\right] \cdot \text{Prob}\left[\, N \geq t \,\right] \\ &\geq \text{Prob}\left[\, S_t^* \geq x \mid N \geq t \,\right] \cdot \text{Prob}\left[\, N \geq t \,\right] \\ &= \text{Prob}\left[\, S_t^* \geq x \text{ and } N \geq t \,\right] \\ &\geq \text{Prob}\left[\, S_t^* \geq x \,\right] - \text{Prob}\left[\, N < t \,\right] \\ &\geq \text{Prob}\left[\, \texttt{Bin}_{t,p} \geq x \,\right] - \text{Prob}\left[\, N < t \,\right]. \end{aligned}$$

□

7.1.1 Running times in Theorem 7.2

While the main thrust in this paper is information-theoretic, the algorithms in Theorem 7.2 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 $\tilde{O}(n^2)$. We show how to reduce it to to $\tilde{O}(n)$.

Lemma 7.5. *The (b, h)-EDP Pruning Algorithm can be implemented in $\tilde{O}(n)$ time for constant b and h.*

Proof. We maintain a queue of edges to be checked, initially containing all edges of E_{sg}. In each step, one edge $e = (u, v)$ is removed from the queue and checked for pruning with respect to the current pruned graph E_{cur}. If E_{cur} does not contain the requisite b-tuple of edge-disjoint paths of length at most h, then e is pruned permanently. Otherwise, the b-tuple of paths provides a "certificate" for e. Later iterations may remove edges from this certificate; therefore, for each edge e' in the certificate, the algorithm stores a pointer that e' is part of the certificate for e. If e' is pruned at any point, then, following the pointers, the algorithm can determine all edges e whose certificates e' participates in. Upon pruning e', all such edges e are then re-enqueued and will need to be checked again for alternative certificates. Once the queue becomes empty, the algorithm terminates.

Without loss of generality, the target degree k_{sg} is $O(\log^2 n)$ (otherwise, the much more efficient Two-Hop Test from Section 4 would be used). By Chernoff Bounds, all node degrees are $O(\log^2 n)$ with high probability. Finding a certificate for a given edge using brute force then takes only $\text{polylog}(n)$ time. Moreover, for each edge e, there can be at most $\text{polylog}(n)$ edges whose certificates e participates in. No new edges are added to the queue if the current edge is not pruned, and at most $\text{polylog}(n)$ edges are added otherwise. Therefore, the running time is $\tilde{O}(n)$. □

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 [43]: 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.

7.2 Adapting to the "optimal" richness

Theorem 7.2 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 $r_{\texttt{EDP}}(\alpha)$ in Equation (12), subject to the constraint that E_{loc} is (b,h)-rich with small distortion. We show that such an automatic adaptation can be achieved if E_{loc} is "robust," in the sense defined below.

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 $r_{\texttt{EDP}}(\alpha)$ 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 prove that it works. That is, we need to identify a suitable "robustness property" of E_{loc} and argue that under this property, the chosen (b,h) pair is optimal. Let $T_{b,h}(E)$ denote the pruned graph if (b,h)-EDP Pruning Algorithm is applied to the edge set E. We rely on the following crucial observation:

Lemma 7.6. *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*

$$(2\delta h)^d \, C_{UD}^2 C_{sg} k_{sg} \leq \tfrac{1}{6}. \tag{17}$$

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

Remark. Since $C_{\text{sg}} = \Theta(1/\log n)$ and $C_{\text{UD}} = \Theta(1)$, condition (17) holds, for large enough n, whenever k_{sg}, δ and h are constants.

Lemma 7.6 is proved below. It naturally motivates the following definition of "robustness."

Definition 7.7. 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.[15]

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 Section 7.2.1.

Theorem 7.8. *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* (17) *holds. Then, when the Adaptive EDP algorithm is run with the candidate set H, it will obtain the guarantees of Theorem 7.2 for the optimum pair (b,h) among all $h \in H$.*

Proof. The Adaptive EDP algorithm picks the pair (b,h) with optimal $r_{\texttt{EDP}}(\alpha)$ among all pairs $(b,h), h \in H$ such that the pruned graph $T_{b,h}(E_{\text{sg}})$ is connected. By Lemma 7.6, with high probability, this is the set of all pairs $(b,h), h \in H$ such that the local structure E_{loc} is (b,h)-rich with distortion (σ,δ). □

[15] Note that any graph G in Definition 7.7 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), which in turn implies that it is a (σ,δ)-spanner.

Proof of Lemma 7.6. Fix (b,h) and let $T = T_{b,h}$. Let I be the set of ϵn isolated nodes in $T(E_{\text{loc}})$.

The high-level idea of the proof is as follows. For each node $u \in I$ and any edge $(u,v) \in E_{\text{loc}}$, the local structure E_{loc} alone does not contain b edge-disjoint paths of length at most h. Thus, for u not to be isolated in $T(E_{\text{sg}})$, a small neighborhood of u would have to be incident on at least one random edge. Because there are at least ϵn such isolated nodes u, we will be able to show that with high probability, at least one of them will end up isolated in $T(E_{\text{sg}})$. This is not trivial as there is significant dependence between the isolation events for different nodes; we solve this issue by considering a sufficiently spread-out subset $\mathcal{N}$ of I (which limits the dependence), and then applying Lemma 7.4 to a carefully designed revelation process. We now fill in the remaining technical details.

For any set $S \subseteq V$, let $\mathcal{E}(S)$ denote the event that E_{sg} contains no random edges incident on S. We begin by lower-bounding $\text{Prob}\,[\,\mathcal{E}(u)\,]$ for individual nodes u. Fix u and a distance scale r, and let U_r be the set of nodes v with $\mathcal{D}(u,v) \in (r, 2r]$. There are at most $C_{\text{UD}} \cdot (2r)^d$ nodes in U_r, and for each node $v \in U_r$, an edge (u,v) is created independently with probability at most $q \triangleq C_{\text{sg}}\, k_{\text{sg}}\, r^{-d}$. Thus, the probability that u has no edges to any nodes in U_r is at least

$$(1-q)^{|U_r|} = \left[(1-q)^{1/q}\right]^{2^d\, C_{\text{UD}} \cdot C_{\text{sg}}\, k_{\text{sg}}} \geq 4^{-2^d\, C_{\text{UD}} \cdot C_{\text{sg}}\, k_{\text{sg}}}.$$

Here, we used the fact that the function $f(q) = (1-q)^{1/q}$ is decreasing in q, so in particular $f(q) \geq f(\frac{1}{2}) = \frac{1}{4}$ for any $q \leq \frac{1}{2}$.

The event that u has no random edges is now the intersection of the events that u has no random edges at scale r, with r ranging over powers of 2. Thus, $\mathcal{E}(u)$ is the intersection of $\log(n)$ independent events, each with probability at least $4^{-2^d\, C_{\text{UD}} \cdot C_{\text{sg}}\, k_{\text{sg}}}$. Thus, for each node u,

$$\text{Prob}\,[\,\mathcal{E}(u)\,] \;\geq\; 4^{-2^d\, C_{\text{UD}} \cdot C_{\text{sg}}\, k_{\text{sg}}\, \log n} \;=\; n^{-2 \cdot 2^d\, C_{\text{UD}} \cdot C_{\text{sg}}\, k_{\text{sg}}}.$$

For any node $u \in I$, let V_u be the $(h-1)$-hop neighborhood of u in E_{loc}. Note that $V_u \subseteq B(u, \delta h)$, so it contains at most $C_{\text{UD}}\,(\delta h)^d$ nodes. We consider events $\mathcal{E}(V_u)$ that no node in V_u is incident on any random edges. The absence of any random edges incident on a subset of nodes V' can only increase the probability that no random edge is incident on a given node u, as there are fewer remaining candidate edges. In this sense, the events $\{\mathcal{E}(v) \mid v\}$ are positively correlated, and we can bound

$$\text{Prob}\,[\,\mathcal{E}(V_u)\,] \;=\; \text{Prob}\left[\,\textstyle\bigcap_{v \in V_u} \mathcal{E}(v)\,\right] \;\geq\; \textstyle\prod_{v \in V_u} \text{Prob}\,[\,\mathcal{E}(v)\,] \;\geq\; n^{-2 \cdot (2\delta h)^d\, C_{\text{UD}}^2 \cdot C_{\text{sg}}\, k_{\text{sg}}}. \qquad (18)$$

By the assumption (17), the above expression is at most $p \triangleq n^{-1/3}$.

We claim that whenever $\mathcal{E}(V_u)$ happens, the node $u \in I$ is isolated in $T(E_{\text{sg}})$. First, note that under the event $\mathcal{E}(V_u)$, u itself has no incident random edges. Let $(u,v) \in E_{\text{loc}}$ be arbitrary. We show that (u,v) must be pruned. Because no random edges are incident on V_u, no path in $T(E_{\text{sg}})$ of length at most h starting from u can contain any random edge. Thus, all u-v paths of length at most h in $T(E_{\text{sg}})$ must be entirely in E_{loc}. However, $(u,v) \notin T(E_{\text{loc}})$, so E_{loc} does not contain b edge-disjoint u-v paths of length at most h. Hence, $(u,v) \notin T(E_{\text{sg}})$.

It remains to show that with high probability, at least one of the events $\mathcal{E}(V_u), u \in I$ happens. To limit the dependence between the events under consideration, we focus on a subset $\mathcal{N} \subseteq I$. Let $\mathcal{N} \subseteq I$ be a $2Ch$-net for $(I, \mathcal{D})$.[16] Because there are at most $O((Ch)^d)$ nodes within distance $2Ch$

[16] Recall that an r-net for a metric space $(V, \mathcal{D})$ is a set of points $\mathcal{N} \subseteq V$ such that (i) any two points in $\mathcal{N}$ are at distance at least r from one another, and (ii) any point in V is within distance at most r from some point in $\mathcal{N}$. It is a well-known fact that such sets exist and can be constructed greedily by adding one point at a time.

of any node u, we obtain that $|\mathcal{N}| \geq \epsilon n / O((Ch)^d)$. Furthermore, because E_{loc} has distortion at most C, we get that $V_u \subseteq B(u, C(h-1))$, implying that the neighborhoods $V_u, u \in \mathcal{N}$ are pairwise disjoint.

The events $\mathcal{E}(V_u), u \in \mathcal{N}$ are still not independent, but their dependence is now more limited, making them amenable to the technique of Lemma 7.4. We define an ordering for revealing the presence (or absence) of edges (u, v), along with a revelation of the events $\mathcal{E}(V_u), u \in \mathcal{N}$. Fix some ordering φ on $\mathcal{N}$, and start with $R = \mathcal{N}$. R throughout will be a set of candidate nodes u such that the event $\mathcal{E}(V_u)$ has not been ruled out. In step $t = 1, 2, \ldots$, if $R \neq \emptyset$, let $u_t \in R$ be the first remaining node in R according to φ. Reveal the presence or absence of all random edges incident on V_{u_t} which have not been revealed yet. Whenever a random edge (v, v') is revealed to be present such that $v \in V_{u_t}, v' \in V_{u'}$ for some $u' \in R$, remove u' from R. (In this case, $\mathcal{E}(V_{u'})$ clearly cannot happen any more.) Once R is empty, reveal the presence or absence of all remaining random edges. Clearly, this is an equivalent way of revealing the random edge set E_{sg}.

Consider a particular step t, during which a node $u_t \in R$ is processed. If no edges incident on V_{u_t} are revealed, the event $\mathcal{E}(V_{u_t})$ has happened, and $T(E_{\text{sg}})$ will be disconnected. Conditioned on processing node u_t, the event $\mathcal{E}(V_{u_t})$ happens with probability at least $p = n^{-4/9}$, as the absence of some edges incident on V_{u_t} may already have been revealed earlier, whereas no edges can have been revealed as present. (Otherwise, u_t would have been removed from R.)

Let N be the number of steps t of the revelation process, and let X_t be the indicator variable of the event $\mathcal{E}(V_{u_t})$. Thus, whenever each $V_u, u \in \mathcal{N}$ has an incident random edge, we have that $\sum_{t=1}^{N} X_t = 0$. It thus suffices to upper-bound the probability that $\sum_{t=1}^{N} X_t = 0$, which can be accomplished using the lower bound of Lemma 7.4 with $x = 0$:

$$\text{Prob}\left[\sum_{t=1}^{N} X_t \geq 1\right] \geq (1 - (1-p)^t) - \text{Prob}\left[N < t\right], \quad \text{for all } t \in \mathbb{N},$$

or equivalently,

$$\text{Prob}\left[\sum_{t=1}^{N} X_t = 0\right] \leq (1-p)^t + \text{Prob}\left[N < t\right], \quad \text{for all } t \in \mathbb{N}. \tag{19}$$

We choose $t = \epsilon\sqrt{n}$. Then,

$$(1-p)^t \;\leq\; (1 - n^{-4/9})^{\epsilon\sqrt{n}} \;\leq\; e^{-\epsilon n^{1/18}},$$

so $(1-p)^t$ is exponentially small. Finally, we bound the probability that $N < \epsilon\sqrt{n}$. Consider a step t of the revelation process. With high probability, each node in V_{u_t} has at most $O(\log n)$ incident random edges, so that the total number of random edges incident on V_u is at most $O(k_{\text{loc}}^h \log n)$. Thus, with high probability, at most $O(k_{\text{loc}}^h \log n)$ other nodes u can be removed from R in any one step, implying that the process will take at least

$$\frac{|\mathcal{N}|}{O(k_{\text{loc}}^h \log n)} \;\geq\; \Omega\left(\frac{\epsilon n}{(Ch)^d k_{\text{loc}}^h \log n}\right) \;\geq\; \epsilon\sqrt{n}$$

steps, for sufficiently large n. In particular, $N \geq t$ with high probability, completing the proof. □

7.2.1 Examples of robust graphs

Recall that the toroidal grid is $(1, h)$-robust for any h. The grid example extends to graphs that are edge-transitive on a small scale, in the following sense.

Definition 7.9. Fix a graph G and a path length h. For any edge e, let H_e be the induced subgraph of the h-hop neighborhood of e in G. Two edges e, e' are *locally h-equivalent* if there exists an isomorphism $\phi_{e,e'} : H_e \to H_{e'}$ with $\phi(e) = e'$. G is called *edge-transitive* at scale h if any two edges are locally h-equivalent.

Notice that the traditional definition of edge-transitive graphs is obtained when h equals the graph's diameter.

Let G be an edge-transitive graph at scale h that is a (σ, δ)-spanner for $\mathcal{D}$. It is easy to see that G is $(1, h)$-robust with distortion (σ, δ). Indeed, the h-hop neighborhood of a given edge (u, v) determines whether this edge is (b, h)-connected (i.e., whether there exist b edge-disjoint u-v paths of at most h hops each). So for every given b, either every edge in G is (b, h)-connected, or every edge in G is not (b, h)-connected and therefore pruned by the (b, h)-EDP Pruning Algorithm.

We further generalize this example to graphs G with some short edges added. Specifically, pick an arbitrary node set $S \subseteq V$ such that its $(h+1)$-neighborhood in G contains at most $1 - \epsilon n$ nodes, for some $\epsilon \in (0, 1)$. Add arbitrary edges (u, v) such that $u, v \in S$ and $\mathcal{D}(u, v) \leq \delta$. Note that the resulting graph G' is also a (σ, δ)-spanner for $\mathcal{D}$.

We claim that G' is (ϵ, h)-robust with distortion (σ, δ). Indeed, if G is (b, h)-connected for some b then G' is (b, h)-rich with distortion (σ, δ) and connectivity witness G. Otherwise, no edge in G is (b, h)-connected in G alone. Consider the complement S' of the $(h + 1)$-neighborhood of S. Any edge e in G' with at least one endpoint in S' is also present in G, and moreover has the same h-neighborhood in both graphs. It follows that e is not (b, h)-connected in G'; consequently, it is pruned by the (b, h)-EDP Pruning Algorithm. Therefore every node in S' is isolated in $T_{b,h}(G')$.

8 Category Disjointness and Random Permutations

Recall that our motivation for the definition of the Local Category-Disjointness and Scale-R Category-Disjointness conditions was that they intuitively capture the notion of categories looking random with respect to one another "locally." In this section, we confirm the intuition guiding the definition, by showing that both conditions are satisfied with high probability when the metric space for each category i is randomly permuted, in the sense that $\mathcal{D}_i(u, v) = \mathcal{D}'_i(\sigma_i(u), \sigma_i(v))$ for some "base metric" $\mathcal{D}'_i$ and a random permutation σ_i on the node set. Accordingly, both conditions are indeed significantly weaker (in particular, more local) than requiring that metrics be randomly permuted.

Lemma 8.1. *Consider a multi-category social graph with near-uniform density. For each category i, let $\mathcal{D}'_i$ be a "base" metric, and σ_i a uniformly random permutation of the node set. (The permutations for different metrics are pairwise independent.) For each node pair (u, v), the category-i distance is $\mathcal{D}_i(u, v) = \mathcal{D}'_i(\sigma_i(u), \sigma_i(v))$. Then, with high probability, the Local Category-Disjointness and Scale-∞ Category-Disjointness conditions are satisfied.*[17]

Proof. Our proof uses an extension of Chernoff Bounds to dependent random variables in which the randomness comes from a random permutation (Theorem 8.2, stated and proved below).

We begin by proving that the Local Category-Disjointness condition is satisfied. Fix two categories $i \neq i'$. Consider balls B, B' of radii $r, r' = \text{polylog}(n)$ in categories i, i', respectively. Note that $\mathbb{E}[|B \cap B'|] = O((rr')^d/n) < 1$.

[17] Therefore Scale-R Category-Disjointness is satisfied for any R.

Define a mapping from category i to category j by $\sigma(u) \triangleq \sigma_j^{-1}(\sigma_i(u)) : V \to V$. $\sigma(u)$ captures at what point of the metric space $\mathcal{D}_j$ a node in the metric space $\mathcal{D}_i$ ends up. Because σ_i, σ_j were independent uniform permutations on V, so is σ. We will consider nodes $u \in B$, which we capture by setting $\alpha_u = \mathbf{1}_{\{u \in B\}}$. Such a node is also in B' iff $\sigma(u) \in B'$. Thus, defining $X_u = \mathbf{1}_{\{\sigma(u) \in B'\}}$, we get that $|B \cap B'| = \sum_{u \in V} \alpha_u X_u$, and by Theorem 8.2, this sum is at most $O(\log n)$ with high probability.

Next, we prove that the Scale-∞ Category-Disjointness condition holds as well. Fix a category j, distance scale $r > 0$, and two disjoint sets $B, B' \subseteq V, |B'| \geq |B|$ (which will be balls in category i). Define the random variable $f(B, B') \triangleq \sum_{v \in B, v' \in B'} \mathbf{1}_{\{\mathcal{D}_j(v,v') < r\}}$ to be the number of node pairs at category-j distance at most r.

We will prove a high-probability bound on $f(B, B')$ conditioned on the choice of all permutations σ_i for $i \neq j$. In other words, we consider the probability space induced by the random choice of $\sigma = \sigma_j$. We will prove that with high probability,

$$f(B, B') = O(r^d/n) \cdot |B|\,|B'| + O(\log^2 n). \tag{20}$$

Then the Scale-∞ Category-Disjointness condition follows by taking a Union Bound over all categories i, j, all pairs of balls B, B' in category i, and all distinct distances r in category j.

We begin by calculating the expectation of $f(B, B')$ using linearity of expectation. Notice that $\mathbb{E}\left[\mathbf{1}_{\{\mathcal{D}_j(v,v') < r\}}\right] = \text{Prob}\left[\,\mathcal{D}_j(v, v') < r\,\right]$ is the probability that v' is mapped to a node in a ball around v of radius r. Since there are $\Theta(r^d)$ nodes in the ball around v of radius r (wherever v itself is mapped), we get that $\mathbb{E}\left[\mathbf{1}_{\{\mathcal{D}_j(v,v') < r\}}\right] = \Theta(r^d/n)$, and $\mathbb{E}\left[f(B, B')\right] = \Theta(r^d/n) \cdot |B|\,|B'|$.

It remains to prove that $f(B, B')$ is concentrated around its expectation. Thereto, we will use Theorem 8.2 twice. First, focus on an arbitrary node v' and consider $f(B, \{v'\})$. We have that $\mathbb{E}\left[f(B, \{v'\})\right] = \Theta(r^d/n) \cdot |B|$. We can reveal the randomness of σ by first revealing $\sigma(v')$, which defines a set $U = \{u \in V \mid \mathcal{D}'_j(u, \sigma(v')) < r\}$. Then,

$$f(B, \{v'\}) = \textstyle\sum_{v \in V} \ \alpha_v \, \mathbf{1}_{\{\sigma(v) \in U\}},$$

where $\alpha_v = 1$ if $v \in B$, and $\alpha_v = 0$ otherwise. Thus, Theorem 8.2 implies concentration of $f(B, \{v'\})$ for any v', and gives us that with high probability, $f(B, \{v'\}) = O(\max(\log n, \frac{r^d}{n} \cdot |B|))$ for all v'. Let $N := \Theta(\max(\log n, \frac{r^d}{n} \cdot |B|))$ denote this high-probability bound.

Next, our goal is to sum over all $v' \in B'$. First, reveal $\sigma(v)$ for all $v \in B$, and condition on this choice, writing $T = \{\sigma(v) \mid v \in B\}$. Then, σ is defined by a uniformly random permutation from $V \setminus B$ to $V \setminus T$, or — equivalently — by a uniformly random permutation σ^{-1} from $V \setminus T$ to $V \setminus B$. For each $u' \in V \setminus T$, let $\beta_{u'} = \sum_{u \in T} \mathbf{1}_{\{\mathcal{D}'_j(u,u') < r\}}$ be the number of nearby locations to which nodes in B were mapped. Then, we can write

$$f(B, B') \;=\; \textstyle\sum_{u' \in V \setminus T} \ \beta_{u'} \, \mathbf{1}_{\{\sigma^{-1}(u') \in B'\}} \;=\; N \cdot \sum_{u' \in V \setminus T} \ \frac{\beta_{u'}}{N} \cdot \mathbf{1}_{\{\sigma^{-1}(u') \in B'\}}.$$

Defining $\alpha_{u'} = \min(1, \frac{\beta_{u'}}{N}) \in [0, 1]$, we get that with high probability (in the high-probability event that $f(B, \{v'\}) \leq N$ for all v'),

$$f(B, B') \leq N \cdot \textstyle\sum_{u' \in V \setminus T} \ \alpha_{u'} \, \mathbf{1}_{\{\sigma^{-1}(u') \in B'\}},$$

and σ^{-1} is a uniformly random permutation. By Theorem 8.2, with high probability,

$$\begin{aligned} f(B,B') \leq\ & N \cdot O(\textstyle\sum_{u' \in V \setminus T}\ \alpha_{u'}\ \mathbf{1}_{\{\sigma^{-1}(u') \in B'\}} + \log n) \\ =\ & O(\mathbb{E}\left[f(B,B')\right] + N \log n). \end{aligned}$$

If $N = \Theta(\log n)$, this bound is obviously $O(\mathbb{E}\left[f(B,B')\right] + \log^2 n)$. Otherwise, $N = \Theta(\frac{r^d}{n} \cdot |B|)$, and $\frac{r^d}{n} \cdot |B| = \Omega(\log n)$, which implies (because $r^d \leq n$) that $|B| = \Omega(\log n)$. And because we assumed that $|B'| \geq |B|$, we get that

$$\mathbb{E}\left[f(B,B')\right] \ =\ \Theta(r^d/n) \cdot |B|\,|B'| \ \geq\ \Theta(r^d/n) \cdot |B| \, \log n \ \geq\ \Theta(N \log n)$$

so that the $N \log n$ term is subsumed in the $\mathbb{E}\left[f(B,B')\right]$ term. This completes the proof of the lemma. □

Theorem 8.2 (Chernoff Bounds for permutations)**.** *Fix $n \in \mathbb{N}$ and a subset $I \subseteq \{1,\ldots,n\}$. Let σ be a uniformly random permutation of $\{1,\ldots,n\}$. For each $i \in \{1,\ldots,n\}$, fix $\alpha_i \in [0,1]$ and let $X_i = \mathbf{1}_{\{\sigma(i) \in I\}}$. Then $X = \sum_{i=1}^n \alpha_i X_i$ satisfies both conditions from Theorem 3.1:*

$$\begin{aligned} \text{Prob}\left[\,|X-\mu| > \delta\mu\,\right] &\leq \exp(-\mu\,\delta^2/3), && \textit{for any } \delta > 0 \\ \text{Prob}\left[\,X > (1+\delta)\mu'\,\right] &\leq \exp(-\mu'\,\delta^2/3), && \textit{for any } \delta \in (0,1). \end{aligned}$$

While the result appears standard, we are not aware of a published proof, so for completeness we provide a self-contained proof. The proof uses Chernoff Bounds for *negatively associated* random variables (see, e.g., [17]). We summarize the relevant result in the following theorem:

Theorem 8.3 ([17, pages 34–35 and Problem 3.1])**.** *Let $X_1,\ldots,X_n$ be random variables jointly distributed on $[0,1]^n$ such that $\sum_i X_i$ is a constant. For any subset $I \subseteq \{1,\ldots,n\}$, write $S_I \triangleq \sum_{i \in I} X_i$. Assume that the following hold for any such subset:*

- *Any X_i with $i \in I$ is conditionally independent of the X_j with $j \notin I$ given S_I.*
- *For any coordinate-wise non-decreasing function $f : \mathbb{R}^{|I|} \to \mathbb{R}$, the conditional expectation $\mathbb{E}\left[f(X_i,\, i \in I) \mid S_I = t\right]$ is non-decreasing as a function of $t \in \mathbb{R}$.*

Then, the random variables $X_1,\ldots,X_n$ are said to be negatively associated. *In particular, it follows that $X \triangleq \alpha_i\, X_i$ satisfies the bounds from Theorem 8.2, for any fixed $\alpha_1,\ldots,\alpha_n \in [0,1]$.*

Proof of Theorem 8.2. First note that by definition, $\sum_{i=1}^n X_i = |I|$ is a constant. Thus, it suffices to verify that the random variables X_i are negatively associated.

Fix $I \subseteq \{1,\ldots,n\}$. For each $t \in \mathbb{N}$, let $\mathcal{F}_t$ be the set of all tuples $(x_i,\, i \in I)$ such that $x_i \in \{0,1\}$ and $\sum_{i \in I} x_i = t$. Let U_t be the uniform distribution over $\mathcal{F}_t$.

To establish the first property of negative association, simply note that the conditional distribution of $(X_i,\, i \in I)$ given $S_I = t$ and any assignment for $(X_i,\, i \notin I)$ is U_t, so independence is established.

For the second property, fix a coordinate-wise non-decreasing function $f : \mathbb{R}^{|I|} \to \mathbb{R}$. Since the conditional distribution of $(X_i,\, i \in I)$ given $\{S_I = t\}$ is U_t, we have that

$$g(t) \triangleq \mathbb{E}\left[f(X_i,\, i \in I) \mid S_I = t\right] = \mathbb{E}_{\vec{x} \sim U_t}\left[f(\vec{x})\right].$$

We need to show that $g(t+1) \geq g(t)$. We couple selections according to U_t and U_{t+1} as follows.

- Pick $\vec{x} \sim U_t$.
- Pick j uniformly at random from $\{i \in I \mid x_i = 0\}$.
- Set $y_j = 1$, and $y_i = x_i$ for all $j \neq i$.

Notice that $\vec{y} \sim U_{t+1}$. By monotonicity of f, we have that $f(\vec{y}) \geq f(\vec{x})$. It follows that

$$g(t+1) = \mathbb{E}_{\vec{y} \sim U_{t+1}}\left[f(\vec{y})\right] \geq \mathbb{E}_{\vec{x} \sim U_t}\left[f(\vec{x})\right] = g(t).$$

The claim now follows from applying the result for negatively associated random variables. □

9 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 space with near uniform density, such as ultrametrics, more general "group structures" (e.g., [41]), 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] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2003.

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

[3] 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*, 2012. To appear. arXiv: 1112.1831.

[4] James Aspnes, Tolga Eren, David K. Goldenberg, A. Stephen Morse, Walter Whiteley, Yang Richard Yang, Brian D. O. Anderson, and Peter N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 12(5):1663–1678, 2006.

[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. I like her more than you: Self-determined communities. arXiv: 1201.4899, 2012.

[7] David Barbella, George Kachergis, David Liben-Nowell, Anna Sallstrom, and Ben Sowell. Depth of field and cautious-greedy routing in social networks. In *Proc. 18th Intl. Symp. on Algorithms and Computation*, pages 574–586, 2007.

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

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

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

[11] 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.

[12] Carter T. Butts. Models for generalized location systems. *Sociological Methodology*, 37(1):283–348, 2007.

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

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

[15] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. In *Proc. 23rd ACM SIGCOMM Conference*, pages 15–26, 2004.

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

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

[18] Philippe Duchon, Nicolas Hanusse, Emmanuelle Lebhar, and Nicolas Schabanel. Towards small world emergence. In *Proc. 19th ACM Symp. on Parallel Algorithms and Architectures*, pages 225–232, 2006.

[19] 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.

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

[21] 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.

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

[23] Pierre Fraigniaud and Cyril Gavoille. Polylogarithmic network navigability using compact metrics with small stretch. In *Proc. 21st ACM Symp. on Parallel Algorithms and Architectures*, pages 62–69, 2008.

[24] Pierre Fraigniaud, Cyril Gavoille, Adrian Kosowski, Emmanuelle Lebhar, and Zvi Lotker. Universal augmentation schemes for network navigability: Overcoming the $\sqrt{n}$-barrier. In *Proc. 20th ACM Symp. on Parallel Algorithms and Architectures*, pages 1–7, 2007.

[25] Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. In *Proc. 23rd ACM Symp. on Principles of Distributed Computing*, pages 169–178, 2004.

[26] Pierre Fraigniaud and George Giakkoupis. The effect of power-law degrees on the navigability of small worlds. In *Proc. 28th ACM Symp. on Principles of Distributed Computing*, pages 240–249, 2009.

[27] Pierre Fraigniaud and George Giakkoupis. On the searchability of small-world networks with arbitrary underlying sructure. In *Proc. 41st ACM Symp. on Theory of Computing*, pages 389–398, 2010.

[28] 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.

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

[30] George Giakkoupis and Nicolas Schabanel. Optimal path search in small worlds: Dimension matters. In *Proc. 42nd ACM Symp. on Theory of Computing*, pages 393–402, 2011.

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

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

[33] 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.

[34] 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.

[35] Piotr Indyk and Jiří Matoušek. Low-distortion embeddings of finite metric spaces. In *in Handbook of Discrete and Computational Geometry*, pages 177–196. CRC Press, 2004.

[36] David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 741–750, 2002.

[37] David Kempe, Jon Kleinberg, and Alan Demers. Spatial gossip and resource location protocols. *Journal of the ACM*, 51:943–967, 2005.

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

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

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

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

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

[43] Jon Kleinberg, Aleksandrs Slivkins, and Tom Wexler. Triangulation and embedding using small sets of beacons. *Journal of the ACM*, 56(6), 2009. Preliminary version appeared in Proc. 45th IEEE Symp. on Foundations of Computer Science.

[44] Robert Krauthgamer and James R. Lee. Navigating nets: simple algorithms for proximity search. In *Proc. 15th ACM-SIAM Symp. on Discrete Algorithms*, pages 798–807, 2004.

[45] 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.

[46] Ravi Kumar, David Liben-Nowell, and Andrew Tomkins. Navigating low-dimensional and hierarchical population networks. In *Proc. 14th European Symp. on Algorithms*, pages 480–491, 2006.

[47] 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.

[48] Emmanuelle Lebhar and Nicolas Schabanel. Close to optimal decentralized routing in long-range contact networks. *Theoretical Computer Science*, 348(2–3):294–310, 2005.

[49] Emmanuelle Lebhar and Nicolas Schabanel. Graph augmentation via metrics embedding. In *Proc. 12th Intl. Conf. on Principles of Distributed Systems (OPODIS)*, pages 217–225, 2008.

[50] David Liben-Nowell. Wayfinding in social networks. In *Algorithms for Next Generation Networks*, Computer Communications and Networks. Springer, 2010.

[51] 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.

[52] 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.

[53] 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.

[54] Gurmeet S. Manku, Moni Naor, and Udi Wieder. Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks. In *Proc. 35th ACM Symp. on Theory of Computing*, pages 54–63, 2004.

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

[56] 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.

[57] 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.

[58] 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.

[59] Diana Mok and Barry Wellman. Did distance matter before the Internet? Interpersonal contact and support in the 1970s. *Social Networks*, 29(3):430–461, 2007.

[60] T.S. Eugene Ng and Hui Zhang. Predicting internet network distance with coordinates-based approaches. In *Proc. 21st IEEE INFOCOM Conference*, 2002.

[61] 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.

[62] 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.

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

[64] 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.

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

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

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

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

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

[70] Aleksandrs Slivkins. Towards fast decentralized construction of locality-aware overlay networks. In *Proc. 26th ACM Symp. on Principles of Distributed Computing*, pages 89–98, 2007.

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

[72] Anthony Man-Cho So and Yinyu Ye. A semidefinite programming approach to tensegrity theory and realizability of graphs. In *Proc. 17th ACM-SIAM Symp. on Discrete Algorithms*, pages 766–775, 2006.

[73] 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.

[74] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proc. 35th ACM Symp. on Theory of Computing*, pages 281–290, 2004.

[75] 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.

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

[77] Duncan J. Watts and Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

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

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

[80] Bernard Wong, Aleksandrs Slivkins, and Emin Gün Sirer. Meridian: A lightweight network location service without virtual coordinates. In *Proc. 24th ACM SIGCOMM Conference*, pages 85–96, 2005.

[81] Zhisu Zhu, Anthony Man-Cho So, and Yinyu Ye. Universal rigidity and edge sparsification for sensor network localization. *SIAM Journal on Optimization*, 20(6):3059–3081, 2010.

[82] Uri Zwick. Exact and approximate distances in graphs — a survey. In *Proc. 9th European Symp. on Algorithms*, pages 33–48, 2001.