Massively-Parallel Similarity Join, Edge-Isoperimetry, and Distance Correlations on the Hypercube*

Paul Beame† Cyrus Rashtchian‡

Abstract

We study distributed protocols for finding all pairs of similar vectors in a large dataset. Our results pertain to a variety of discrete metrics, and we give concrete instantiations for Hamming distance. In particular, we give improved upper bounds on the overhead required for similarity defined by Hamming distance $r > 1$ and prove a lower bound showing qualitative optimality of the overhead required for similarity over any Hamming distance r. Our main conceptual contribution is a connection between similarity search algorithms and certain graph-theoretic quantities. For our upper bounds, we exhibit a general method for designing one-round protocols using edge-isoperimetric shapes in similarity graphs. For our lower bounds, we define a new combinatorial optimization problem, which can be stated in purely graph-theoretic terms yet also captures the core of the analysis in previous theoretical work on distributed similarity joins. As one of our main technical results, we prove new bounds on distance correlations in subsets of the Hamming cube.

1 Introduction

The task of reporting all pairs of similar items arises as a core computation in many applications. Known as a similarity join in the database community, this task generalizes a relational join [1, 33, 53]. Similarity joins aid in the removal of near-duplicate data [16, 56, 57] and the search for images with similar content [25, 39, 55]. In social media and other web applications, collaborative filtering methods employ similarity joins to find related users and items [21, 26, 38, 60].

We give new parallel algorithms and prove new communication lower bounds for computing a similarity join in a distributed, shared-nothing cluster. The similarity join problem is defined relative to a metric space (V, ρ). Given an input set S of points in V and a distance threshold r, the goal is to output all pairs x, y in S satisfying $\rho(x, y) \leq r$. For the parallel version of this problem, the input S starts equally partitioned among p processors. The challenge in designing an efficient parallel algorithm comes from outputting every close pair while balancing the workload among the processors and completing the overall similarity join quickly.

Our algorithms and lower bounds are for a *simultaneous* model, where the processors have one round to communicate input points before locally computing and outputting their portion of the set of similar pairs. This model is consistent with previous work [4, 12, 49, 59], and with the fact that each additional round of communication greatly increases the runtime in practice. We adopt the natural assumption that the local computation time increases proportionally to the maximum number of received vertices, and we focus on minimizing the latter quantity.

If there are p processors and n input points, an ideally balanced load with no replication of data would send precisely n/p points to each processor. However, since every similar *pair* must end up at some processor, some replication is required, and it will not be possible to achieve this ideal. We measure the *overhead* of an algorithm as the ratio of the maximum number of points any processor receives to the n/p ideal[1]. To avoid large overhead in the worst case, any adequate algorithm must employ a randomized load balancing strategy.

We exhibit new algorithms that improve on the overhead of the best previous similarity join algorithms from [1, 2, 49] for Hamming distance on $\{0,1\}^d$ with distance thresholds $r > 1$. Our improvements are both

*Supported in part by NSF grants CCF-1217099 and CCF-1524246

†Computer Science & Engineering, University of Washington, Seattle, WA, USA, beame@cs.washington.edu

‡Computer Science & Engineering, University of Washington, Seattle, WA, USA, cyrash@cs.washington.edu

[1]Overhead resembles the replication rate used in analyzing MapReduce algorithms, but it does not depend on the notion of reducer size [49], and thus it applies more generally than to just MapReduce algorithms.

quantitative in terms of the overhead bound and more general in terms of the range of set sizes n for which they work. More fundamentally, we show that every randomized similarity join algorithm for Hamming distance r on $\{0,1\}^d$, even one that only weakly approximates the similarity join, requires overhead $d^{\Omega(r)}$. This qualitatively matches the overhead of the best algorithms. Moreover, ours is the first lower bound for similarity join that applies for any $r \neq 1$, for any set size that is $o(2^d)$, or for any algorithm that is only approximately correct.

We develop our improved similarity join algorithm as well as our lower bound by analyzing certain graph-theoretic quantities of a similarity graph. For distance threshold r on $\{0,1\}^d$, the Hamming similarity graph has edges between all pairs of points with Hamming distance at most r. To approximate the similarity join, both endpoints of most edges of this graph must arrive together at some processor. This criterion has a natural interpretation in graph terms. Let $A_1, \ldots, A_p$ be the sets of points that are sent to each of the p processors. Then, the collection of p subgraphs induced by these sets must contain many of the edges in the input set. Since the input consists of an arbitrary set of points, it follows that randomized algorithms correspond to distributions $\mathcal{A}$ on approximate coverings $(A_1, \ldots, A_p)$ of the edges of the similarity graph. We refer to such a p-tuple as an *edge-covering*.

The design of edge-coverings connects the similarity join problem with the edge-isoperimetric question in graph theory. To see this connection, notice two facts. First, our overhead measure is minimized by reducing the number of input points contained by each set A_i. Second, an edge-covering is improved if each set A_i contains many edges relative to its size. *Edge-isoperimetric sets* in a graph are sets of vertices of a given size that maximize the number of induced edges [14]. Therefore, we observe that it is good to build each set A_i as a union of randomly chosen edge-isoperimetric sets of suitable size in the similarity graph. In particular, we choose that size to be roughly n/p, the ideal number of input elements sent to a processor. This ensures bounded overhead and avoids imbalanced correlations of the input set with the edge-isoperimetric sets. In fact, this class of algorithms is a generic one that is suitable for use with any edge-transitive similarity graph, such as the graphs defined by the Euclidean or Manhattan distance.

In the case of Hamming distance $r = 1$ on $\{0,1\}^d$, the optimal edge-isoperimetric sets are near-subcubes [13, 29, 30, 40]. It is worth noting that random subcubes are a feature of the best similarity join algorithm for this case [1, 49]. However, for $r > 1$, previous algorithms use other methods [1, 2, 28, 44, 49, 56], and good edge-isoperimetric sets were not identified as suitable constructs. Though the optimal edge-isoperimetric sets for $r > 1$ are not known, in our improved similarity join algorithm we use Hamming balls (which interestingly are vertex-isoperimetric sets for $r = 1$) and show that they are good approximations to optimal edge-isoperimetric sets.

Our general lower bound applies much more broadly than this edge-isoperimetric set construction. It holds for any randomized way of choosing sets $(A_1, \ldots, A_p)$. We show that edge-covering and similarity join are hard even if the input set of points S is itself a random, sub-sampled Hamming ball.

The general intuition for our lower bound proof is that either the total number of elements among the A_i is large, in which case large overhead follows immediately, or the sets A_i have a large density of pairs of points in $\{0,1\}^d$ of distance $\leq r$. For this second case we derive our lower bound on the overhead by proving a new connection between the density of different Hamming distances of subsets on $\{0,1\}^d$. We show that for $A \subseteq \{0,1\}^d$ that if there is a sufficiently high density of pairs at Hamming distance r in A, then there is also a high density of pairs at much larger Hamming distances. This property is likely to be of independent interest since it can be viewed as a Sidorenko-type result [20, 24, 50] for shortest paths in the r-th power of the hypercube.

Related Work. We review two baseline algorithms for the similarity join problem under any metric. The first is a randomized join algorithm and has overhead $O(\sqrt{p})$. We discuss the details of this universal all-pairs algorithm in Section 2. The second baseline (known as the "Ball-hashing-2" algorithm in [1]) works by randomly partitioning the points z of the metric space among the processors and sending each input point x to the processors associated with all z at distance at most $\lceil r/2 \rceil$ from x. In other words, processors are responsible for unions of balls of radius $\lceil r/2 \rceil$. The expected overhead of this algorithm is the size of balls of radius $\lceil r/2 \rceil$ in the metric space, which in the case of the Hamming distance on $\{0,1\}^d$ we denote by $B(d, \lceil r/2 \rceil)$ and is $O(d^{\lceil r/2 \rceil})$.

Recent work [1, 2, 49] demonstrates improvements over the baseline algorithms for $\{0,1\}^d$ with Hamming distance. For distance $r = 1$, Afrati *et al.* [1] present

an algorithm (called "Splitting") based on a fixed grid of subcubes. They analyze how subcube dimensionality affects the overall communication. We observe that choosing the dimension to be $\lfloor \log_2(n/p) \rfloor$ leads to an edge-covering that achieves overhead $O(d/\log(n/p))$. For $r > 1$, in [1, 49] they also provide a variety of related algorithms. They analyze subcubes for larger distances, but this is not as effective as the Ball-hashing-2 algorithm in terms of overall communication. Their Ball-hashing-1 algorithm sends input points to processors based on balls of radius r centered at input points. However, the Ball-hashing-1 communication scales with $B(d,r)$, worse than the Ball-hashing-2 algorithm.

The best previous algorithm for $r > 1$ is the AnchorPoints from [2], which improves upon [1]. The basic building block of their algorithm is a *covering code* $\mathcal{C}$ of radius r, defined as a set of points $\mathcal{C}$ in $\{0,1\}^d$ with the property that every $x \in \{0,1\}^d$ is within distance r of some element of $\mathcal{C}$. After randomly partitioning the elements z of the code $\mathcal{C}$ among the processors, the algorithm sends each input point x to every processor associated with an element of the covering code that is at distance at most $\lceil 3r/2 \rceil$ from x. In other words, processors are responsible for unions of balls of radius $\lceil 3r/2 \rceil$. By using covering codes of density at most $O(1/B(d,r))$, the work in [2] improves on the overhead of the Ball-hashing-2 algorithm by roughly a $2^{\lceil r/2 \rceil}$ factor. By basing our construction on nearly edge-isoperimetric sets and larger Hamming balls, our algorithm improves the previous bound by factor of roughly $(\frac{1}{2}\log_d(n/p))^{\lceil r/2 \rceil}$.

Similarity search algorithms often employ Locality Sensitive Hashing (LSH) [17, 28, 36]. An LSH family is any distribution over hash functions such that near points map to the same bucket with a higher probability than far points. For Hamming space, an example LSH family picks a subset of k coordinates in $\{1,2,\ldots,d\}$ and maps a vector to its k-bit projection on these coordinates [28]. LSH may be used as part of a similarity join algorithm by simply hashing the input points, checking all pairs of points within the same hash bucket, and outputting the close pairs. The success and approximation of such an algorithm depends on k and is amplified through independent trials.

The randomness in classical LSH schemes [17, 28, 36] leads to the possibility of both false positives and false negatives, but recent work remedies things for false negatives [44, 48]. Their LSH family for Hamming distance covers all close pairs with a correlated family of subcubes.

In a MapReduce setting, LSH buckets correspond to groups of points sent together. Hashing a vector in $\{0,1\}^d$ by choosing k coordinates corresponds to partitioning $\{0,1\}^d$ into subcubes of dimension k and determining the subcube to which the vector belongs. Achieving a balanced load requires that each hash bucket contain a bounded number of vectors. In contrast, all-pairs similarity search algorithms based on LSH ignore the sizes of each bucket. Instead they focus on the total work done by the algorithm, and, over many rounds, maintain a counter for each vector and simply abort after comparing it to too many far vectors [28, 45]. In fact, as we show, the buckets maintained by LSH algorithms for distances $r > 1$ do not lead to the best load balance.

The approximate similarity join problem we study requires a sharp distance threshold – only producing pairs with distance at most r; false positives can be easily filtered before being output. Therefore, our notion of approximation only refers to the fraction of false negatives. This differs from problems such as Approximate Near Neighbor (ANN) search in terms of the notion of approximation [27], where ANN benefits from allowing false positive pairs with distance $\leq cr$, where $c > 1$ determines the overall space usage and running time.

There are well-known lower bounds for LSH, such as [10, 41, 42], but these only apply for $r = \Omega(d)$. Indeed, they employ analytic techniques, such as the Bonami-Beckner inequality, that do not capture the small distances that are the focus of our work.

Panigrahy, Talwar, and Wieder [46, 47] prove cell-probe lower bounds for randomized algorithms solving the approximate near neighbor search problem (recent improvements appear in [9]). Their approach resembles ours in that they consider (a generalization of) the edge expansion of the similarity graph for a metric space (i.e., the graph with edges connecting every pair of points with distance at most r), though the specific application and techniques are different.

Recently, there has been work on proving strong conditional lower bounds (assuming the strong exponential time hypothesis [32]) on the time complexity of computing all-pairs (approximate) nearest neighbors [3, 5, 6]. However these bounds do not apply to thresholded similarity join or to our model, which measures the communication required for a single round.

There are also other kinds of algorithms for similarity search. In particular, there is a substantial body

of work on the design and analysis of data structures that first preprocess the data and then answer nearest-neighbor and range queries. Most of the analysis considers time complexity and space usage for serial algorithms. Therefore, the majority of these results are orthogonal to the study of distributed similarity joins and edge coverings (cf. [27]).

For metrics other than Hamming distance, objects resembling edge-coverings underlie state-of-the-art algorithms for similarity joins on datasets of real vectors: Working in Euclidean space, Aiger, Kaplan, and Sharir [4] demonstrate an efficient edge-covering using randomly shifted and rotated grids. In the ℓ_∞ metric, Lenhof and Smid [37] provide serial and shared-memory algorithms using a data-dependent partition of space into ℓ_∞ hypercubes. For angular distance, LSH schemes use random half-spaces or spherical caps as randomized edge-coverings [8, 17, 28, 51, 54].

Many authors discovered the optimal edge-isoperimetric shape for the standard (distance one) hypercube [13, 29, 30, 40]. Bezrukov's survey [14] contains a thorough, modern treatment. Kahn, Kalai, and Linial [34] consider the generalization for r greater than one, but leave asymptotic results for all set sizes as a open question. Bollobás and Leader [15] prove a generalization of the hypercube result for both Hamming and ℓ_1 distance on vectors over a larger alphabet (only for distance one in both cases). Our edge-covering method implies that the Bollobás and Leader result can be used to design protocols for this space.

Finally, in the combinatorics community, researchers have studied optimal coverings for pairs of elements in designs [18, 31]. This relates to theoretical algorithms for parallel joins [12]. We exploit the additional structure of the metric space to design vastly more efficient protocols and algorithms than those for all-pairs joins when the number of processors is very large.

2 Similarity Graphs, Edge-Coverings, and Overhead

Similarity graphs. Any undirected graph $G = (V, E)$ can be viewed as defining a notion of similarity on V in which similar pairs are joined by an edge. We study *edge-transitive* graphs, those having automorphisms that make every neighborhood look like any other. We particularly focus on the case that V is the set $\{0,1\}^d$ equipped with the Hamming distance $\mathsf{dist}(u, v)$ that equals the number of bits differing in $u, v \in \{0,1\}^d$. For a distance threshold $r \in [d]$, let $\Gamma_{\le r}$ denote the Hamming similarity graph with vertices $\{0,1\}^d$ and edges connecting u, v whenever $\mathsf{dist}(u, v) \le r$, with self-loop edges for $u = v$. This regular graph has degree $B(d, r) \triangleq \binom{d}{r} + \binom{d}{r-1} + \ldots + d + 1$. For a subset A in $\{0,1\}^d$, let $\mathsf{E}_{\le r}(A)$ denote the set of edges in $\Gamma_{\le r}$ with both endpoints in A. Similarly, define Γ_r as the graph with edges connecting vertices with distance exactly r, and define $\mathsf{E}_r(A)$ as the set of edges in Γ_r with both endpoints in A. It will be convenient to define notation for the average density of edges in the induced subgraphs $\Gamma_{\le r}[A]$ and $\Gamma_r[A]$, so we define $\mathsf{e}_{\le r}(A) \triangleq \frac{|\mathsf{E}_{\le r}(A)|}{|A|}$ and $\mathsf{e}_r(A) \triangleq \frac{|\mathsf{E}_r(A)|}{|A|}$.

Similarity Join Algorithms and Overhead. We study a broad, natural class of one-round, p-processor similarity join algorithms. Each point x in the input set S is sent to some non-empty subset of processors $P(x) \subseteq [p]$. The algorithm then uses local operations at each destination processor to report all similar pairs among the elements it receives from S. We allow the mapping P to depend on the (approximate) size n of the set S but not S itself. The goal is to minimize the maximum number of elements received by any processor, while ensuring that (almost) every similar pair in S goes to at least one processor.

The ideal load balancing would be a load of n/p elements per processor, but we show that this is not always possible. Randomization is essential in defining P, since for any deterministic mapping P any set $S \subseteq P^{-1}(i)$ for any $i \in [p]$ will yield a load of $|S|$. It is convenient to use an equivalent and convenient perspective by considering the distribution $(P^{-1}(1), \ldots, P^{-1}(p))$ of tuples of sets produced by the algorithm. This emphasizes the combinatorial aspects of the problem, and is formalized in the following definition in terms of randomized edge-coverings.

Definition 2.1. *A* randomized (p, δ)-edge covering *for the n-subsets of $\Gamma_{\le r}$ is a distribution $\mathcal{A}$ on p-tuples $(A_1, \ldots, A_p)$ of subsets of $\{0,1\}^d$ such that for every subset $S \subseteq \{0,1\}^d$ of size n,*

$$\mathop{\mathbb{E}}_{(A_1,\ldots,A_p)\sim\mathcal{A}} \left[\left| \bigcup_i \mathsf{E}_{\le r}(A_i \cap S) \right| \right] \ge \delta \cdot |\mathsf{E}_{\le r}(S)|.$$

There is an analogous definition of a (p, δ)-edge covering for n-subsets of Γ_r with E_r replacing $\mathsf{E}_{\le r}$.

As a way to capture the maximum load of randomized algorithms, our primary complexity measure will

be the expectation of the maximum ratio of $|A_i \cap S|$ to $|S|/p$.

Definition 2.2. *Let $\mathcal{A}$ be a distribution on p-tuples $(A_1, \ldots, A_p)$ of subsets of $\{0,1\}^d$. Let $S \subseteq \{0,1\}^d$ with $p \leq |S|$. Define the* overhead *of $\mathcal{A}$ on set S,*

$$\mathsf{overhead}(\mathcal{A}, S) = \mathop{\mathbb{E}}_{(A_1,\ldots,A_p)\sim\mathcal{A}} \left[\max_{i\in[p]} |A_i \cap S| \cdot \frac{p}{|S|} \right],$$

*where the expectation is over the randomness of $\mathcal{A}$. The n-*overhead *of $\mathcal{A}$ is defined as*

$$\mathsf{overhead}_n(\mathcal{A}) = \max_{S\subseteq\{0,1\}^d,\, |S|=n} \mathsf{overhead}(\mathcal{A}, S)$$

Note that defining the overhead in terms of the *maximum* value of $|A_i \cap S|$ is critical, since a trivial algorithm that sends S entirely to a random processor achieves average load exactly $|S|/p$.

Any randomized (p, δ)-edge covering $\mathcal{A}$ yields a natural one-round, p-processor, data-parallel algorithm for computing a δ-approximate similarity join. First, choose $(A_1, \ldots, A_p) \sim \mathcal{A}$ using shared randomness. Then, send x to processor i whenever $x \in A_i$. In other words, on input S, processor i receives $A_i \cap S$ for $i \in [p]$. Over the random choices of the algorithm, this strategy outputs in expectation at least a δ fraction of all pairs in S with Hamming distance at most r.

These *local* algorithms capture a very broad class of natural algorithms for similarity joins including the ones discussed in the introduction. For example, in the ball-hashing-2 algorithm, the sets $A_1, \ldots, A_p$ are unions over a random partition of Hamming balls of radius $\lceil r/2 \rceil$.

A Universal All-pairs Algorithm. As another example of such algorithms we present the other baseline load-balancing algorithm mentioned in the introduction which works for any notion of similarity and yields overhead $O(\sqrt{p})$.

Proposition 2.3. *There is a randomized 1-round local algorithm for any similarity measure that has expected per-processor load $O(|S|/\sqrt{p})$ on all sets S with $p \leq |S|$.*

Proof. Assume without loss of generality that $p = \binom{q}{2}$ for integer q. Choose a random mapping $h : \{0,1\}^d \to [q]$. Now send $x \in \{0,1\}^d$ to processors indexed by the pair $\{h(x), i\}$ for every $i \in [q]$ not equal to $h(x)$. Every pair of inputs in S will be seen by some processor. The per-processor load is almost surely $O(|S|/q)$. □

This yields a randomized $(p, 1)$-edge-covering of $\{0,1\}^d$ with overhead $O(\sqrt{p})$ for all set sizes. In Appendix B we note that for very small sets S, this bound essentially cannot be improved, even on $\Gamma_{\leq r}$. For a limited number of available processors, this algorithm is still useful in practice, even for large datasets. However, many important cases involve sets S that are a much larger portion of $\{0,1\}^d$. In fact, some previous work on similarity join on the binary cube has often focused on the case that the input set size is completely at the other extreme – a constant fraction of all possible vectors. We consider a wide range of input sizes and not just these extremes.

Definition 2.4 (Main Complexity Measure). *Let $f_{\leq r}(n, d, p, \delta)$ be the minimum of $\mathsf{overhead}_n(\mathcal{A})$ over all randomized (p, δ)-edge coverings for n-subsets of $\Gamma_{\leq r}$. Analogously define $f_r(n, d, p, \delta)$ for Γ_r.*

Lower bounds on $f_{\leq r}(n, d, p, \delta)$ are lower bounds on the amount of overhead required for any randomized p-processor 1-round local algorithm that approximately solves similarity join on the binary cube with Hamming distance threshold r for subsets $S \subseteq \{0,1\}^d$ of size n. Upper bounds on $f_{\leq r}(n, d, p, \delta)$ and $f_r(n, d, p, \delta)$ will be particularly interesting when $\delta = 1$, which implies no error, or when δ is sufficiently close to 1, which implies the probability of missing any similar pair in S is extremely small.

Comparison to MapReduce Measures. In MapReduce algorithms, there is a collection of individual tasks called *reducers* that are randomly assigned to one of the p processors. Previous similarity join algorithms (as well as our edge-isoperimetric algorithm) define reducers for each subset $R_1, \ldots, R_K$ of $\{0,1\}^d$ for some $K \gg p$. Then, during a shuffling phase, the sets $A_1, \ldots, A_p$ are formed randomly as unions of some of the $\{R_j\}$. The key complexity measure analyzed in the previous work [1, 2, 49] is the *replication rate*, defined as the average number of reducers to which each point is sent. We claim that overhead is a better measure.

The replication rate is a useful measure only when the reducers are small enough that they can be mapped randomly and uniformly to yield tasks that are nicely load-balanced across the processors. Our measure, overhead, captures a much wider class of algorithms than replication rate can. Moreover, in the regime where replication rate is a useful measure, overhead essentially equals replication rate. The main advantage of overhead concerns very large reducers. For example, if one reducer was responsible for essentially the

whole input, the replication rate would be one, but this algorithm would behave badly in practice and has the worst possible overhead of p. Therefore, overhead both captures the lower bounds using replication rate and defines a measure of load balancing for large reducers.

3 Our Results

Theorem 3.1. *Let $0 < \delta \leq 1$ and r, d, n, p be such that $r \leq d$, $p \leq n \leq 2^d$, and $\gamma = \log_n p$.*

(a) For all integers k with $\lceil r/2 \rceil \leq k \leq (1-\gamma)\log_d n$ there is a MapReduce algorithm witnessing the bound

$$f_{\leq r}(n,d,p,\delta) \leq \max\left\{6 \cdot 2^{\lceil r/2 \rceil} \ln\left(\frac{1}{1-\delta}\right) \cdot \left(\frac{d}{k}\right)^{\lceil r/2 \rceil},\ 9\log_2 p\right\}$$

The algorithm has reducer size $B(d,k)$, and with probability $1-2^{-d}$, it maps each input to at most

$$7 \cdot \max\left\{d \ln 2,\ \ 6 \cdot 2^{\lceil r/2 \rceil} \ln\left(\frac{1}{1-\delta}\right) \cdot \left(\frac{d}{k}\right)^{\lceil r/2 \rceil}\right\}$$

processors. Moreover, with $\delta = 1 - 1/n^3$, the edge-covering of the input is error-free with probability at least $1 - 1/n$.

(b) There is constant $c_\delta > 0$ such that for every r, n, d, p, δ with $r \leq \sqrt{d/2}$, $n \geq d^{r \log_2 d}$, and $\delta \geq 4/\sqrt{d}$,

$$f_{\leq r}(n,d,p,\delta) \geq (c_\delta/r)^r \cdot (d/\log_d^2 n)^{\gamma r/2}.$$

Further, if $\gamma r \leq 1$ then

$$f_{\leq r}(n,d,p,\delta) \geq (c_\delta/r)^r \cdot (d/\log_d n)^{\gamma r}.$$

(c) The bounds given in (a) and (b) also hold with $f_{\leq r}(n,d,p,\delta)$ replaced by $f_r(n,d,p,\delta)$.

Observe that our lower bound in (b) (in its range of applicability) is qualitatively very similar to the upper bound in (a) with maximal choice of k, especially as the number of processors approaches the set size. The only previous lower bound on replication rate or overhead for similarity join [49] used the edge-isoperimetric bound for $r = 1$ to derive a lower bound of $(\log_2 n)/\log_2 q$ where q is an upper bound on the reducer size for input sets of size n that are dense subsets of a subcubes; however, for q as large as n/p, the regime in which our bounds apply, this only yields a nearly trivial lower bound of $1/(1-\gamma)$.

We outline the upper bound improvements compared to previous algorithms [1, 49, 2]. Our upper bound with maximal k comes from an algorithm with overhead $O(2d/\log_m(n/p))^{\lceil r/2 \rceil}$.

This improves on the Ball-hashing-2 algorithm [1], which has overhead $(2d/r)^{\lceil r/2 \rceil}$ using reducers of size $B(d, \lceil r/2 \rceil)$. Our algorithm also improves over the AnchorPoints algorithm in [2], which has overhead $O(B(d, \lceil 3r/2 \rceil)/B(d,r))$, approximately $(d/r)^{\lceil r/2 \rceil}$, using reducers of size $B(d, \lceil 3r/2 \rceil)$. We finally note that the subcube algorithm [1, 49] has good overhead when $r = 1$ and is not improved by our results, but it is not competitive for $r > 1$.

Our algorithm is best for large k with reducers of size $B(d,k) \approx n/p$. We improve the overhead by up to a $(\frac{1}{2}\log_d(n/p))^{\lceil r/2 \rceil}$ factor, taking advantage of larger reducers to obtain better bounds. For a comparison using a natural range of parameters, consider $d = (\log_2 \sqrt{n})^2$ and $p = 2^{\sqrt{d}} = \sqrt{n}$. Our overhead for even r scales like $(d \log_2 d)^{r/4}$, which improves over the best previous algorithm by a factor of roughly $(d/r^2 \log_2 d)^{r/4}$.

Our algorithm is nearly-optimal for $n = 2^{\Theta(d)}$ and $p = n^\gamma$ for constant $\gamma < 1$. In this case, choosing $\delta = 1 - 1/2^{3d}$ we obtain an algorithm that is almost certainly correct and has overhead only $2^{O(r)} d \log^{\lceil r/2 \rceil} d$, since $\log_2 p \leq d$. This is a significant improvement over the best previous algorithm [2] which has overhead and replication rate $(d/r)^{\lceil r/2 \rceil}$ in this regime.

4 Randomized Edge-Coverings using Edge-Isoperimetric Sets

In this section, we describe our randomized edge-covering for $\Gamma_{\leq r}$ with overhead achieving the upper bound in Theorem 3.1. This randomized edge-covering defines a one-round, data-parallel protocol for reporting all pairs in $S \subseteq \{0,1\}^d$ with Hamming distance at most r. It thus also provides a MapReduce algorithm for Hamming similarity joins.

We begin with our general protocol for arbitrary edge-transitive similarity graphs, since it is easy to state and explains the intuition. We then specialize this general protocol to Hamming distance and provide the specific analysis in this case.

General Edge-Transitive Graph Covering. We build a randomized (p,δ)-edge-covering for G using unions of random translates of a fixed set $U^* \subseteq V$. An *edge-isoperimetric set* of size s is any set $U^*(s)$ of vertices in G that maximizes the number of edges among all induced subgraphs with s vertices. We will use translates of $U^* \triangleq U^*(s)$ for $s \leq n/p$. Let π be a graph automorphism on G. Define $\pi(U^*)$ to be the set of vertices $\pi(x)$ for each $x \in U^*$. By edge-transitivity, $\pi(U^*)$ contains

the same number of edges as U^* does.

To construct the (p,δ)-edge covering, we choose a parameter L and choose Lp automorphisms $\pi_1,\ldots,\pi_{Lp}$ of G uniformly at random, and define

$$A_i = \bigcup_{j \equiv i \bmod p} \pi_j(U^*).$$

With Lp large enough compared to the ratio of the number of edges in G to that of U^*, the sets A_i cover a δ-fraction of edges in S with high probability. As we show, the fact that $|U^*| \le n/p$ ensures a bounded overhead via Bernstein's concentration inequality.

When interpreted as a one-round MapReduce algorithm, the construction above admits good bounds on both the maximum reducer size and the maximum number of times a vertex is replicated by the mappers.

Lemma 4.1. *Let U^* be a subset of vertices in $G=(V,E)$. The MapReduce algorithm that chooses Lp uniformly random automorphisms $\pi_1,\ldots,\pi_{Lp}$ and maps $x \in V$ to the reducer assigned $\pi_j(U^*)$ whenever $x \in \pi_j(U^*)$ has reducer size $|U^*|$, and with probability $1-|V|^{-1}$, it maps each vertex to at most $7\cdot\max\{\ln|V|,\ Lp|U^*|/|V|\}$ reducers.*

Proof. Any $x \in V$ is contained in $\pi_j(U^*)$ with probability $|U^*|/|V|$. The expected number of indices j such that $x \in \pi_j(U^*)$ is $Lp|U^*|/|V|$. By independence of the automorphisms, a standard Chernoff bound implies that x is mapped to more than $7\cdot\max\{\ln|V|,\ Lp|U^*|/|V|\}$ reducers with probability at most $|V|^{-2}$. Taking a union bound over V finishes the proof. □

We now analyze the above algorithm for $\Gamma_{\le r}$.

Edge-Isoperimetry for Hamming Distance. The classical edge-isoperimetric result for Γ_1 states that s vertices contain at most $\frac{1}{2}s\log_2 s$ edges [13, 29, 30, 40]. The optimal set is a subcube when $\log_2 s$ is integral, and it is a natural interpolation between subcubes for other s values.

For $\Gamma_{\le r}$ with $r > 1$, an exact edge-isoperimetric inequality is not known. Surprisingly, optimal shapes must be far from subcubes. For example, Hamming balls, which are essentially the optimal *vertex-isoperimetric* sets for Γ_1, are not only much better than subcubes, they are also not too far from optimal edge-isoperimetric sets for $\Gamma_{\le r}$. We give an easy argument for very rough approximate optimality of Hamming balls below and state a much sharper bound that was recently shown by the second author and co-author [23] in parallel work.

We start with preliminaries about the number of edges in a Hamming ball. We will lower bound the average degree in $\Gamma_{\le r}$ of a ball of radius k, that is, the value $\mathsf{e}_{\le r}(\mathsf{Ball}(0^d,k))$.

Proposition 4.2. *For $k \in [d]$ with $\lceil r/2\rceil \le k$,*

$$\mathsf{e}_{\le r}(\mathsf{Ball}(0^d,k)) \ge \frac{1}{2}\cdot\binom{k}{\lceil r/2\rceil}\cdot B(d-k,\lfloor r/2\rfloor)$$

Proof. Observe that

$$\mathsf{e}_{\le r}(\mathsf{Ball}(0^d,k)) \ge \frac{1}{2}\min_{x\in\mathsf{Ball}(0^d,k)}|\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x,r)|.$$

We first argue that the minimum occurs when $|x|=k$. Indeed, let $|x'|<k$ and $|x|=|x'|+1$ such that $x\oplus x'$ contains a single non-zero coordinate j.

CLAIM:

$$|\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x',r)| \ge |\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x,r))|.$$

First note that for any $z\in\{0,1\}^d$, $\mathsf{Ball}(z,r) = z\oplus\mathsf{Ball}(0^d,r)$. Let $y\in\mathsf{Ball}(0^d,r)$ have $y_j=0$ and let y' be the same as y except that bit $y_j=1$. Now $x'\oplus y = x\oplus y'$ and $x\oplus y = x'\oplus y'$. Therefore, (i) the number of such y for which $x'\oplus y\in\mathsf{Ball}(0^d,k)$ but $x\oplus y\notin\mathsf{Ball}(0^d,k)$ is equal to (ii) the number of such y' for which $x\oplus y'\in\mathsf{Ball}(0^d,k)$ but $x'\oplus y'\notin\mathsf{Ball}(0^d,k)$. Observe that (i) counts $\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x',r)\setminus(\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x,r))$ and (ii) counts a (strict) superset of $\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x,r)\setminus(\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x',r))$ since for any $z'\in\mathsf{Ball}(0^d,r)$ with $z'_j=1$, the element z equal to z' except that $z_j=0$ is also a member of $\mathsf{Ball}(0^d,r)$. (The strictness follows because for those $y\in\mathsf{Ball}(0^d,r)$ with $|y|=k$ have $y'\notin\mathsf{Ball}(0^d,r)$.) This proves the claim which implies that the minimum occurs when $|x|=k$.

Now fix some $x\in\mathsf{Ball}(0^d,k)$ with $|x|=k$. We count vectors z in $\mathsf{Ball}(0^d,k)\cap\mathsf{Ball}(x,r)$. The vector z may be obtained from x by flipping i bits from one to zero, and j bits from zero to one, as long as $i\le k$, $i+j\le r$, and $j\le i$. Therefore, by summing over (i,j) that meet these conditions, the number of vectors z is at least

$$\begin{aligned}
&\sum_{i\le\min\{r,k\}}\ \sum_{j\le\min\{r-i,i\}}\binom{k}{i}\binom{d-k}{j}\\
&\quad\ge\sum_{j\le\lfloor r/2\rfloor}\binom{k}{\lceil r/2\rceil}\binom{d-k}{j}\\
&\quad=\binom{k}{\lceil r/2\rceil}\cdot B(d-k,\lfloor r/2\rfloor).
\end{aligned}$$

This is a lower bound on the minimum, and thus average, degree of $\mathsf{Ball}(0^d, k)$ in $\Gamma_{\le r}$. □

Hamming Balls are nearly optimal for $r > 1$. We sketch the ideas behind the edge-isoperimetric inequality for $r > 1$. Consider $A \subseteq \{0,1\}^d$. Using standard down-shifting arguments, we may assume $A \subseteq \mathsf{Ball}(0^d, \lfloor \log_2 |A| \rfloor)$. Thus, the average degree of A in $\Gamma_{\le r}$ is roughly bounded by that of a ball of radius $\lfloor \log_2 |A| \rfloor$. In particular, $\mathsf{e}_{\le r}(A) \le B(d, \lfloor r/2 \rfloor) \cdot B(\lfloor \log_2 |A| \rfloor, \lceil r/2 \rceil)$. Ellis and Rashtchian [23] improve this estimate by roughly a factor of $(\log_2(d/\log_2 |A|))^{\lfloor r/2 \rfloor}$, nearly matching the Hamming ball upper bound.

4.1 Proof of Theorem 3.1(a). Now we prove the upper bound in our main theorem by exhibiting a randomized (p, δ)-edge-covering for sets of size n in $\Gamma_{\le r}$ achieving the claimed overhead.

For $p \le (d/\log_d n)^{r/2}$ we can achieve the bound simply using the universal algorithm. Our construction for $p > (d/\log_d n)^{r/2}$ consists of a partition of a random set of Hamming balls. We denote the radius of these balls by k, which will be carefully chosen to achieve efficient coverage and ensure proper load balancing. Intuitively, if k is too small, then each ball has too few edges and we will need too many balls to cover all the edges in $\Gamma_{\le r}$. On the other hand, if k is too large, then we risk having a large intersection $|S \cap A_i|$ on input S. We will see that the value of k that achieves the best overhead bounds is such that $B(d,k) \approx |S|/p = n/p$, but we consider all values of k for which the algorithm makes sense.

Randomized Edge-Covering Construction using Hamming Balls. Let n be the input set size and $k \ge \lceil r/2 \rceil$ to be any integer such that $B(d,k) \le n/p$. Observe that by our assumption $p > (d/\log_d n)^{r/2}$, we have that $k \le d/2 - \Omega(\sqrt{rd \log d})$.

The number of Hamming balls in the covering will be proportional to the ratio of the number of edges in $\Gamma_{\le r}$ and in $\mathsf{Ball}(0^d, k)$. Let this ratio ℓ_k be

$$\ell_k \triangleq \frac{|\mathsf{E}_{\le r}(\{0,1\}^d)|}{|\mathsf{E}_{\le r}(\mathsf{Ball}(0^d,k))|}.$$

Define a random distribution $\mathcal{A}_k(\delta)$ of p-tuples of subsets of $\{0,1\}^d$ with $(A_1, \ldots, A_p) \sim \mathcal{A}_k(\delta)$ chosen as follows: Let $x_1, \ldots, x_{Lp} \subseteq \{0,1\}^d$ be a uniformly random sequence of vectors where $L = \ln(1/(1-\delta)) \cdot \lceil \ell_k / p \rceil$. Define the random sets A_i as

$$A_i = \bigcup_{j \equiv i \bmod p} \mathsf{Ball}(x_i, k).$$

We begin by bounding ℓ_k, the ratio of the number of edges in $\Gamma_{\le r}$ to that in $\mathsf{Ball}(0^d, k)$. This bound on ℓ_k (and thus Lp) combines with Lemma 4.1 to immediately provide the claimed upper bounds in Theorem 3.1 on the reducer size and the number of times that a vector is replicated.

Proposition 4.3. *Fix $k \in [d]$ with*

$$\lceil r/2 \rceil \le k \le d/2 - \lceil r/2 \rceil.$$

Then,

$$\frac{|\mathsf{E}_{\le r}(\{0,1\}^d)|}{|\mathsf{E}_{\le r}(\mathsf{Ball}(0^d,k))|} < \left(\frac{2d}{k}\right)^{\lceil r/2 \rceil} \cdot \frac{2^{d+1}}{B(d,k)}.$$

Proof. The definition of $\mathsf{E}_{\le r}(\{0,1\}^d)$ and the lower bound on $\mathsf{e}_{\le r}(\mathsf{Ball}(0^d,k))$ in Proposition 4.2 imply that

$$\begin{aligned}\ell_k &= \frac{|\mathsf{E}_{\le r}(\{0,1\}^d)|}{|\mathsf{E}_{\le r}(\mathsf{Ball}(0^d,k))|} \\ &\le \frac{B(d,r)2^d}{\binom{k}{\lceil r/2 \rceil} \cdot B(d-k, \lfloor r/2 \rfloor) \cdot B(d,k)}.\end{aligned}$$

We invoke Proposition A.1 from the appendix and use the assumption $d/2 \ge 2\lceil r/2 \rceil$ to bound $B(d,r)/B(d, \lfloor r/2 \rfloor) \le \frac{5}{4} d^{\lceil r/2 \rceil} / r^{\lfloor r/2 \rfloor}$. Then, combining this with the simple bound $\binom{k}{j} \ge (k/j)^j$ we obtain that the upper bound on ℓ_k is at most

$$\begin{aligned}&\frac{5}{4} \cdot \frac{(\lceil r/2 \rceil)^{\lceil r/2 \rceil}}{r^{\lfloor r/2 \rfloor}} \cdot \frac{B(d, \lfloor r/2 \rfloor)}{B(d-k, \lfloor r/2 \rfloor)} \cdot \left(\frac{d}{k}\right)^{\lceil r/2 \rceil} \cdot \frac{2^d}{B(d,k)} \\ &< \frac{d^{\lfloor r/2 \rfloor}}{(d-k)^{\lfloor r/2 \rfloor}} \cdot \left(\frac{d}{k}\right)^{\lceil r/2 \rceil} \cdot \frac{2^{d+1}}{B(d,k)}.\end{aligned}$$

We used that $(\lceil r/2 \rceil)^{\lceil r/2 \rceil} / r^{\lfloor r/2 \rfloor}$ is at most 1 with r even and achieves its maximum value of 1.35 for r odd at $r = 5$. Finally, since $d - k - \lfloor r/2 \rfloor \ge d/2$, this is at most $(2d/k)^{\lceil r/2 \rceil} \cdot 2^{d+1}/B(d,k)$. □

The upper bound on the overhead in Theorem 3.1 follows from the following two lemmas, and the fact that $B(d,k) \le d^k$, which implies that $B(d,k)$ for $k \le (1-\gamma)\log_d n = \log_d(n/p)$ is at most n/p.

Lemma 4.4. *For all $S \subseteq \{0,1\}^d$ with $|S| = n$, for $k \ge \lceil r/2 \rceil$ and $B(d,k) \le n/p$, we have*

$$\begin{aligned}&\mathbb{E}[\mathsf{overhead}(\mathcal{A}_k(\delta), S)] \\ &\quad \le \max\{6\ln(1/(1-\delta)) \cdot (2d/k)^{\lceil r/2 \rceil}, 9\log_2 p\}.\end{aligned}$$

Moreover, the bound on the overhead holds almost surely.

Proof. Let $S \subseteq \{0,1\}^d$ with $|S| = n$. We will upper bound $\mathbb{E}[\max_i |S \cap A_i|]$ by proving an upper bound on $\mathbb{E}[|S \cap A_i|]$ for each $i \in [p]$, where the latter expectation is over the distribution on A_i induced by $\mathcal{A}_k(\delta)$. After doing so, we union bound over $[p]$ to guarantee the bound on the expected max with high probability.

Fix $i \in [p]$. The set A_i is the union of $\mathsf{Ball}(x_j, k)$ for L vectors x_j chosen uniformly and independently from $\{0,1\}^d$. Define the random variable $Z_j = |S \cap \mathsf{Ball}(x_j, k)|$ for $j \in [L]$. Then, defining $Z = \sum_j Z_j$, we have

$$|S \cap A_i| \leq \sum_j Z_j = Z.$$

We first upper bound $\mathbb{E}[Z] \geq \mathbb{E}[|S \cap A_i|]$. Observe that for a uniformly random x_j from $\{0,1\}^d$, every element of $\{0,1\}^d$ is equally likely to be in $\mathsf{Ball}(x_j, k)$. Therefore every $j \in [L]$,

$$\begin{aligned}\mathbb{E}[Z_j] &= \mathbb{E}[|S \cap \mathsf{Ball}(x_j, k)|] \\ &= \sum_{y \in S} \mathbf{Pr}[y \in \mathsf{Ball}(x_j, k)] \\ &= \sum_{y \in S} \frac{B(d,k)}{2^d} = \frac{n \cdot B(d,k)}{2^d}.\end{aligned}$$

Thus, recalling $L = \ln(1/(1-\delta)) \cdot \lceil \ell_k / p \rceil$, and using the upper bound on ℓ_k (the ratio of the number of edges in $\mathsf{Ball}(0^d, k)$ and in $\Gamma_{\leq r}$) from Proposition 4.3, we compute

$$\begin{aligned}\mathbb{E}[Z] &= L \cdot \frac{n \cdot B(d,k)}{2^d} \\ &= \ln(1/(1-\delta)) \cdot \lceil \ell_k / p \rceil \cdot \frac{n \cdot B(d,k)}{2^d} \\ &< 3 \ln(1/(1-\delta)) \cdot (2d/k)^{\lceil r/2 \rceil} \cdot \frac{n}{p}.\end{aligned}$$

For concentration, we use Bernstein's inequality (e.g., Thm. 3.6 in [19]), which applies since $0 \leq Z_j \leq B(d,k)$ and $\mathbb{E}[Z_j^2] \leq B(d,k)\mathbb{E}[Z_j]$. Therefore, for any $\lambda \geq \mathbb{E}[Z]$,

$$\begin{aligned}\mathbf{Pr}[Z > 2\lambda] &\leq e^{-\frac{1}{2}\lambda^2/(B(d,k)\mathbb{E}[Z] + \lambda B(d,k)/3)} \\ &\leq e^{-\frac{3}{8}\lambda/B(d,k)}.\end{aligned}$$

For every $\lambda = \lambda(\theta) \geq \max\{\mathbb{E}[Z], 4B(d,k)(\log_2 p + \theta)\}$ the probability that $|S \cap A_i| \geq 2\lambda$ is at most $e^{-\theta}/p^{3/2}$. By the union bound over $[p]$, the probability that $\max_i |S \cap A_i|$ exceeds 2λ is at most $e^{-\theta}/p^{1/2}$. In particular, we have an upper bound on $\mathsf{overhead}(\mathcal{A}_k(\delta), S)$ of 2λ. □

We conclude with the correctness argument.

Lemma 4.5. *For any n, $\mathcal{A}_k(\delta)$ forms a (p,δ)-edge-covering of the n-subsets of $\{0,1\}^d$. Further, if δ is chosen to be $1 - 1/n^3$, then $(A_1, \ldots, A_p) \sim \mathcal{A}_k(\delta)$ covers all edges of any fixed input set of size n with probability at least $1 - 1/n$.*

Proof. Fix $S \in \{0,1\}^d$ with $|S| = n$. Since $\Gamma_{\leq r}$ is edge-transitive, for $x_j \in \{0,1\}^d$ chosen uniformly at random, each pair $\{u,v\} \in S$ of distance at most r satisfies $\{u,v\} \subseteq \mathsf{Ball}(x_j, k)$ with probability

$$1/\ell_k = |\mathsf{E}_{\leq r}(\mathsf{Ball}(0^d, k))| / |\mathsf{E}_{\leq r}(\{0,1\}^d)|.$$

Therefore, the probability that a fixed $\{u,v\}$ is not covered by $(A_1, \ldots, A_p) \in \mathcal{A}_k(\delta)$ is at most

$$\begin{aligned}(1 - 1/\ell_k)^{Lp} &\leq (1 - 1/\ell_k)^{\ln(1/(1-\delta))\ell_k} \\ &\leq e^{-\ln(1/(1-\delta))} = 1 - \delta.\end{aligned}$$

Each edge of $\mathsf{E}_{\leq r}(S)$ is covered with probability at least δ, and in expectation $(A_1, \ldots, A_p) \sim \mathcal{A}_k(\delta)$ covers at least a δ fraction of $\mathsf{E}_{\leq r}(S)$.

There are at most n^2 pairs in S so with $\delta = 1 - 1/n^3$ (in which case $\mathcal{A}_k(\delta)$ chooses $3p\lceil \ell_k/p \rceil \ln n$ uniformly random balls of radius k) a union bound implies that the probability that $(A_1, \ldots, A_p) \sim \mathcal{A}_k(\delta)$ covers all pairs in S is at least $1 - 1/n$. □

5 The Lower Bound

We now prove the lower bound of Theorem 3.1(b). Let $\mathcal{A}$ be any randomized (p,δ)-edge cover of the n-subsets of $\{0,1\}^d$ for $\Gamma_{\leq r}$.

We will use a variant of Yao's minimax principle by exhibiting a "hard" distribution on subsets S of $\{0,1\}^d$ of size n. The usual form of the argument would then be to show that the distribution has the property that the expected overhead of any fixed $(A_1, \ldots, A_p)$ in the support of $\mathcal{A}$ is large and conclude that the expected overhead for $\mathcal{A}$ on some element of the support of the hard distribution is large.

However, this does not quite work. Since $\mathcal{A}$ produces a large fraction of the similar pairs of S *only in expectation*, its support may contain $(A_1, \ldots, A_p)$ that achieve low overhead but without producing many similar pairs. Instead, we show that (1) the expected overhead under the hard distribution must be large for any tuple $(A_1, \ldots, A_p)$ such that $|\bigcup_{i \in [p]} \mathsf{E}_{\leq r}(A_i)|$ is at least a $\delta/2$ fraction of $|\mathsf{edges}(\Gamma_{\leq r})|$ and that (2) any $(A_1, \ldots, A_p)$ that fails to have this property does badly at covering subsets S chosen according to this distribution and hence such tuples must only be a small fraction of the probability mass of $\mathcal{A}$.

The hard input distribution $\mathcal{D}_{n,d}$ over sets $S \subseteq \{0,1\}^d$ of size n will choose S to be a random sub-sampled Hamming ball of suitable size so that S has relatively high density within that ball.

Definition 5.1. *Let $d \in \mathbb{Z}^+$ and $r \in [d]$ be positive integers. Given $n \in [2^{d-1}]$, let $R = R(n,d,r) \in \mathbb{Z}^+$ denote the unique positive integer such that r divides R and*

$$B(d, R-r) < n \leq B(d,R).$$

Define distribution $\mathcal{D}_{n,d}$ on subsets S of $\{0,1\}^d$ of size n by first choosing $x \in \{0,1\}^d$ uniformly at random, then choosing a random subset of n elements of $\mathsf{Ball}(x,R)$.

Using the fact that $\mathcal{D}_{n,d}$ samples edges uniformly from $\Gamma_{\leq r}$ we have the following simple property of potential covers.

Proposition 5.2. *If $(A_1,\ldots,A_p)$ satisfies*

$$|\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i)| \leq \alpha \cdot |\mathsf{edges}(\Gamma_{\leq r})|$$

for some $\alpha \in [0,1]$, then

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}}\left[\left|\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i \cap S)\right|\right] \leq \alpha \cdot |\mathsf{E}_{\leq r}(S)|.$$

Proof. Observe that

$$\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i \cap S) = (\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i)) \cap \mathsf{E}_{\leq r}(S).$$

For S chosen according to $\mathcal{D}_{n,d}$, the set $\mathsf{E}_{\leq r}(S)$ contains each edge of $\Gamma_{\leq r}$ with equal probability, so the claim follows by linearity of expectation. □

Observing that $f_{\leq r}(n,d,p,\delta)$ is precisely

$$f_{\leq r}(n,d,p,\delta) = \min_{\mathcal{A}} \max_{S\subseteq\{0,1\}^d:\,|S|=n} \mathsf{overhead}(\mathcal{A},S),$$

Yao's minimax principle [58] applied to distribution $\mathcal{D}_{n,d}$ yields the following.

Lemma 5.3. *Let $\delta \in [0,1]$ and A_δ^{good} be the set of tuples $(A_1,\ldots,A_p)$ of subsets of $\{0,1\}^d$ such that $|\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i)| \geq \frac{\delta}{2}\cdot|\mathsf{edges}(\Gamma_{\leq r})|$.*

$$f_{\leq r}(n,d,p,\delta) \geq \frac{\delta}{2}\cdot \max_{(A_1,\ldots,A_p)\in A_\delta^{good}} \mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}}\left[\max_i |A_i\cap S|\cdot\frac{p}{n}\right].$$

Proof. Let $\mathcal{A}$ be a randomized (p,δ)-edge cover for the n subsets of $\Gamma_{\leq r}$. Therefore the expectation over $(A_1,\ldots,A_p)\sim\mathcal{A}$ and $S\sim\mathcal{D}_{n,d}$ of $|\bigcup_{i\in[p]}\mathsf{E}_{\leq r}(A_i\cap S)|$ must be at least $\delta|\mathsf{E}_{\leq r}(S)|$, since the $\delta|\mathsf{E}_{\leq r}(S)|$ bound holds for every choice of S. By Proposition 5.2 for $(A_1,\ldots,A_p)\notin A_\delta^{good}$ we have

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}} |\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i\cap S)| \leq \delta|\mathsf{E}_{\leq r}(S)|/2.$$

Since the most this quantity can be is $|\mathsf{E}_{\leq r}(S)|$, by Markov's inequality, the probability that $\mathcal{A}$ produces $(A_1,\ldots,A_p)$ that is in A_δ^{good} is at least $\delta/2$ and since all quantities are non-negative, the lower bound follows. □

For the remainder of this section we fix $(A_1,\ldots,A_p)$ in A_δ^{good}, so that

$$|\bigcup_{i\in[p]} \mathsf{E}_{\leq r}(A_i)| \geq \frac{\delta}{2}|\mathsf{edges}(\Gamma_{\leq r})| = \delta B(d,r)2^{d-2}. \quad (5.1)$$

We prove that $\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}} \max_i |A_i\cap S|$ must be large. Note that for sufficiently small r, and sufficiently large δ, equation (5.1) implies a lower bound on the total number of edges of Γ_r covered by the A_i.

Proposition 5.4. *Suppose that $r \leq \sqrt{d/2}$ and $\delta \geq 4/\sqrt{d}$. If $(A_1,\ldots,A_p)\in A_\delta^{good}$. Then*

$$|\bigcup_{i\in[p]} \mathsf{E}_r(A_i)| \geq \frac{\delta}{4}\cdot\binom{d}{r}\cdot 2^{d-1}$$

Proof. There are $B(d,r-1)2^{d-1}$ edges of $\Gamma_{\leq r}$ that are not in Γ_r. For $r \leq \sqrt{d/2}$, $\binom{d}{r-1}/\binom{d}{r} = \frac{r}{d-r+1} < \frac{1}{\sqrt{d}+1}$ and hence $B(d,r-1) \leq B(d,r)/\sqrt{d} \leq \frac{\delta}{4}\cdot B(d,r)$. Combining with (5.1), we have that at least $\frac{\delta}{4}B(d,r)2^{d-1} \geq \frac{\delta}{4}\binom{d}{r}2^{d-1}$ edges of Γ_r are contained in $\bigcup_{i\in[p]}\mathsf{E}_r(A_i)$. □

Next, we reduce the problem to reasoning about covers that only include necessary elements. Intuitively, it suffices to cover each vertex with fewer sets than its degree.

Definition 5.5. *A tuple $(A_1,\ldots,A_p)$ is* r-pruned *if for every $x\in\{0,1\}^d$, $|\{i:\ x\in A_i\}| \leq B(d,r)-1$.*

Lemma 5.6. *For any $(A_1,\ldots,A_p)$ there exists a r-pruned $(A'_1,\ldots,A'_p)$ such that $A'_i\subseteq A_i$ for all $i\in[p]$ and $\bigcup_{i\in[p]}\mathsf{E}_{\leq r}(A'_i) = \bigcup_{i\in[p]}\mathsf{E}_{\leq r}(A_i)$.*

Proof. Define $\mathsf{w}(x) = |\{i \in [p] \ : \ x \in A_i\}|$. For $i \in [p]$, say that a set A_i is *pivotal with respect to* x if there exists an edge $\{x,y\}$ in $\Gamma_{\le r}$ with $x \neq y$ such that $\{x,y\} \subseteq A_i$ while $\{x,y\} \subsetneq A_j$ for all $j < i$. For each $x \in \{0,1\}^d$, remove x from all but the pivotal sets for x. Let $(A'_1, \ldots, A'_p)$ be the resulting tuple of sets. Clearly, $\bigcup_i \mathsf{E}_{\le r}(A'_i) = \bigcup_i \mathsf{E}_{\le r}(A_i)$ by construction. For each x, the number of pivotal sets for is at most the size of the open neighborhood of x in $\Gamma_{\le r}$, which is $B(d,r) - 1$. Therefore, $w(x) \le B(d,r) - 1$ for all x, and hence, $(A'_1, \ldots, A'_p)$ is r-pruned. □

With these basic preliminaries out of the way we describe the overall proof strategy.

Proof Strategy. Lemma 5.6 allows us to assume without loss of generality that $(A_1, \ldots, A_p)$ is r-pruned since pruning yields the same set of similar pairs covered and does not increase overhead.

Our argument capitalizes on the following tradeoff. We have two cases: For the first case, if $\sum_{i\in[p]} |A_i|$ is much larger than 2^d then the algorithm will replicate vertices many times, implying a large overhead. For the second case, if $\sum_{i\in[p]} |A_i|$ is closer to 2^d, then we will prove that the average edge density of induced edges of $\Gamma_{\le r}$ in the sets A_i must be large. In other words, in the second case, the weighted average of $\mathsf{e}_{\le r}(A_i)$ is large. By Proposition 5.4, this means that the weighted average of $\mathsf{e}_r(A_i)$ is also large.

Using a key combinatorial lemma which shows that sets A with large $\mathsf{e}_r(A)$ must also have large $\mathsf{e}_{\le R}(A)$ where R is the distance defined by distribution $\mathcal{D}_{n,d}$ and hence, in the second case, the sum of the $|\mathsf{E}_{\le R}(A_i)|$ is large.

Finally, it is not hard to see that $|\mathsf{E}_{\le R}(A_i)|$ being large is equivalent to saying that the expected size of $|A_i \cap \mathsf{Ball}(x,R)|$ is large for x a randomly chosen element of A_i and it is not hard to see that a similar property also applies to high density random subsets S of $\mathsf{Ball}(x,R)$; like elements chosen according to $\mathcal{D}_{n,d}$.

This roughly suggests that a set S chosen from $\mathcal{D}_{n,d}$ will end up producing a large overhead at some set A_i that contains the center x of the ball $\mathsf{Ball}(x,R)$ used to define S. However, as x varies, which of these sets A_i are possible also varies and the choice of x and the set index i are correlated so choosing just one A_i would not let the rough argument succeed. Instead, using the fact that, without loss of generality, no element of $\{0,1\}^d$ needs to appear in too many sets A_i, we show that we can eliminate the bias by lower bounding the maximum overhead by the average overhead over all sets A_i with $x \in A_i$.

We begin by proving the last part of this strategy; i.e., we first relate the expected maximum size of $A_i \cap S$ to the number of pairs with distance at most R in A_i.

Lemma 5.7. *Let $d \in \mathbb{Z}^+$, $r \in [d]$, $n \in [2^{d-1}]$ and $R = R(n,d,r) = kr \le d/2$ for integer $k \ge 2$. Then for any r-pruned $\{A_1, \ldots, A_p\}$ with each $A_i \subseteq \{0,1\}^d$,*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}} \left[\max_{i\in[p]} |A_i \cap S| \right] \\
&> \frac{B(d,R-r)}{2^{d-1}B(d,R)(B(d,r)-1)} \sum_{i\in[p]} |\mathsf{E}_{\le R}(A_i)| \\
&\ge \frac{R^{(r)} r!}{d^{2r}} \sum_{i\in[p]} |\mathsf{E}_{\le R}(A_i)|/2^d.
\end{aligned}$$

Proof. The distribution $\mathcal{D}_{n,d}$ first chooses a center $x \in \{0,1\}^d$ uniformly at random and then chooses $S \subseteq \mathsf{Ball}(x,R)$ uniformly at random from all subsets of size n. Let $\mathcal{D}^x_{n,d}$ denote the conditional distribution of S given a fixed choice of x. Then, since $\{A_1, \ldots, A_p\}$, for all $x \in \{0,1\}^d$,

$$\mathsf{w}(x) = |\{i \in [p] \ : \ x \in A_i\}| \le B(d,r) - 1.$$

We can restrict our choice of maximum to sets containing the center x and lower bound the maximum over such sets by their average to obtain

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}} \left[\max_{i\in[p]} |A_i \cap S| \right] \\
&= \frac{1}{2^d} \sum_{x\in\{0,1\}^d} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[\max_{i\in[p]} |A_i \cap S| \right] \\
&\ge \frac{1}{2^d} \sum_{x\in\bigcup_i A_i} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[\max_{i\in[p]} |A_i \cap S| \right] \\
&\ge \frac{1}{2^d} \sum_{x\in\bigcup_i A_i} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[\max_{i:\ x\in A_i} |A_i \cap S| \right] \\
&\ge \frac{1}{2^d} \sum_{x\in\bigcup_i A_i} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[\sum_{i:\ x\in A_i} \frac{|A_i \cap S|}{B(d,r)-1} \right] \\
&\text{since } 0 < w(x) \le B(d,r) - 1 \\
&= \frac{1}{2^d} \sum_{x\in\bigcup_i A_i} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[\sum_{i\in[p]} \mathbb{1}_{x\in A_i} \cdot \frac{|A_i \cap S|}{B(d,r)-1} \right] \\
&= \frac{1}{2^d} \sum_{x\in\bigcup_i A_i} \sum_{i\in[p]} \mathbb{1}_{x\in A_i} \cdot \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[\frac{|A_i \cap S|}{B(d,r)-1} \right] \\
&= \frac{1}{2^d(B(d,r)-1)} \sum_{i\in[p]} \sum_{x\in A_i} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^x_{n,d}} \left[|A_i \cap S| \right]. \qquad (5.2)
\end{aligned}$$

Now for each $i \in [p]$ and $x \in A_i$, we have $S \subseteq \mathsf{Ball}(x, R)$; therefore

$$\begin{aligned}
&\sum_{x \in A_i} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^x_{n,d}} [|A_i \cap S|] \\
&= \sum_{x \in A_i} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^x_{n,d}} \left[\sum_{y \in \mathsf{Ball}(x,R) \cap A_i} \mathbb{1}_{y \in S} \right] \\
&= \sum_{x \in A_i} \sum_{y \in \mathsf{Ball}(x,R) \cap A_i} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^x_{n,d}} [\mathbb{1}_{y \in S}] \\
&= \sum_{x \in A_i} \sum_{y \in \mathsf{Ball}(x,R) \cap A_i} \frac{n}{B(d,R)} \\
&= \frac{n}{B(d,R)} \cdot 2|\mathsf{E}_{\le R}(A_i)|
\end{aligned}$$

since the double summation counts each pair $\{x, y\}$ in A_i of Hamming distance at most R exactly twice. Now, by construction of distribution $\mathcal{D}_{n,d}$, n is larger than $B(d, R-r)$, so inserting this bound in (5.2) yields that $\mathop{\mathbb{E}}_{S \sim \mathcal{D}_{n,d}} [\max_i |A_i \cap S|]$ is larger than $\frac{B(d,R-r)}{2^{d-1}B(d,R)(B(d,r)-1)} \sum_{i \in [p]} |\mathsf{E}_{\le R}(A_i)|$ as required. Now, $B(d,r) \le d^r/r!$ and, since $R > r$, Proposition A.1 proved in the appendix shows that $B(d,R)/B(d,R-r) \le (d+1)d^{r-1}/R^{(r)}$ which is at most $2d^r/R^{(r)}$. Together these yield the final claimed bound. □

To apply Lemma 5.7, we need a lower bound on $|\mathsf{E}_{\le R}(A_i)|$. We obtain such a lower bound by proving a purely graph-theoretic result. This result relates different distances in subsets of the hypercube. More precisely, we prove that subsets of $\{0,1\}^d$ with sufficiently large density of Hamming distance r pairs (i.e., sufficiently dense induced subgraphs of Γ_r) also have a relatively large density of pairs at Hamming distance at most R when R is a multiple of r. This means that we can reduce the problem of lower bounding $|\mathsf{E}_{\le R}(A_i)|$ to that of lower bounding $\mathsf{e}_r(A_i)$. We give this lemma and its proof in the next subsection.

5.1 Relating distance r density and distance R density on the hypercube

Lemma 5.8. *Fix $b \in \{0, 1, \ldots, \lfloor r/2 \rfloor\}$, and assume that r divides R with $r < R$. Assume that $A \subseteq \{0,1\}^d$ satisfies $\mathsf{e}_r(A) \ge 4\binom{R-r}{b+1}\binom{d}{r-b-1}$. Then*

$$\mathsf{e}_{\le R}(A) \ge \frac{\mathsf{e}_r(A)^{R/r}}{4^{R/r} \cdot R! \cdot d^{bR/r}}.$$

This lemma is nearly tight; examples include Hamming balls of varying radii and subcubes of varying dimensionality. Some lower bound on $\mathsf{e}_r(A)$ is necessary for this to hold since unions of well-separated balls of radius r have many pairs at distance up to r, but there are no pairs precisely at distances between $2r$ and R.

The overall approach to proving Lemma 5.8 is to define a certain class of paths of total length R/r in the induced subgraph $\Gamma_r[A]$ and prove that this set of paths is large. The start and end of each such path yields a pair vertices of A at distance at most R in the hypercube. We then show that any pair of vertices in the hypercube can be connected by relatively few such paths, yielding a large lower bound on the total number of pairs of vertices in A of distance at most R.

Definition 5.9. *Fix $b \in \{0, 1, \ldots, \lfloor r/2 \rfloor\}$, and assume that r divides R with $r < R$. An (R, b)-*path *is a sequence $(v_0, v_1, \ldots, v_{R/r}) \in \{0,1\}^{d \times (R/r+1)}$ with $\mathsf{dist}(v_{j-1}, v_j) = r$ and $\mathsf{dist}(v_0, v_j) \ge j(r-2b)$ for every $j \in [R/r]$. Let $\pi_{R,b}(A)$ denote the number of (R,b)-paths with all vectors in $A \subseteq \{0,1\}^d$.*

Note that an (R,b)-path generalizes a Γ_r shortest path, which is an $(R,0)$-path. Lemma 5.8 is the immediate consequence of the following two lemmas.

Lemma 5.10. *Fix $b \in \{0, 1, \ldots, \lfloor r/2 \rfloor\}$, and assume that r divides R with $r < R$. Let $A \subseteq \{0,1\}^d$ be a subset and define $N = |A|$ and $M = |\mathsf{E}_r(A)|$. If $\frac{M}{N} \ge 4\binom{R-r}{b+1}\binom{d}{r-b-1}$, then*

$$\pi_{R,b}(A) \ge N \left(\frac{M}{4N} \right)^{R/r}.$$

Lemma 5.11. *Fix $b \in \{0, 1, \ldots, \lfloor r/2 \rfloor\}$, and assume that r divides R with $r < R$. For any $A \subseteq \{0,1\}^d$,*

$$|\mathsf{E}_{\le R}(A)| \ge \frac{\pi_{R,b}(A)}{R! \cdot d^{bR/r}}.$$

More precisely, from Lemmas 5.10 and 5.11 we have the following, which yields Lemma 5.8,

$$\begin{aligned}
\mathsf{e}_{\le R}(A) = \frac{|\mathsf{E}_{\le R}(A)|}{|A|} &\ge \frac{\pi_{R,b}(A)}{R! \cdot d^{Rb/r} \cdot |A|} \\
&\ge \frac{e_r(A)^{R/r}}{4^{R/r} \cdot R! \cdot d^{Rb/r}}.
\end{aligned}$$

We begin by proving Lemma 5.10; its proof is similar to proofs of lower bounds on the number of walks in a graph given by Alon and Rusza [7] and Katz and Tao [35].

Proof of Lemma 5.10. We prove the bound by induction on N. For $N \le 2$, the bound holds trivially. Let δ^* denote the minimum degree in the subgraph of Γ_r

induced by A. We first consider the case with $\delta^* \geq M/(2N)$. We will count (R,b)-paths $(v_0,\ldots,v_{R/r})$ by lower bounding the number of choices for v_{j+1} given the path up to v_j. We will argue that

$$\pi_{R,b}(A) \geq \delta^* N \prod_{j=1}^{R/r-1} \left[\delta^* - \binom{jr}{b+1}\binom{d}{r-b-1}\right]. \quad (5.3)$$

There are $\delta^* N$ possibilities for v_0. For $j = 1,2,\ldots,R/r-1$, we say that a vertex v_{j+1} is *bad* if

$$\mathsf{dist}(v_0,v_{j+1}) < \mathsf{dist}(v_0,v_j) + (r-2b).$$

We must exclude the bad neighbors of v_j when choosing v_{j+1}. Each neighbor v_{j+1} of v_j in Γ_r differs from v_j in exactly r coordinates. Crucially, for v_{j+1} to be bad, at least $b+1$ of the differing coordinates must be in the support of $v_0 \oplus v_j$ because each such coordinate used in this step reduces the distance of v_{j+1} from v_0 by exactly 2. Since $|v_0 \oplus v_j| = \mathsf{dist}(v_0,v_j) \leq jr$, and the other $r-b-1$ coordinates are arbitrary, there are at most

$$\binom{jr}{b+1}\binom{d}{r-b-1}$$

bad neighbors of v_j in Γ_r. We have assumed that δ^* satisfies

$$\delta^* \geq \frac{M}{2N} \geq 2\binom{R-r}{b+1}\binom{d}{r-b-1} \geq 2\binom{jr}{b+1}\binom{d}{r-b-1}$$

Since each term in the product in (5.3) is at least $\delta^*/2$, we conclude $\pi_{R,b}(A) \geq N\left(\frac{M}{4N}\right)^{R/r}$.

Now assume that the minimum degree δ^* is less than $M/(2N)$. Remove a vertex with degree δ^*. The resulting graph on $N-1$ vertices has larger average degree. By induction, the number of (R,b)-paths on $N-1$ vertices is at least

$$(N-1)\left(\frac{M-\delta^*}{4(N-1)}\right)^{R/r}.$$

We claim this is at least $N(M/(4N))^{R/r}$. Rearranging both sides, we need to show

$$\left(\frac{M-\delta^*}{M}\right)^{(R/r)/((R/r)-1)} \geq \frac{N-1}{N}.$$

By the assumptions $\delta^* < M/(2N)$ and $R/r \geq 2$, we have, as desired,

$$\left(1-\frac{\delta^*}{M}\right)^{(R/r)/((R/r)-1)} > \left(1-\frac{1}{2N}\right)^{(R/r)/((R/r)-1)} \geq \left(1-\frac{1}{2N}\right)^2 > 1-\frac{1}{N}.$$

□

We now finish the proof of Lemma 5.8 by proving Lemma 5.11 which says that the number of pairs of distance $\leq R$ in A is relatively large compared to the number of (R,b)-paths in A.

Proof of Lemma 5.11. Consider a pair $(u,v) \in A \times A$ with $\mathsf{dist}(u,v) \leq R$. We claim that the number of (R,b)-paths between u and v is at most $R! \cdot d^{bR/r}$. Such a path is specified by a sequence of edges in Γ_r, and once we fix u and v, this sequence is specified by

1. a size R multiset M containing coordinates in $[d]$ differing between adjacent vertices,
2. a partition Π of M into R/r sets of size r.

We claim that there are at most $d^{bR/r}$ choices for M and $R!$ choices for Π. We simply upper bound the count for Π by the number of permutations of R elements. Moving on to M, let $k = \mathsf{dist}(u,v)$ and recall that the (R,b)-path definition requires $k \geq R - 2bR/r$. Observe that M contains the k elements in the support of $u \oplus v$. Additionally, M contains $(R-k)$ elements that come in identical pairs. That is, they form a partition into $(R-k)/2$ pairs of identical elements. Once $u \oplus v$ is fixed, these pairs determine M, and there are at most $d^{(R-k)/2} \leq d^{bR/r}$ choices for the pairs. □

5.2 Proof of Theorem 3.1(b). As described in our overall proof strategy, our lower bound involves a tradeoff between the average number of times $t = \sum_{i\in[p]} |A_i|/2^d$ that elements of $\{0,1\}^d$ are covered by $\{A_1,\ldots,A_p\}$ and the edge density of the individual A_i, and hence the size of the intersections of sets $S \sim \mathcal{D}_{n,d}$, which are similar to Hamming balls in structure. The function g that we define below helps us capture this overhead tradeoff.

Definition 5.12. *Define* $g : [1,\infty) \to \mathbb{R}^+$ *by*

$$g(\tau) = \min_{\substack{(A_1,\ldots,A_p)\in A_\delta^{good} \\ \sum_i |A_i| \leq \tau\cdot 2^d}} \mathop{\mathbb{E}}_{S\sim\mathcal{D}_{n,d}} \left[\max_i |A_i \cap S|\right] \cdot \frac{p}{n}$$

In particular, our overhead tradeoff is encapsulated in the following lemma.

Lemma 5.13. *Let $(A_1, \ldots, A_p) \in A_\delta^{good}$ and let $t = 2^{-d} \sum_i |A_i|$. Then,*

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}_{n,d}} [\max_i |A_i \cap S|] \geq \max\{t, g(t)\} \cdot \frac{n}{p}$$

Proof. We first prove the result for $f_{\leq r}(n, d, p, \delta)$. The bound in terms of $g(t)$ follows immediately from the definition of g. For the bound in terms of t, observe that, since each $x \in \{0,1\}^d$ has $\mathbf{Pr}_S[x \in S] = n2^{-d}$, by linearity of expectation we have

$$\begin{aligned}\mathop{\mathbb{E}}_{S \sim \mathcal{D}_{n,d}} [\max_i |A_i \cap S|] &\geq \mathop{\mathbb{E}}_{S \sim \mathcal{D}_{n,d}} \left[\frac{1}{p} \sum_{i \in [p]} |A_i \cap S|\right] \\ &= \frac{1}{p} \sum_{i \in [p]} |A_i| \cdot \frac{n}{2^d} = t \cdot \frac{n}{p}.\end{aligned}$$

□

We prove Theorem 3.1(b) by bounding $\max\{t, g(t)\}$. A lower bound on $\max\{t, g(t)\}$ is a value t^* such that either $t \geq t^*$ or $g(t) \geq t^*$ for any t. Therefore, we must prove $g(t) \geq t^*$ whenever $t \leq t^*$. To prove this, we will use the fact that g is a decreasing function and show that the value t^* is such that $g(t) \geq t$ whenever $t \leq t^*$. Indeed, this implies $g(t) \geq g(t^*) \geq t^*$ for all $t \leq t^*$.

Proof of Theorem 3.1(b). We first state the value of a threshold t^* such that $g(t) \geq t$ for all $t \leq t^*$ and then argue that it implies Theorem 3.1(b).

Lemma 5.14. *Let n, d, r, p be as in the statement of Theorem 3.1(b) and $R = R(n, d, r)$ be as in Definition 5.1. Define*

$$\beta = \begin{cases} \gamma/2 & \text{if } \gamma r > 1 \\ \gamma & \text{if } \gamma r \leq 1. \end{cases}$$

Then, $g(t) \geq t$ whenever $1 \leq t \leq t^*$ for

$$t^* = t^*(n, d, p, r) \triangleq \frac{\delta d^{\beta r - 2r^2/R}}{64 r^r R^{\gamma r}}.$$

By construction, R is $\Theta(\log_d n)$ and the assumption that $n \geq d^{r \log_2 d}$ implies that $d^{2r^2/R}$ is at most c^r for some constant c. Plugging these values into the formula for t^*, the two bounds of Theorem 3.1(b) follow immediately by combining Lemma 5.3 and Lemma 5.13.

The lower bound proof for $f_r(n, d, p, \delta)$ is almost identical, but it uses a variant definition of g and Lemma 5.13 that replaces the condition $(A_1, \ldots, A_p) \in A_\delta^{good}$ with the condition that the conclusion of Proposition 5.4 holds, which is the only place that membership in A_δ^{good} is used in the proof of Lemma 5.13. □

It only remains to prove Lemma 5.14.

Proof of Lemma 5.14. Let $t \leq t^*$ and $(A_1, \ldots, A_p) \in A_\delta^{good}$ be a tuple satisfying $\sum_{i \in [p]} |A_i| = t \cdot 2^d$ and $g(t) = \mathop{\mathbb{E}}_{S \sim \mathcal{D}_{n,d}} [\max_i |A_i \cap S|] \cdot \frac{p}{n}$. Since Lemma 5.7 gives us a lower bound on $g(t)$ in terms of $\sum_i |\mathsf{E}_{\leq R}(A_i)|$, our goal is to lower bound the latter quantity, assuming $t \leq t^*$. We rewrite this quantity in terms of the distribution μ over $[p]$ with $\mu(i) = \frac{|A_i|}{t2^d}$ as

$$\begin{aligned}\frac{1}{2^d} \sum_{i \in [p]} |\mathsf{E}_{\leq R}(A_i)| &= \frac{1}{2^d} \sum_{i \in [p]} \mathsf{e}_{\leq R}(A_i) |A_i| \\ &= t \cdot \mathop{\mathbb{E}}_{i \sim \mu} [\mathsf{e}_{\leq R}(A_i)]. \end{aligned} \tag{5.4}$$

We apply Lemma 5.8 to lower bound the RHS in (5.4). Let $I \subseteq [p]$ be the indices i such that A_i satisfies $\mathsf{e}_r(A_i) \geq 4\binom{R-r}{b+1}\binom{d}{r-b-1}$ for $b = \lfloor \gamma r/2 \rfloor$. Observe that with this choice of b, we have $\beta r \leq b + 1$ since either $\beta r = \gamma r \leq 1 \leq b + 1$ or $\beta r = \gamma r/2 \leq \lfloor \gamma r/2 \rfloor + 1 = b + 1$. Using Jensen's inequality for $z \mapsto z^{R/r}$, we obtain

$$\begin{aligned}\mathop{\mathbb{E}}_{i \sim \mu} [\mathsf{e}_{\leq R}(A_i)] &\geq \sum_{i \in I} \mu(i) \cdot \frac{\mathsf{e}_r(A_i)^{R/r}}{4^{R/r} \cdot R! \cdot d^{bR/r}} \\ &\geq \left(\sum_{i \in I} \mu(i) \mathsf{e}_r(A_i)\right)^{R/r} \cdot \frac{1}{4^{R/r} \cdot R! \cdot d^{bR/r}}. \end{aligned} \tag{5.5}$$

Now, since $\mu(i)\mathsf{e}_r(A_i) = \frac{|A_i| e_r(A_i)}{t2^d} = \frac{|\mathsf{E}_r(A_i)|}{t2^d}$, and $(A_1, \ldots, A_p) \in A_\delta^{good}$, Proposition 5.4 implies that

$$\sum_{i \in [p]} \mu(i) \mathsf{e}_r(A_i) \geq \frac{\delta}{8t} \binom{d}{r}.$$

Therefore, to lower bound $\mathop{\mathbb{E}}_{i \sim \mu} [\mathsf{e}_{\leq R}(A_i)]$, it suffices to upper bound $\sum_{i \notin I} \mu(i) \mathsf{e}_r(A_i)$, which is at most $4\binom{R-r}{b+1}\binom{d}{r-b-1}$ by definition. Now, since $t \leq t^*$, $\binom{d}{r} \geq (d/r)^r$ and $\beta r \leq \min\{b + 1, \gamma r\}$, we have

$$\begin{aligned}\frac{\delta}{16t} \binom{d}{r} &\geq \frac{\delta}{16t^*} \binom{d}{r} \geq 4R^{\gamma r} d^{r - \beta r + 2r^2/R} \\ &> 4R^{b+1} d^{r-b-1} \geq \sum_{i \notin I} \mu(i) \mathsf{e}_r(A_i).\end{aligned}$$

Therefore,

$$\sum_{i \in I} \mu(i) \mathsf{e}_r(A_i) \geq \frac{\delta}{16t} \binom{d}{r} \geq 4R^{\gamma r} d^{r-\beta r+2r^2/R},$$

and from (5.5) we have

$$\begin{aligned} &\mathop{\mathbb{E}}_{i \sim \mu} [\mathsf{e}_{\leq R}(A_i)] \\ &\quad \geq \left(4R^{\gamma r} d^{r-\beta r+2r^2/R}\right)^{R/r} \cdot \frac{1}{4^{R/r} \cdot R! \cdot d^{bR/r}} \\ &\quad \geq \frac{R^{\gamma R} d^{R-(\beta R+bR/r)+2r}}{R!} \end{aligned} \tag{5.6}$$

Now if $\gamma r \leq 1$ then $b = 0$ and $\beta = \gamma$ so $\beta R + bR/r = \gamma R$; alternatively, if $\gamma r > 1$, then $b \leq \gamma r/2$ and $\beta = \gamma/2$ and we have $\beta R + bR/r \leq \gamma R$. Therefore, from (5.6) we have

$$\mathop{\mathbb{E}}_{i \sim \mu} [\mathsf{e}_{\leq R}(A_i)] \geq \frac{R^{\gamma R} d^{R-\gamma R+2r}}{R!}. \tag{5.7}$$

Using Lemma 5.7 and plugging in (5.4) and (5.7) and the definition of $\gamma = \log_n p$ into the definition of $g(t)$, we have

$$\begin{aligned} g(t) &= \frac{\mathop{\mathbb{E}}_{S \sim \mathcal{D}_{n,d}} [\max_i |A_i \cap S|]}{n^{1-\gamma}} \\ &\geq t \cdot \frac{R^{(r)} r!}{2d^{2r} \cdot n^{1-\gamma}} \cdot \frac{R^{\gamma R} d^{R-\gamma R+2r}}{R!}. \end{aligned}$$

Since $R \geq 2r \geq 2$, we see that $g(t) \geq t \cdot \frac{R^{\gamma R} d^{(1-\gamma)R}}{n^{1-\gamma}}$. Since $n \leq B(d,R) \leq d^R/R!$, we derive $g(t) \geq t$ as required. □

6 Discussion and Future Work

We provided improved parallel algorithms for similarity joins under Hamming distance. We also proved communication lower bounds for one-round, local algorithms. Qualitatively, we showed that an overhead of $d^{\Theta(r)}$ is necessary and sufficient. Although we stated our technical results for Hamming distance, we gave a template for upper and lower bounds in general edge-transitive similarity graphs. A main algorithmic theme running through our results was to perform upfront data replication so that the processors need only one round of communication. This methodology may lead to improved algorithms for other distributed computation tasks.

Local Computation. We optimized for the maximum number of vertices assigned to any one processor and hence the maximum difficulty of the local computation required; however, our focus on communication leaves unanswered details about the actual computation of the close pairs. After the communication, any local algorithm may be used to compare candidates and output pairs satisfying the distance threshold. For example, processors could simply compare all received pairs. Clearly, by enlarging the groups of points sent to each processor compared to [1, 2, 49] we lose the efficiency of being able to only check pairs of points within small subcubes or balls. However, during the local computation stage, the processor may use a local partitioning scheme, possibly involving subcubes or balls, to reduce the overall number of comparisons.

Locality Sensitive Hashing (LSH) has been used successfully in practice for range queries and nearest neighbor search [8, 11, 28, 54]. Recent work [44, 48] exhibits an LSH-scheme for Hamming distance without false negatives.

Overall, we believe that further optimizations are needed to implement an efficient all-pairs similarly search. In the ℓ_2 metric, Aiger, Kaplan, and Sharir [4] design an algorithm for finding all close pairs using an edge-covering consisting of randomly shifted and rotated grids. Moreover, they demonstrate improvements over direct LSH approaches.

Our approach also extends naturally to support dynamic and streaming data efficiently: only the processors responsible for a vector need to be notified for its addition or deletion. Regarding randomness used for communication, work in the theoretical computer science community suggests bounded independence often suffices and improves performance [22, 43, 52].

Open Questions. We point out five concrete future directions left open by our work.

1. Can we close the gap between our upper and lower bounds on $f_{\leq r}(n, d, p, \delta)$? Currently, there is a gap between the exponents of $\gamma r/2$ versus $r/2$.

2. Our randomized edge-covering needed a random construction to find a family of Hamming balls that cover all edges with high probability. Is it possible to remove the error and decrease the communication by exhibiting an explicit construction?

3. Our lower bound holds for the restricted model where processors must communicate sets of vertices. Can we generalize our result and prove a lower bound on the maximum number of *bits* some processor must receive during the one round of communication?

4. The protocol for arbitrary similarity graphs uses copies of the optimal edge-isoperimetric shape.

Can we determine this shape and analyze the resulting protocol for other metrics? Prime candidates include the ℓ_1 distance and the edit distance over small alphabets.

5. Does our edge-covering methodology lead to improvements in practice? Empirical research on similarity joins often employs a "filter-then-verify" strategy. First, the algorithm or data structure coarsely prunes the set of candidates, often conservatively. Then, a brute force search reveals actual close pairs. It is worth exploring filters based on edge-optimal sets.

Acknowledgements

We thank Sivaramakrishnan Natarajan Ramamoorthy for bringing Sidorenko's conjecture and related results to our attention.

References

[1] Foto N Afrati, Anish Das Sarma, David Menestrina, Aditya Parameswaran, and Jeffrey D Ullman. Fuzzy Joins using MapReduce. In *ICDE*. IEEE, 2012.

[2] Foto N. Afrati, Anish Das Sarma, Anand Rajaraman, Pokey Rule, Semih Salihoglu, and Jeffrey D. Ullman. Anchor-Points Algorithms for Hamming and Edit Distances Using MapReduce. In *ICDT*, 2014.

[3] Thomas D Ahle, Rasmus Pagh, Ilya Razenshteyn, and Francesco Silvestri. On the Complexity of Inner Product Similarity Join. *arXiv*, abs/1510.02824, 2015.

[4] Dror Aiger, Haim Kaplan, and Micha Sharir. Reporting Neighbors in High-Dimensional Euclidean Space. *SIAM J. Comput.*, 43(4):1363–1395, 2014.

[5] Josh Alman, Timothy M Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. *arXiv*, abs/1608.04355, 2016.

[6] Josh Alman and Ryan Williams. Probabilistic Polynomials and Hamming Nearest Neighbors. In *Proceedings, 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 136–150, 2015.

[7] Noga Alon and Imre Z Ruzsa. Non-averaging subsets and non-vanishing transversals. *Journal of Combinatorial Theory, Series A*, 86(1):1–13, 1999.

[8] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and Optimal LSH for Angular Distance. In *NIPS*, pages 1225–1233, 2015.

[9] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal Hashing-based Time-Space Trade-offs for Approximate Near Neighbors. *arXiv*, abs/1608.03580, 2016.

[10] Alexandr Andoni and Ilya Razenshteyn. Tight Lower Bounds for Data-dependent Locality-sensitive Hashing. In *32nd Intl. Symp. on Comp. Geom. (SoCG)*, pages 9:1–9:11, 2016.

[11] Bahman Bahmani, Ashish Goel, and Rajendra Shinde. Efficient distributed locality sensitive hashing. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2174–2178, New York, NY, USA, 2012. ACM.

[12] Paul Beame, Paraschos Koutris, and Dan Suciu. Communication Steps for Parallel Query Processing. In *PODS*. ACM, 2013.

[13] Arthur J Bernstein. Maximally connected arrays on the n-cube. *SIAM J. Applied Mathematics*, 15(6):1485–1489, 1967.

[14] Sergei Bezrukov. Edge Isoperimetric Problems on Graphs. *Graph Theory and Combinatorial Biology*, 7:157–197, 1999.

[15] Béla Bollobás and Imre Leader. Sums in the grid. *Discrete Mathematics*, 162(1-3):31–48, 1996.

[16] Andrei Z Broder. Identifying and filtering near-duplicate documents. In *Combinatorial pattern matching*, pages 1–10. Springer, 2000.

[17] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388. ACM, 2002.

[18] Yeow Meng Chee, Charles J Colbourn, Alan CH Ling, and Richard M Wilson. Covering and packing for pairs. *Journal of Combinatorial Theory, Series A*, 120(7):1440–1449, 2013.

[19] Fan Chung and Linyuan Lu. Concentration Inequalities and Martingale Inequalities: a Survey. *Internet Mathematics*, 3(1):79–127, 2006.

[20] David Conlon, Jacob Fox, and Benny Sudakov. An Approximate Version of Sidorenkos Conjecture. *Geometric and Functional Analysis*, 20(6):1354–1366, 2010.

[21] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google News Personalization: Scalable Online Collaborative Filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.

[22] Martin Dietzfelbinger. Universal Hashing and k-Wise Independent Random Variables via Integer Arithmetic Without Primes. In *STACS*, 1996.

[23] David Ellis and Cyrus Rashtchian. Edge-Isoperimetric Inequalities for Powers of the Hypercube. *In Preparation*, 2016.

[24] Paul Erdos and Miklos Simonovits. Compactness Results in Extremal Graph Theory. *Combinatorica*, 2(3):275–288, 1982.

[25] Yunchao Gong and Svetlana Lazebnik. Iterative Quantization: A Procrustean Approach to Learning Binary Codes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 817–824. IEEE,

2011.

[26] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. WTF: The Who to Follow Service at Twitter. In *WWW*, 2013.

[27] Sariel Har-Peled. *Geometric Approximation Algorithms*, volume 173. American mathematical society Providence, 2011.

[28] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of Computing*, 8(1):321–350, 2012.

[29] Lawrence H Harper. Optimal assignments of numbers to vertices. *Journal of the Society for Industrial and Applied Mathematics*, pages 131–135, 1964.

[30] Sergiu Hart. A note on the edges of the n-cube. *Discrete Mathematics*, 14(2):157–163, 1976.

[31] Daniel Horsley. Generalising fisher's inequality to coverings and packings. *arXiv*, abs/1409.0485, 2014.

[32] R. Impagliazzo and R. Paturi. On the complexity of *k*-SAT. *J. Comp. Sys. Sci.*, 67:367–375, 2001.

[33] Yu Jiang, Guoliang Li, Jianhua Feng, and Wen-Syan Li. String Similarity Joins: An Experimental Evaluation. *Proceedings of the VLDB Endowment*, 7(8):625–636, 2014.

[34] Jeff Kahn, Gil Kalai, and Nati Linial. The Influence of Variables on Boolean Functions. In *SFCS*, 1988.

[35] Nets Hawk Katz and Terence Tao. A New Bound on Partial Sum-sets and Difference-sets, and Applications to the Kakeya Conjecture. *arXiv*, abs/9906.097, 1999.

[36] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000.

[37] Hans-Peter Lenhof and Michiel Smid. Sequential and Parallel Algorithms for the *k*-closest Pairs Problem. *International J. Comput. Geometry & Applications*, 5(03):273–288, 1995.

[38] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[39] Yeqing Li, Chen Chen, Wei Liu, and Junzhou Huang. Sub-Selective Quantization for Large-Scale Image Search. In *AAAI*, 2014.

[40] John H Lindsey. Assignment of Numbers to Vertices. *The American Mathematical Monthly*, 71(5):508–516, 1964.

[41] Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM J. Discrete Mathematics*, 21(4):930–935, 2007.

[42] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Transactions on Computation Theory*, 6(1):5, 2014.

[43] Anna Pagh, Rasmus Pagh, and Milan Ružić. Linear Probing with Constant Independence. *SIAM J. Comput.*, 39(3):1107–1120, September 2009.

[44] Rasmus Pagh. Locality-sensitive hashing without false negatives. In *SODA*, pages 1–9. SIAM, 2016.

[45] Rasmus Pagh, Ninh Pham, Francesco Silvestri, and Morten Stöckel. I/O-Efficient Similarity Join. In *Algorithms-ESA 2015*, pages 941–952. Springer, 2015.

[46] Rina Panigrahy, Kunal Talwar, and Udi Wieder. A Geometric Approach to Lower Bounds for Approximate Near-Neighbor Search and Partial Match. In *FOCS*, 2008.

[47] Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower Bounds on Near Neighbor Search via Metric Expansion. In *FOCS*, 2010.

[48] Ninh Pham and Rasmus Pagh. Scalability and total recall with Fast CoveringLSH. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM*, pages 1109–1118, 2016.

[49] Anish Das Sarma, Foto N. Afrati, Semih Salihoglu, and Jeffrey D. Ullman. Upper and Lower Bounds on the Cost of a Map-reduce Computation. *Proc. VLDB Endow.*, 6(4):277–288, February 2013.

[50] Alexander Sidorenko. A Correlation Inequality for Bipartite Graphs. *Graphs and Combinatorics*, 9(2-4):201–204, 1993.

[51] Narayanan Sundaram, Aizana Turmukhametova, Nadathur Satish, Todd Mostak, Piotr Indyk, Samuel Madden, and Pradeep Dubey. Streaming Similarity Search Over One Billion Tweets Using Parallel Locality-Sensitive Hashing. *Proceedings of the VLDB Endowment*, 6(14):1930–1941, 2013.

[52] Mikkel Thorup. High Speed Hashing for Integers and Strings. *arXiv*, abs/1504.06804, 2015.

[53] Rares Vernica, Michael J Carey, and Chen Li. Efficient Parallel Set-similarity Joins using MapReduce. In *SIGMOD*, pages 495–506. ACM, 2010.

[54] Hongya Wang, Jiao Cao, LihChyun Shu, and Davood Rafiei. Locality Sensitive Hashing Revisited: Filling the Gap Between Theory and Algorithm Analysis. In *CIKM*, 2013.

[55] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. Learning to Hash for Indexing Big Data — A Survey. *Proceedings of the IEEE*, 104(1):34–57, 2016.

[56] Chuan Xiao, Wei Wang, Xuemin Lin, Jeffrey Xu Yu, and Guoren Wang. Efficient Similarity Joins for Near-duplicate Detection. *ACM Transactions on Database Systems*, 36(3):15, 2011.

[57] Ismet Zeki Yalniz, Ethem F. Can, and R. Manmatha. Partial Duplicate Detection for Large Book Collections. In *CIKM*, 2011.

[58] Andrew C. Yao. Lower Bounds by Probabilistic Arguments. In *FOCS*, 1983.

[59] Reza Bosagh Zadeh and Ashish Goel. Dimension Independent Similarity Computation. *The Journal of Machine Learning Research*, 14(1):1605–1626, 2013.

[60] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. Discrete Collaborative Filtering. In *SIGIR*, 2016.

A Ball Ratios

Proposition A.1. *For $r < R \leq (d+1)/2$, $B(d,R)/B(d,R-r) \leq (d+1)d^{r-1}/R^{(r)}$.*

Proof. For $i \geq 0$ define

$$b_i = \frac{\binom{d}{R-i} + \cdots + \binom{d}{R}}{\binom{d}{R-i}}.$$

Both $B(d,R)$ and $B(d,R-r)$ contain the common terms $\binom{d}{0} + \cdots + \binom{d}{R-r} \geq \binom{d}{R-r}$ and so

$$B(d,R)/B(d,R-r) \leq b_r = \frac{\binom{d}{R-r} + \cdots + \binom{d}{R}}{\binom{d}{R-r}}.$$

Observe that by definition $b_0 = 1$ and for $i \geq 0$ we have the recurrence

$$\begin{aligned} b_{i+1} =& 1 + \frac{\binom{d}{R-i}}{\binom{d}{R-i-1}} \cdot b_i \\ =& 1 + \frac{d-R+i+1}{R-i} \cdot b_i. \end{aligned}$$

We prove by induction that for $i \geq 1$ we have $2 \leq b_i \leq (d+1)d^{i-1}/R^{(i)}$.
For the base case, since $b_0 = 1$ we have $b_1 = 1 + \frac{d-R+2}{R-1} b_0 = (d+1)/R$ and since $R \leq (d+1)/2$, we also have $b_1 \geq 2$. For $i \geq 1$,

$$\begin{aligned} b_{i+1} &= 1 + \frac{d-R+i+1}{R-i} \cdot b_i \\ &= 1 + \left(\frac{d}{R-i} - \frac{R-i-1}{R-i} \right) \cdot b_i \\ &\leq 1 + \frac{d}{R-i} \cdot b_i - \frac{R-i-1}{R-i} \cdot 2 \end{aligned}$$

since $b_i \geq 2$ by the inductive hypothesis

$$\leq \frac{d}{R-i} \cdot b_i \qquad \text{since } R > r \geq i.$$

Now applying the inductive hypothesis we have

$$\begin{aligned} b_{i+1} \leq \frac{d}{R-i} \cdot b_i &= \frac{d}{R-i} \cdot (d+1)d^{i-1}/R^{(i)} \\ &= (d+1)d^i/R^{(i+1)} \end{aligned}$$

as claimed. □

B Other Lower Bounds

We sketch two simple, but nontrivial, lower bounds for Hamming distance that achieve a different dependence on $|S|$ than the bounds we obtain using isoperimetric inequalities. For the discussion, let L denote the number of vertices a processor receives, not bits.

For the first hard distribution, pick one random ball of radius $r/2$ and include each vector in this ball in S with probability half. A processor may output at most all pairs among vertices it receives. On the other hand, all pairs in the input S are within distance r. Thus, we need

$$p\binom{L}{2} \geq \binom{B(d,r/2)}{2}.$$

Since $|S| = \Theta(B(d,r/2))$, we can rewrite this as

$$L \geq \Omega(|S|/\sqrt{p}).$$

For another hard distribution, consider picking

$$|S|/B(d,r/2)$$

random balls each of radius $r/2$, then subsampling each vector with probably half. Receiving L vertices leads to at most $\binom{L}{2}$ close pairs. The input has $\Theta(|S| \cdot B(d,r/2))$ close pairs. This gives

$$L \geq \Omega\left(\sqrt{B(d,r/2)} \cdot \sqrt{|S|/p} \right).$$