On Explicit Constructions of Extremely Depth Robust Graphs

Jeremiah Blocki
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Mike Cinkoske
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Seunghoon Lee
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Jin Young Son
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Abstract

A directed acyclic graph $G = (V, E)$ is said to be (e, d)-depth robust if for every subset $S \subseteq V$ of $|S| \leq e$ nodes the graph $G - S$ still contains a directed path of length d. If the graph is (e, d)-depth-robust for any e, d such that $e + d \leq (1 - \epsilon)|V|$ then the graph is said to be ϵ-extreme depth-robust. In the field of cryptography, (extremely) depth-robust graphs with low indegree have found numerous applications including the design of side-channel resistant Memory-Hard Functions, Proofs of Space and Replication and in the design of Computationally Relaxed Locally Correctable Codes. In these applications, it is desirable to ensure the graphs are locally navigable, i.e., there is an efficient algorithm GetParents running in time polylog $|V|$ which takes as input a node $v \in V$ and returns the set of v's parents. We give the first explicit construction of locally navigable ϵ-extreme depth-robust graphs with indegree $O(\log |V|)$. Previous constructions of ϵ-extreme depth-robust graphs either had indegree $\tilde{\omega}(\log^2 |V|)$ or were not explicit.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Mathematics of computing → Graph theory; Mathematics of computing → Paths and connectivity problems; Mathematics of computing → Combinatorics

Keywords and phrases Depth-Robust Graphs, Explicit Constructions, Data-Independent Memory Hard Functions, Proofs of Space and Replication

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.14

Related Version *Full Version*: https://arxiv.org/abs/2110.04190

Funding *Jeremiah Blocki*: This author was supported in part by the National Science Foundation under NSF CAREER Award CNS-2047272 and NSF Awards CCF-1910659 and CNS-1931443.
Seunghoon Lee: This author was supported in part by the Center for Science of Information at Purdue University (NSF CCF-0939370).
The opinions in this paper are those of the authors and do not necessarily reflect the position of the National Science Foundation.

1 Introduction

A depth-robust graph $G = (V, E)$ is a directed acyclic graph (DAG) which has the property that for any subset $S \subseteq V$ of at most e nodes the graph $G - S$ contains a directed path of length d, i.e., there is a directed path $P = v_0, \ldots, v_d$ such that $(v_i, v_{i+1}) \in E$ for each $i < d$ and $v_i \in V \setminus S$ for each $i \leq d$. As an example the complete DAG $K_N = (V = [N], E = \{(i, j) : 1 \leq i < j \leq n\}$ has the property that it is (e, d)-depth-robust for any integers e, d such that $e + d \leq N$. Depth-robust graphs have found many applications in cryptography including the design of data-independent Memory-Hard Functions (e.g.,[1, 3]), Proofs of

Space [9], Proofs of Replication [15, 11] and Computationally Relaxed Locally Correctable Codes [7]. In many of these applications it is desirable to construct depth-robust graphs with low-indegree (e.g., $\mathsf{indeg}(G) = O(1)$ or $\mathsf{indeg}(G) = O(\log N)$) and we also require that the graphs are *locally navigable*, i.e., given any node $v \in V = [N]$ there is an efficient algorithm $\mathsf{GetParents}(v)$ which returns the set $\{u : (u, v) \in E\}$ containing all of v's parent nodes in time $O(\mathsf{polylog}\, N)$. It is also desirable that the graph is (e, d)-depth robust for e, d as large as possible, e.g., the cumulative pebbling cost of a graph can be lower bounded by the product ed and in the context of Memory-Hard Functions we would like to ensure that the cumulative pebbling cost is as large as possible [5, 3]. Some cryptographic constructions rely on an even stronger notion called ϵ-*extreme depth-robust graphs* $G = (V, E)$ which have the property of being (e, d)-depth-robust for any integers e, d such that $e + d \leq (1 - \epsilon)N$, e.g., see [15, 14].

Erdös, Graham, and Szemeredi [10] gave a randomized construction of (e, d)-depth-robust graphs with $e, d = \Omega(N)$ and maximum indegree $O(\log N)$. Alwen, Blocki, and Harsha [2] modified this construction to obtain a locally navigable construction of (e, d)-depth-robust graphs with constant indegree 2 for $e = \Omega(N/\log N)$ and $d = \Omega(N)$. For any constant $\epsilon > 0$, Schnitger [17] constructed $(e = \Omega(N), d = \Omega(N^{1-\epsilon}))$-depth-robust graphs with constant indegree – the indegree $\mathsf{indeg}(G)$ does increase as ϵ gets smaller. These results are essentially tight as *any* DAG G which is $\left(\frac{N \cdot i \cdot \mathsf{indeg}(G)}{\log N}, \frac{N}{2^i}\right)$-reducible[1] for any $i \geq 1$ [1, 18]. If $\mathsf{indeg}(G) = o(\log N)$ then the graph cannot be (e, d)-depth robust with $e, d = \Omega(N)$ and similarly if $\mathsf{indeg}(G) = \Theta(1)$ plugging in $i = O(\log \log N)$ demonstrates that G cannot be $(e = \omega(N \log \log N/\log N), d = \omega(N))$-depth-robust.

Explicit Depth-Robust Graphs

All of the above constructions are randomized and do not yield explicit constructions of depth-robust graphs. For example, the DRSample construction of [2] actually describes a randomized distribution over graphs and proves that a graph sampled from the distribution is (e, d)-depth-robust with high probability. Testing whether a graph is actually (e, d)-depth-robust is computationally intractable [8, 6] so we cannot say that a particular sampled graph is depth-robust with 100% certainty. In fact, it might be possible for a dishonest party to build a graph $G = (V, E)$ which looks like an honestly sampled depth-robust graph but actually contains a small (secret) depth-reducing set $S \subseteq V$, i.e., such that $G - S$ does not contain any long paths. Thus, in many cryptographic applications one must assume that the underlying depth-robust graphs were generated honestly.

Li [13] recently gave an explicit construction of constant-indegree depth-robust graphs, i.e., for any $\epsilon > 0$, Li constructs a family of graphs $\{G_{N,\epsilon}\}$ such that each $G_{N,\epsilon}$ has N nodes, constant indegree, and is $(\Omega(N^{1-\epsilon}), \Omega(N^{1-\epsilon}))$-depth-robust. The construction of Li [13] is also locally navigable, but the graphs are not as depth-robust as we would like. Mahmoody, Moran, and Vadhan [14] gave an explicit construction of an ϵ-extreme depth-robust graph for any constant $\epsilon > 0$ using the Zig-Zag Graph Product constructions of [16]. However, the maximum indegree is as large as $\mathsf{indeg}(G) \leq \log^3 N$. Alwen, Blocki, and Pietrzak [4] gave a tighter analysis of [10] showing that the randomized construction of [10] yields ϵ-extreme depth-robust graphs with $\mathsf{indeg}(G) = O(\log N)$ although their randomized construction is not explicit nor was the graph shown to be locally navigable.

[1] If a DAG G is not (e, d)-depth-robust we say that it is (e, d)-reducible, i.e., there exists some set $S \subseteq V$ of size e such that $G - S$ contains no directed path of length d.

1.1 Our Contributions

We give explicit constructions of ϵ-extreme depth-robust graphs with maximum indegree $O(\log N)$ for any constant $\epsilon > 0$ and we also give explicit constructions of $(e = \Omega(N/\log N), d = \Omega(N))$-depth-robust graphs with maximum indegree 2. Both constructions are explicit and locally navigable. In fact, our explicit constructions also satisfy a stronger property of being δ-local expanders. A δ-local expander is a directed acyclic graph G which has the following property: for any $r, v \geq 0$ and any subsets $X \subseteq A = [v, v + r - 1]$ and $Y \subseteq B = [v + r, v + 2r - 1]$ of at least $|X|, |Y| \geq \delta r$ nodes the graph G contains an edge (x, y) with $x \in X$ and $y \in Y$. We remark that the construction of Computationally Relaxed Locally Correctable Codes [7] relies on a family of δ-local expanders which is a strictly stronger property than depth-robustness – for any $\epsilon > 0$, there exists a constant $\delta > 0$ such that any δ-local expander automatically becomes ϵ-extreme depth-robust [4].

1.2 Our Techniques

We first provide explicit, locally navigable, constructions of δ-bipartite expander graphs with constant indegree for any constant $\delta > 0$. A bipartite graph $G = ((A, B), E)$ with $|A| = |B| = N$ is a δ-bipartite expander if for *any* $X \subseteq A$ and $Y \subseteq B$ of size $|X|, |Y| \geq \delta N$ the bipartite graph G contains at least one edge $(x, y) \in E$ with $x \in X$ and $y \in Y$. The notion of a δ-bipartite expander is related to, but distinct from, classical notions of a graph expansion, e.g., we say that G is an (N, k, d)-expander if $\mathsf{indeg}(G) \leq k$ and for every subset $X \subseteq A$ (resp. $Y \subseteq B$) we have $|\mathsf{N}(X)| \geq (1 + d - d|X|/N)|X|$ (resp. $|\mathsf{N}(Y)| \geq (1 + d - d|Y|/N)|Y|$), where $\mathsf{N}(X)$ is defined to be all of the neighbors of X, i.e., $\mathsf{N}(X) \doteq \{y \in B : \exists x \in X \text{ s.t. } (x, y) \in E\}$. (Notation: We use $\mathsf{N}(X)$ (resp. N) to denote the neighbors of nodes in X (resp. number of nodes in a graph/bipartition).) Erdös, Graham, and Szemeredi [10] argued that a random degree k_δ bipartite graph will be a δ-bipartite expander with non-zero probability where the constant k_δ depends only on δ. As a building block, we rely on an explicit, locally navigable, construction of $(n = m^2, k = 5, d = (2 - \sqrt{3})/4)$-expander graphs for any integer m due to Gabber and Galil [12]. For any constant $\delta > 0$ we show how any (N, k, d)-expander graph G with $d < 0.5$ and $k = \Theta(1)$ can be converted into a δ-bipartite expander graph G' with N nodes and maximum indegree $\mathsf{indeg}(G') = \Theta(1)$. Intuitively, the construction works by "layering" $\ell = \Theta(1)$ copies of the (N, k, d)-expander graphs and then "compressing" the layers to obtain a bipartite graph G' with maximum indegree $k' \leq k^\ell$ – paths from the bottom layer to the top layer are compressed to individual edges.

The depth-robust graph construction of Erdös et al. [10] uses δ-bipartite expanders as a building block. By swapping out the randomized (non-explicit) construction of δ-bipartite expanders with our explicit and locally navigable construction, we obtain a family of explicit and locally navigable depth-robust graphs. Furthermore, for any $\epsilon > 0$ we can apply the analysis of Alwen et al. [4] to obtain explicit constructions of ϵ-extreme depth-robust graphs by selecting the constant $\delta > 0$ accordingly. Finally, we can apply a standard indegree reduction gadget of Alwen et al. [3] to obtain an $(e = N/\log N, d = \Omega(N))$-depth-robust graph with indegree 2.

2 Preliminaries

We use $[N] = \{1, \ldots, N\}$ to denote the set of all integers between 1 and N and we typically use $V = [N]$ to denote the set of nodes in our graph. It is often convenient to assume that $N = 2^n$ is a power of 2. Given a graph $G = (V = [N], E)$ and a subset $S \subseteq [N]$ we use $G - S$ to denote the graph obtained by deleting all nodes in S and removing any incident edges. Fixing

a directed graph $G = (V = [N], E)$ and a node $v \in V$, we use $\mathsf{parents}(v) = \{u : (u, v) \in E\}$ to denote the parents of node v and we let $\mathsf{indeg}(G) = \max_{v \in [N]} |\mathsf{parents}(v)|$ denote the maximum indegree of any node in G. We say a DAG G is (e, d)-reducible if there exists a subset $S \subseteq [N]$ of $|S| \leq e$ nodes such that $G - S$ contains no directed path of length d. If G is not (e, d)-reducible we say that G is (e, d)-depth-robust.

We introduce the notion of a δ-bipartite expander graph where the concept was first introduced by [10] and used as a building block to construct depth-robust graphs. Note that the specific name "δ-bipartite expander" was not used in [10]. We follow the notation of [2, 4].

▶ **Definition 1.** *A directed bipartite graph $G = ((A, B), E)$ with $|A| = |B| = N$ is called a* δ-bipartite expander *if and only if for any subset $X \subseteq A, Y \subseteq B$ of size $|X| \geq \delta N$ and $|Y| \geq \delta N$ there exists an edge between X and Y.*

▶ Remark 2. Observe that if $G = ((A, B), E)$ is a δ-bipartite expander then for any subset $X \subseteq A$ with $|X| \geq \delta N$ we must have $|\mathsf{N}(X)| > (1 - \delta)N$ where $\mathsf{N}(X) = \{y \in B : \exists x \in X \text{ s.t. } (x, y) \in E\}$ denotes the neighbors of X. If this were not the case then we could take $Y = B \setminus \mathsf{N}(X)$ and we have $|Y| \geq \delta N$ and, by definition of Y, we have no edges between X and Y contradicting the assumption that G is a δ-bipartite expander.

▶ **Definition 3.** *A directed bipartite graph $G = ((A, B), E)$ with $|A| = |B| = N$ is called an* (N, k, d)-expander *if $|E| \leq kN$ and for every subset $X \subseteq A$ (resp. $Y \subseteq B$) we have $|\mathsf{N}(X)| \geq \left[1 + d\left(1 - \frac{|X|}{N}\right)\right]|X|$ (resp. $|\mathsf{N}(Y)| \geq \left[1 + d\left(1 - \frac{|Y|}{N}\right)\right]|Y|$) where $\mathsf{N}(X) = \{y \in B : \exists x \in X \text{ s.t. } (x, y) \in E\}$ (resp. $\mathsf{N}(Y) = \{x \in A : \exists y \in B \text{ s.t. } (x, y) \in E\}$).*

Gabber and Galil [12] gave explicit constructions of $(N = m^2, k = 5, d = (2 - \sqrt{3})/5)$-expanders. Lemma 4 highlights the relationship between δ-bipartite expanders and the more classical notion of (N, k, d)-expanders.

▶ **Lemma 4.** *Let $0 < d < 1$ and let $\delta = \frac{(d+2)-\sqrt{d^2+4}}{2d}$. If a directed bipartite graph $G = ((A, B), E)$ with $|A| = |B| = N$ is an (N, k, d)-expander for $d < 1$ then G is a δ-bipartite expander.*

Proof. Consider an arbitrary subset $X \subseteq A$ with $|X| \geq \delta N$ and let $Y = B \setminus \mathsf{N}(X)$. We want to argue that $|Y| < \delta N$ or equivalently $|\mathsf{N}(X)| > (1-\delta)N$. Without loss of generality, we may assume that $|X| < N$ (otherwise we have $\mathsf{N}(X) = B$ since $|\mathsf{N}(X)| \geq (1 + d(1 - |X|/N))|X| = |X| = N$). Since G is an (N, k, d)-expander, we have that $|\mathsf{N}(X)| \geq \left[1 + d\left(1 - \frac{|X|}{N}\right)\right]|X| = -\frac{d}{N}|X|^2 + (d + 1)|X|$. Hence, for $N > |X| \geq \delta N$, we have that

$$\begin{aligned} |\mathsf{N}(X)| &\geq -\frac{d}{N}|X|^2 + (d+1)|X| \\ &> -\frac{d}{N}(\delta N)^2 + (d+1)\delta N \\ &\geq (1 - \delta)N, \end{aligned}$$

where the middle inequality follows from the observation that when $d < 1$, the function $f(x) = -\frac{d}{N}x^2 + (d + 1)x$ is an increasing function over the range $0 \leq x \leq N$ and the last inequality follows from the choice of $\delta = \frac{(d+2)-\sqrt{d^2+4}}{2d}$ since $d \geq \frac{1-2\delta}{\delta-\delta^2}$. Now fixing an arbitrary subset $Y \subseteq B$ with $|Y| \geq \delta N$ and setting $X = A \setminus \mathsf{N}(Y)$, a symmetric argument shows that $|X| < \delta N$. Thus, G is a δ-bipartite expander. ◀

3 Explicit Constructions of δ-Bipartite Expanders

In this section, we give an explicit (locally navigable) construction of a δ-bipartite expander graph for any constant $\delta > 0$. As a building block, we start with an explicit construction of $(N = m^2, k = 5, d = (2 - \sqrt{3})/4)$-expander due to Gabber and Galil [12]. Applying Lemma 4 above this gives us a δ-bipartite expander with $\delta \approx 0.492$ whenever $N = m^2$. To construct depth-robust graphs we need to construct δ-bipartite expanders for much smaller values of δ and for arbitrary values of N, i.e., not just when $N = m^2$ is a perfect square. We overcome the first challenge by layering the $(N = m^2, k, d)$-expanders of [12] to obtain δ-bipartite expanders for arbitrary constants $\delta > 0$ – the indegree increases as δ approaches 0. We overcome the second issues simply by truncating the graph, i.e., if G is a $\delta/2$-bipartite expander with $2N$ nodes then we can discard up to $N/2$ sources and $N/2$ sinks and the remaining graph will still be a δ-expander.

3.1 Truncation

By layering the (N, k, d)-expanders of Gabber and Galil [12] we are able to obtain a family $\{G_{m,\delta}\}_{m=1}^{\infty}$ of δ-bipartite expanders for any constant $\delta > 0$ such that G_m has $N = m^2$ nodes on each side of the bipartition and constant indegree. However, our constructions of depth-robust graphs will require us to obtain a family $\{H_{N,\delta}\}_{N=1}^{\infty}$ of δ-bipartite expanders such that $H_{N,\delta}$ has N nodes on each side of the bipartition and constant indegree. In this section, we show how the family $\{H_{N,\delta}\}_{N=1}^{\infty}$ can be constructed by truncating graphs from the family $\{G_{m,\delta}\}_{m=1}^{\infty}$. Furthermore, if the construction of $G_{m,\delta}$ is explicit and locally navigable then so is $H_{N,\delta}$.

For each N we define $m(N) := \min_{m:m^2 \geq N}$ to be the smallest positive integer m such that $m^2 \geq N$. We first observe that for all integers $N \geq 1$ we have $m(N)^2 \geq N \geq m(N)^2/2$.

▷ Claim 5. For all $N \geq 1$ we have $m(N)^2 \geq N \geq m(N)^2/2$.

Proof. The fact that $m(N)^2 \geq N$ follows immediately from the definition of $m(N)$. For the second part it is equivalent to show that $m(N)^2/N \leq 2$ for all $N \geq 1$. The ratio $m(N)^2/N$ is maximized when $N = (m-1)^2 + 1$ for some $m \geq 1$. Thus, it suffices to show that $\frac{m^2}{(m-1)^2+1} \leq 2$ for all $m \geq 1$ or equivalently $1 + \frac{2(m-1)}{(m-1)^2+1} \leq 2$. The function $f(m) = \frac{2(m-1)}{(m-1)^2+1}$ is maximized at $m = 2$ in which case $f(2) = 1$. For all $m \geq 2$ we have $1 + \frac{2(m-1)}{(m-1)^2+1} \leq 2$ and when $m = 1$ we have $1 + \frac{2(m-1)}{(m-1)^2+1} = 1 \leq 2$ so the claim follows. ◁

Suppose that for any constant $\delta > 0$ we are given an explicit locally navigable family $\{G_{m,\delta}\}_{m=1}^{\infty}$ of δ-bipartite expanders with $G_{m,\delta} = ((A_{m,\delta} = \{X_1, \ldots, X_{m^2}\}, B_{m,\delta} = \{Y_1, \ldots, Y_{m^2}\}), E_{m,\delta})$ with edge set $E_{m,\delta} = \{(X_i, Y_j) : i \in \mathsf{GetParents}(m, \delta, j) \wedge j \leq m^2\}$ defined by an algorithm $\mathsf{GetParents}(m, \delta, j)$. We now define the algorithm $\mathsf{GetParentsTrunc}(N, \delta, j) = \mathsf{GetParents}(m(N), \delta/2, j) \cap \{1, \ldots, N\}$ and we define $H_{m,\delta} = ((A'_{N,\delta} = \{a_1, \ldots, a_N\}, B'_{N,\delta} = \{b_1, \ldots, b_N\}), E'_{N,\delta})$ with edge set $E'_{N,\delta} = \{(a_i, b_j) : i \in \mathsf{GetParentsTrunc}(N, \delta, j) \wedge j \leq N\}$. Intuitively, we start with a $\delta/2$-bipartite expander $G_{m,\delta/2}$ with $N' = m(N)^2$ nodes on each side of the partition and drop $N' - N \leq N'/2$ nodes from each side of the bipartition to obtain $H_{m,\delta}$. Clearly, if $\mathsf{GetParents}$ can be evaluated in time $O(\mathsf{polylog}\, m)$ then $\mathsf{GetParentsTrunc}$ can be evaluated in time $O(\mathsf{polylog}\, N)$. Thus, the family $\{H_{N,\delta}\}_{N=1}^{\infty}$ is explicit and locally navigable. Finally, we claim that $H_{m,\delta}$ is a δ-bipartite expander.

▶ **Lemma 6.** *Assuming that $G_{m,\delta}$ is a δ-bipartite expander for each $m \geq 1$ and $\delta > 0$, the graph $H_{m,\delta}$ is a δ-bipartite expander for each $m \geq 1$ and $\delta > 0$.*

Proof. Consider two sets $X \subseteq \{1,\ldots,N\}$ and $Y \subseteq \{1,\ldots,N\}$ and set $m = m(N)$. If $|X| \geq \delta N$ and $|Y| \geq \delta N$ then by Claim 5 we have $|X| \geq (\delta/2)m^2$ and $|Y| \geq (\delta/2)m^2$. Thus, since $G_{m,\delta/2}$ is a $\delta/2$-bipartite expander and $X, Y \subseteq \{1,\ldots,m^2\}$ there must be some pair $(i,j) \in X \times Y$ with $i \in \mathsf{GetParents}(m, \delta/2, j)$. Since $i \leq N$ we also have $i \in \mathsf{GetParentsTrunc}(N,\delta,j) = [N] \cap \mathsf{GetParents}(m,\delta/2,j)$. Thus, the edge (a_i, b_j) still exists in the truncated graph $H_{m,\delta}$. It follows that $H_{m,\delta}$ is a δ-bipartite expander. ◀

In the remainder of this section, we will focus on constructing $G_{m,\delta}$. In the next subsection, we first review the construction of $(N = m^2, k = 5, d = (2-\sqrt{3})/4)$-expanders due to Gabber and Galil [12].

3.2 Explicit (N, k, d)-Expander Graphs

Let $P_m \doteq \{0,1,\ldots,m-1\} \times \{0,1,\ldots,m-1\}$ be the set of pairs of integers (x,y) with $0 \leq x, y \leq m-1$. We can now define the family of bipartite graphs $G_m = ((A_m, B_m), E_m)$ where $A_m = \{X_{i,j} = (i,j) : (i,j) \in P_m\}$ and $B = \{Y_{i,j} = (i,j) : (i,j) \in P_m\}$. The edge set E_m is defined using the following 5 permuatations on P_m:

$$\sigma_0(x,y) = (x,y),$$
$$\sigma_1(x,y) = (x, x+y),$$
$$\sigma_2(x,y) = (x, x+y+1),$$
$$\sigma_3(x,y) = (x+y, y),$$
$$\sigma_4(x,y) = (x+y+1, y),$$

where the operation $+$ is modulo m. Now we can define the edge set E_m as

$$E_m = \{(X_{i',j'}, Y_{i,j}) : \exists\ 0 \leq k \leq 4 \text{ such that } \sigma_k(i',j') = (i,j)\}.$$

Gabber and Galil [12] proved that the graph G_m is a (N,k,d)-expander with $N = m^2$ nodes on each side of the biparition $(A_m \ / \ B_m)$, $k = 5$, and $d = (2-\sqrt{3})/4$.

It will be convenient to encode nodes using integers between 1 and $N = m^2$ instead of pairs in P_m. define $\mathtt{PairToInt}_m(x,y) = xm + y + 1$, a bijective function mapping pairs $(x,y) \in \{0,1,\ldots,m-1\} \times \{0,1,\ldots,m-1\}$ to integers $\{1,\ldots,m^2\}$ along with the inverse mapping $\mathtt{IntToPair}_m(z) = \left(\lfloor \frac{z-1}{m} \rfloor, (z-1) \mod m\right)$. We can then redefine the permutations over the set $\{1,\ldots,m^2\}$ as follows $\sigma'_j(z) = \mathtt{PairToInt}_m\left(\sigma_j\left(\mathtt{IntToPair}_m(z)\right)\right)$ and we can (equivalently) redefine $G_m = ((A_m, B_m), E_m)$ where $A_m = \{X_1,\ldots,X_{m^2}\}$, $B_m = \{Y_1,\ldots,Y_{m^2}\}$ and $E_m = \{(X_i, Y_j) \ : \ 1 \leq j \leq m^2 \wedge i \in \mathsf{GetParentsGG}(m,j)\}$. Here, $\mathsf{GetParentsGG}(m,j) = \{\sigma'_0(j), \sigma'_1(j), \sigma'_2(j), \sigma'_3(j), \sigma'_4(j)\}$.

3.3 Amplification via Layering

Given that we have constructed explicit δ-bipartite expanders with constant indegree for a fixed $\delta > 0$, we will construct explicit δ-bipartite expanders with constant indegree for any arbitrarily small $\delta > 0$. The construction is recursive. As our base case we define $G^0_m = G_m = ((A_m, B_m), E_m)$ where $A_m = \{X_1,\ldots,X_{m^2}\}$, $B_m = \{Y_1,\ldots,Y_{m^2}\}$ and $E_m = \{(X_i, Y_j) \ : \ 1 \leq j \leq m^2 \wedge i \in \mathsf{GetParentsGG}(m,j)\}$ as the $(N = m^2, k = 5, d = (2-\sqrt{3})/4)$-expander of Gabber and Galil [12] and we define $\mathsf{GetParentsLayered}^1(m,j) = \mathsf{GetParentsGG}(m,j)$. We can then define $G^{i+1}_m = ((A_m, B_m), E^{i+1}_m)$ where $A_m = \{X_1,\ldots,X_{m^2}\}, B_m = \{Y_1,\ldots,Y_{m^2}\}$ and $E^{i+1}_m = \{(X_i, Y_j) \ : \ 1 \leq j \leq m^2 \wedge i \in \mathsf{GetParentsLayered}^{i+1}(m,j)\}$ where

$\mathsf{GetParentsLayered}^{i+1}(m,j) = \bigcup_{j' \in \mathsf{GetParentsGG}(m,j)} \mathsf{GetParentsLayered}^{i}(m,j')$. Intuitively, we can form the graph G_m^i by stacking i copies of the graph G_m and forming a new bipartite graph by collapsing all of the intermediate layers. See Figure 1 for an illustration.

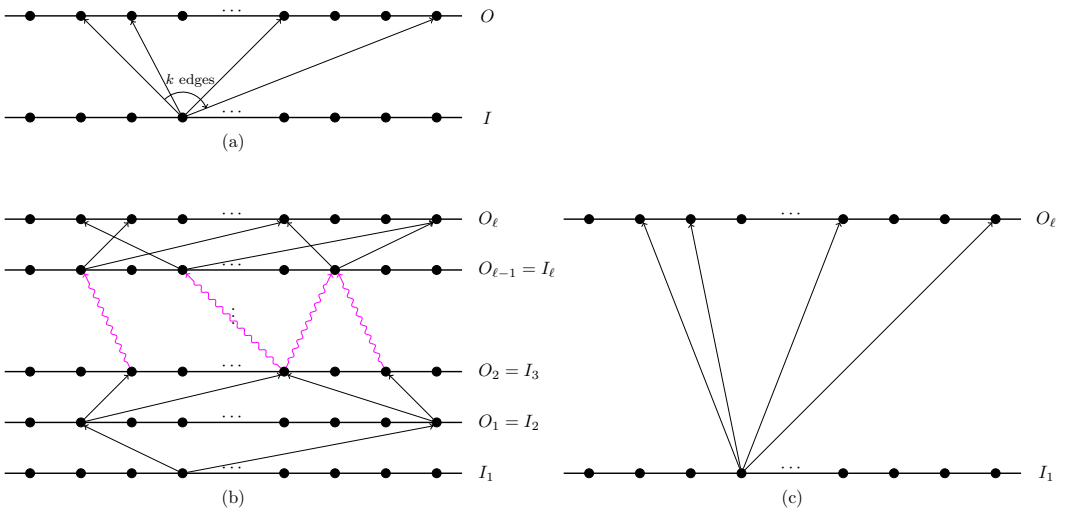

Figure 1 (a) One copy of an (N,k,d)-expander. Here, we remark that each input node has exactly k edges such that the total number of edges is kN. (b) Stack the graph ℓ times to get a graph with $(\ell+1)$ layers. The snaked edges from the third to ℓ^{th} layer indicates that there are connected paths between the nodes. (c) Generate a new bipartite graph by collapsing all of the intermediate layers. A node u on the bottom layer I_1 has an edge to a node v on the top layer O_ℓ if and only if there is a path in the original graph.

We note that $\left|\mathsf{GetParentsLayered}^{i+1}(m,j)\right| \leq k \times \left|\mathsf{GetParentsLayered}^{i}(m,j)\right| \leq k^{i+1}$. Theorem 7 tells us that amplification by layering yields a δ-bipartite expander. In particular, there is a constant L_δ such that G_m^i is a δ-bipartite expander whenever $i \geq L_\delta$. By our previous observation this graph has indegree at most k^{L_δ} which is a constant since k and L_δ are both constants.

▶ **Theorem 7.** *For any constant $\delta > 0$, there exists a constant L_δ such that for any $i \geq L_\delta$ the graph G_m^i is a δ-bipartite expander with $N = m^2$ nodes on each side of the partition.*

Proof. Fix any subset $Y^0 \subseteq [N]$ of size $|Y^0| \geq \delta N$. Let $Y^1 \doteq \bigcup_{j \in Y^0} \mathsf{GetParentsGG}(m,j)$, and recursively define $Y^{i+1} \doteq \bigcup_{j \in Y^i} \mathsf{GetParentsGG}(m,j)$. Since $Y^i = \bigcup_{j \in Y^0} \mathsf{GetParentsLayered}^{i}(m,j)$, it suffices to argue that $|Y^i| > (1-\delta)N$ whenever $i \geq L_\delta \doteq \left\lceil \frac{\log((1-\delta)/\delta)}{\log(1+d\delta)} \right\rceil + 1$. To see this, we note that for each $i \geq 0$, either

(1) $|Y^i|$ has already reached the target size $(1-\delta)N$, or

(2) $|Y^{i+1}| \geq \left[1 + d\left(1 - \frac{|Y^i|}{N}\right)\right] |Y^i| \geq (1+d\delta)|Y^i|$ since $\mathsf{GetParentsGG}$ defines an (N,k,d)-expander.

It follows that $|Y^{i+1}| \geq \min\{(1-\delta)N, (1+d\delta)^i \delta N\}$. Now we want to find i such that $(1+d\delta)^i \delta N = (1-\delta)N$; solving the equation we have $i = \frac{\log((1-\delta)/\delta)}{\log(1+d\delta)}$. Thus, for $i = L_\delta - 1$ we have $|Y^i| \geq (1-\delta)N$ and for $i \geq L_\delta$ we have $|Y^i| > (1-\delta)N$. Thus, for $i \geq L_\delta$ the graph G_m^i is a δ-bipartite expander, i.e., for any subsets $X, Y \subseteq [N]$ of size $|X| \geq \delta N = \delta m^2$ we must have $\left|X \cap \bigcup_{j \in Y} \mathsf{GetParentsLayered}^{i}(m,j)\right| > 0$ as long as $i \geq L_\delta$. ◀

3.4 Final Construction of δ-Bipartite Expanders

Based on the proof of Theorem 7, we can define $L_\delta \doteq \left\lceil \frac{\log((1-\delta)/\delta)}{\log(1+d\delta)} \right\rceil + 1$, $G_{m,\delta} \doteq G_m^{L_\delta}$, and obtain $H_{N,\delta}$ by truncating the graph $G_{m(N),\delta/2}$. The edges are defined by the procedure $\mathsf{GetParentsBE}(N,\delta,j) \doteq [N] \cap \mathsf{GetParentsLayered}^{L_{\delta/2}}(m(N),j)$ – the procedure GetParentsBE is short for "Get Parents Bipartite Expander". Formally, we have $H_{N,\delta} = ((A_N = \{a_1,\ldots,a_N\}, B_N = \{b_1,\ldots,b_N\}), E_{N,\delta})$ where $E_{N,\delta} = \{(a_i,b_j) : i \in \mathsf{GetParentsBE}(N,\delta,j)\}$.

▶ **Corollary 8.** *Fix any constant $\delta > 0$ and define $L_\delta = \left\lceil \frac{\log((1-\delta)/\delta)}{\log(1+d\delta)} \right\rceil + 1$. The graph $G_m^{L_\delta}$ is a δ-bipartite expander and the graph $H_{N,\delta}$ is a δ-bipartite expander for any integers $m, N \geq 1$.*

Proof. By Theorem 7 $G_m^{L_\delta}$ is a δ-bipartite expander. To see that $H_{N,\delta}$ is a δ-bipartite expander we simply note that $G_{m(N),\delta/2}$ is a $\delta/2$-bipartite expander and apply Lemma 6. ◀

4 Explicit Constructions of Depth Robust Graphs

We are now ready to present our explicit construction of a depth-robust graph. For any $N = 2^n$ we define the graph $G(\delta,N) = ([N], E(\delta,N))$ with edge set $E(\delta,N) = \{(u,v) : v \in [N] \wedge u \in \mathsf{GetParentsEGS}(\delta,v,N)\}$. The procedure $\mathsf{GetParentsEGS}(\delta,v,N)$ to compute the edges of $G(\delta,N)$ relies on the procedure GetParentsBE which computes the edges of our underlying bipartite expander graphs. We remark that our construction is virtually identical to the construction of [10] except that the underlying bipartite expanders are replaced with our explicit constructions from the last section.

Algorithm 1 $\mathsf{GetParentsEGS}(\delta,v,N)$.

```
1: procedure GetParentsEGS(δ, v, N)
2:     P = {v − 4n, ..., v − 1}
3:     for t = 1 to ⌈log_2 v⌉ do
4:         m = ⌊v/2^t⌋
5:         x = v mod 2^t
6:         B = GetParentsBE(2^t, L_{δ/5}, x + 1)
7:         for y ∈ B do
8:             P = P ∪ {(m − i)2^t + y : 1 ≤ i ≤ min{m, 10}}
9:     return P ∩ {1, ..., N}
```

Note that for any constant $\delta > 0$ and any integer $n \geq 1$, the graph $G(\delta,N)$ defined by $\mathsf{GetParentsEGS}(\delta,\cdot,N)$ has $N = 2^n$ nodes and maximum indeg $\mathsf{indeg}(G(\delta,N)) = O(n) = O(\log N)$.

Erdös, Graham, and Szemeredi [10] showed that the graph $G(\delta,N)$ is a δ-local expander as long as the underlying bipartite graphs are $\delta/5$-bipartite expanders.

▶ **Theorem 9** ([10]). *For any $\delta > 0$ the graph $G(\delta,N)$ is a δ-local expander.*

Theorem 10 says that any δ-local expander is also $(e, d = N - e\frac{1+\gamma}{1-\gamma})$-depth-robust for any constant $\gamma > 2\delta$. The statement of Theorem 10 is implicit in the analysis of Alwen et al. [4]. We include the proof for completeness.

▶ **Theorem 10.** *Let $0 < \delta < 1/4$ be a constant and let $\gamma > 2\delta$. Any δ-local expander on N nodes is $(e, d = N - e\frac{1+\gamma}{1-\gamma})$-depth-robust for any $e \leq N$.*

Proof. Let G be a δ-local expander with $\delta < 1/4$ and $\gamma > 2\delta$ and let $S \subseteq [N]$ denote an arbitrary subset of size $|S| = e$. To show that $G - S$ has a path of length $d = N - e\frac{1+\gamma}{1-\gamma}$ we rely on two lemmas (Lemma 11, Lemma 12) due to Alwen et al. [4]. We first introduce the notion of a γ-good node. A node $x \in [N]$ is γ-good under a subset $S \subseteq [N]$ if for all $r > 0$ we have $|I_r(x) \backslash S| \geq \gamma |I_r(x)|$ and $|I_r^*(x) \backslash S| \geq \gamma |I_r^*(x)|$, where $I_r(x) = \{x - r - 1, ..., x\}$ and $I_r^*(x) = \{x + 1, ..., x + r\}$.

▶ **Lemma 11** ([4, 10]). *Let $G = (V = [N], E)$ be a δ-local expander and let $x < y \in [N]$ both be γ-good under $S \subseteq [N]$ then if $\delta < \min(\gamma/2, 1/4)$ then there is a directed path from node x to node y in $G - S$.*

▶ **Lemma 12** ([4]). *For any DAG $G = ([N], E)$ and any subset $S \subseteq [N]$ of nodes at least $N - |S|\frac{1+\gamma}{1-\gamma}$ of the remaining nodes in G are γ-good with respect to S.*

Applying Lemma 12 at least $d = N - e\frac{1+\gamma}{1-\gamma}$ nodes $v_1, \ldots, v_d$ are γ-good with respect to S. Without loss of generality, we can assume that $v_1 < v_2 < \ldots < v_d$. Applying Lemma 11 for each $i \leq d$, there is a directed path from v_i to v_{i+1} in $G - S$. Concatenating all of these paths we obtain one long directed path containing all of the nodes $v_1, \ldots, v_d$. Thus, $G - S$ contains a directed path of length $d = N - e\frac{1+\gamma}{1-\gamma}$. ◀

As an immediate corollary of Theorem 9 and Theorem 10 we have

▶ **Corollary 13.** *Let $0 < \delta < 1/4$ be a constant and let $\gamma > 2\delta$ then the graph $G(\delta, N)$ is $(e, d = N - e\frac{1+\gamma}{1-\gamma})$-depth-robust for any $e \leq N$.*

4.1 Explicit Extreme Depth-Robust Graphs

We also obtain explicit constructions of ϵ-extreme depth-robust graphs which have found applications in constructing Proofs of Space and Replication [15], Proofs of Sequential Work [14], and in constructions of Memory-Hard Functions [4].

▶ **Definition 14** ([4]). *For any constant $\epsilon > 0$, a DAG G with N nodes is* ϵ-extreme depth-robust *if and only if G is (e, d)-depth-robust for any $e + d \leq (1 - \epsilon)N$.*

When we set δ_ϵ appropriately the graph $G(\delta_\epsilon, N = 2^n)$ is ϵ-extremely depth robust.

▶ **Corollary 15.** *Given any constant $\epsilon > 0$ we define δ_ϵ to be the unique value such that $1 + \epsilon = \frac{1+2.1\delta_\epsilon}{1-2.1\delta_\epsilon}$ if $\epsilon \leq 1/3$ and $\delta_\epsilon = \delta_{1/3}$ for $\epsilon > 1/3$. For any integer $n \geq 1$ the graph $G(\delta_\epsilon, N = 2^n)$ is ϵ-extreme depth robust.*

Proof. Set $\gamma = 2.1\delta_\epsilon$ and observe that $\delta_{1/3} \leq 0.07 \leq 1/4$ and for $\epsilon < 1/3$ we have $\delta_\epsilon \leq \delta_{1/3} \leq 1/4$ so we can apply Corollary 13 to see that $G(\delta_\epsilon, N = 2^n)$ is $(e, d = N - e\frac{1+2.1\delta_\epsilon}{1-2.1\delta_\epsilon})$-depth robust for any $e \leq N$. Since $\frac{1+2.1\delta_\epsilon}{1-2.1\delta_\epsilon} = (1 + \epsilon)$ it follows that the graph is ϵ-extreme depth robust. ◀

4.2 Depth-Robust Graphs with Constant Indegree

In some applications it is desirable to ensure that our depth-robust graphs have constant indegree. We observe that we can apply a result of Alwen et al. [3] to transform the DAG $G(\delta, N) = (V = [N], E(\delta, N))$ with maximum indegree $\beta = \beta_{\delta,N}$ into a new DAG $H_{\delta,N} = ([N] \times [\beta], E'(\delta, N))$ with $N' = 2N\beta$ nodes and maximum indegree 2. Intuitively, the transformation reduces the indegree by replacing every node $v \in [N]$ from $G(\delta, N)$ with a path of 2β nodes $(v, 1), \ldots, (v, 2\beta)$ and distributing the incoming edges accross this path. In

particular, if v has incoming edges from nodes $v_1, \ldots, v_\beta$ in $G(\delta, N)$ then for each $i \leq \beta$ we will add an edge from the node $(v_i, 2\beta)$ to the node (v, i). This ensures that each node (v, i) has at most two incoming edges. Formally, the algorithm $\mathsf{GetParentsLowIndeg}(\delta, v', N)$ takes as input a node $v' = (v, i)$ and (1) initializes $P' = \{(v, i-1)\}$ if $i > 1$, $P' = \{(v-1, 2\beta)\}$ if $i = 1$ and $v > 1$ and $P' = \{\}$ otherwise, (2) computes $P = \mathsf{GetParentsEGS}(\delta, v, N)$, (3) sets $u = P[i]$ to be the ith node in the set P, and (4) returns $P' \cup \{(u, 2\beta)\}$. It is easy to verify that the algorithm $\mathsf{GetParentsLowIndeg}$ runs in time $\mathsf{polylog}\, N$.

▶ **Corollary 16.** *Let $0 < \delta < 1/4$ be a constant and let $\gamma > 2\delta$ then the graph $H_{\delta,N}$ is $(e, d = N\beta - e\beta\frac{1+\gamma}{1-\gamma})$ depth-robust for any $e \leq N$.*

Proof. (Sketch) Alwen et al. [3] showed that applying the indegree reduction procedure above to any (e, d)-depth-robust graph with maximum indegree β yields a $(e, d\beta)$-depth-robust graph. The claim now follows directly from Theorem 9 and Theorem 10. ◀

5 Conclusion

We give the first explicit construction of ϵ-extreme depth-robust graphs $G = (V = [N], E)$ with indegree $O(\log N)$ which are locally navigable. Applying an indegree reduction gadget of Alwen et al. [3] we also obtain the first explicit and locally navigable construction of $(\Omega(N/\log N), \Omega(N))$-depth-robust graphs with constant indegree. Our current constructions are primarily of theoretical interest and we stress that we make no claims about the practicality of the constructions as the constants hidden by the asymptotic notation are large. Finding explicit and locally navigable constructions of $(c_1 N/\log N, c_2 N)$-depth-robust graphs with small indegree for reasonably large constants $c_1, c_2 > 0$ is an interesting and open research challenge. Similarly, finding explicit and locally navigable constructions of ϵ-extreme depth-robust graphs $G = (V = [N], E)$ with indegree $c_\epsilon \log N$ for smaller constants c_ϵ remains an important open challenge.

References

1. Joël Alwen and Jeremiah Blocki. Efficiently computing data-independent memory-hard functions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 241–271. Springer, Heidelberg, 2016. doi:10.1007/978-3-662-53008-5_9.
2. Joël Alwen, Jeremiah Blocki, and Ben Harsha. Practical graphs for optimal side-channel resistant memory-hard functions. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1001–1017. ACM Press, October / November 2017. doi:10.1145/3133956.3134031.
3. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Depth-robust graphs and their cumulative memory complexity. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 3–32. Springer, Heidelberg, April / May 2017. doi:10.1007/978-3-319-56617-7_1.
4. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Sustained space complexity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 99–130. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78375-8_4.
5. Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 595–603. ACM Press, 2015. doi:10.1145/2746539.2746622.

6 Mohammad Hassan Ameri, Jeremiah Blocki, and Samson Zhou. Computationally data-independent memory hard functions. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 36:1–36:28. LIPIcs, 2020. doi:10.4230/LIPIcs.ITCS.2020.36.

7 Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. *IEEE Transactions on Information Theory*, 67(7):4338–4360, 2021. doi:10.1109/TIT.2021.3076396.

8 Jeremiah Blocki and Samson Zhou. On the computational complexity of minimal cumulative cost graph pebbling. In Sarah Meiklejohn and Kazue Sako, editors, *FC 2018*, volume 10957 of *LNCS*, pages 329–346. Springer, Heidelberg, February / March 2018. doi:10.1007/978-3-662-58387-6_18.

9 Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 585–605. Springer, Heidelberg, 2015. doi:10.1007/978-3-662-48000-7_29.

10 P. Erdös, R.L. Graham, and E. Szemerédi. On sparse graphs with dense long paths. *Computers & Mathematics with Applications*, 1(3):365–369, 1975. doi:10.1016/0898-1221(75)90037-1.

11 Ben Fisch. Tight proofs of space and replication. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 324–348. Springer, Heidelberg, 2019. doi:10.1007/978-3-030-17656-3_12.

12 Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

13 Aoxuan Li. On explicit depth robust graphs. *UCLA*, ProQuest ID: Li_ucla_0031N_-17780. Merritt ID: ark:/13030/m5130rq7, 2019. URL: https://escholarship.org/uc/item/4fx1m6dh.

14 Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan. Publicly verifiable proofs of sequential work. In Robert D. Kleinberg, editor, *ITCS 2013*, pages 373–388. ACM, 2013. doi:10.1145/2422436.2422479.

15 Krzysztof Pietrzak. Proofs of catalytic space. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 59:1–59:25. LIPIcs, 2019. doi:10.4230/LIPIcs.ITCS.2019.59.

16 Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *41st FOCS*, pages 3–13. IEEE Computer Society Press, 2000. doi:10.1109/SFCS.2000.892006.

17 Georg Schnitger. On depth-reduction and grates. In *24th FOCS*, pages 323–328. IEEE Computer Society Press, 1983. doi:10.1109/SFCS.1983.38.

18 Leslie Valiant. Graph-theoretic arguments in low-level complexity. In *International Symposium on Mathematical Foundations of Computer Science*, pages 162–176. Springer, 1977.