Data-Dependent Memory-Hard Functions: Sustained Space and Cumulative Complexity Trade-offs in the Parallel Random Oracle Model

Jeremiah Blocki and Blake Holman

Purdue University

Abstract. Memory-Hard Functions (MHFs) are a cryptographic primitive designed to protect passwords and other low-entropy secrets against brute-force attacks. The strongest and most natural formalization of memory-hardness is sustained space complexity (SSC), which measures how long an evaluator's memory remains above a given threshold. Ideally, one would like to ensure that any parallel evaluator must sustain $\Theta(N)$ memory for $\Theta(N)$ steps, while the function can also be computed in sequential time $\Theta(N)$. Unfortunately, this goal is impossible to achieve. Thus, the appropriate objective is to establish strong tradeoffs between sustained space complexity and cumulative memory complexity (CMC), another strong notion of memory hardness. Blocki and Holman (CRYPTO 2022) achieved strong SSC/CMC tradeoffs in the dynamic pebbling model, but their construction relied on expensive combinatorial graphs, and the pebbling abstraction does not rule out more efficient attacks in the stronger Parallel Random Oracle Model (PROM).
We address both limitations. We construct a new data-dependent MHF (dMHF), DEGSample, and prove the first sustained-space versus cumulative-memory tradeoff directly in the PROM. In the dynamic pebbling model, DEGSample achieves the same ideal tradeoff as prior work: any dynamic pebbling strategy either sustains $\Omega(N)$ memory for $\Omega(N)$ steps or incurs a maximal CMC penalty $\Omega(N^{3-\epsilon})$. In the PROM, we prove that any evaluator either sustains $\Omega(N)$ memory for $\Omega(N)$ steps or incurs a steep CMC penalty of at least $\Omega(N^{2.5-\epsilon})$. Our analysis introduces a new graph property, ancestral robustness, and shows that, together with fractional depth-robustness, it suffices to obtain strong PROM tradeoffs via a natural dynamization procedure to turn the graph into a dMHF. The PROM lower bound combines a low-space time–space argument with an extraction procedure that converts any PROM execution into a cost-equivalent pebbling of the realized graph.

Keywords: Memory-Hard Functions · Parallel Random Oracle Model · Sustained-Space Complexity · Cumulative Memory Complexity

Table of Contents

1 Introduction

Memory-Hard Functions (MHFs) are an important cryptographic primitive which can be used to protect low-entropy secrets, such as user passwords, from brute-force attacks [Per09,BHZ18,BDK16]. They have also been used to construct egalitarian proofs-of-work [Lee11,BK16a,BK16b,BS25] which resist attacks using customized hardware such as Field Programmable Gate Arrays (FPGAs) or Application Specific Integrated Circuits (ASICs). The strongest and most natural formalization of memory hardness is sustained space complexity (SSC) [ABP18]. Ideally, we would like for a MHF to have the property that (1) an honest party can evaluate the function in sequential time $O(N)$ on a personal computer, while (2) any parallel algorithm that evaluates the function is forced to lock up $\Omega(N)$ memory for $\Omega(N)$ steps. Intuitively, this would ensure that the evaluation costs for a MHF will be dominated by memory costs which tend to be egalitarian across different architectures making it difficult for an parallel attacker to exploit customized hardware (e.g., ASICs) to reduce the amortized cost of evaluating the function across multiple different inputs (e.g., password guesses). Such a rigorous Memory-Hardness guarantee can help to ensure that a brute-force attack is economically infeasible [BHZ18].

Unfortunately, this ideal sustained-space requirement is impossible to achieve. In particular, if an MHF can be evaluated in sequential time $O(N)$, then there always exists an alternative evaluation strategy that uses at most $O(N/\log N)$ space [HPV77,BH22]. While this attack significantly reduces the peak memory usage, it requires exponential running time and is therefore impractical. This fundamental limitation shows that absolute sustained-space lower bounds are unattainable. Consequently, a more realistic goal is to establish strong ***trade-offs***: any algorithm that does not achieve near-maximal sustained space complexity (i.e., that does not lock up $\Omega(N)$ memory for $\Omega(N)$ rounds) must incur a substantial cost penalty elsewhere. The objective is not to rule out all low-space strategies, but to ensure that any such strategy becomes economically infeasible in practice.

1.1 Background: Memory-Hard Functions

As research on memory-hard functions has progressed, several increasingly strict measures of memory-hardness have been introduced: space-time complexity (ST), cumulative memory complexity (CMC) [AS15], and sustained space complexity (SSC) [ABP18]. Intuitively, space-time complexity aims to capture the "full cost" [Wie04] of an attack by measuring the space-time product i.e., total memory resources required to run the attack multiplied by the duration of time those resources are required. However, as Alwen and Serbinenko observed a function with high space-time complexity may not necessarily have high amortized space-time complexity [AS15]. In particular, the space-time costs for a parallel attacker to evaluate an MHF f on m distinct inputs is not necessarily m times the space-time complexity of f for a single input. As it turned

this observation was not merely theoretical and many prominent MHF candidates have been shown vulnerable to amortization attacks which allow the attacker to evaluate the function on multiple inputs at a greatly reduced cost e.g., see [AB16,AB17,ABP17,BZ17,AGK+18]. These results included amortization attacks against Argon2i, the winner of the password hashing competition [HPM15,pas15].

The notion of cumulative memory complexity (CMC) [AS15] was introduced as a tool to quantify the amortized space-time complexity of a MHF. In the context of password hashing it is crucial for the password hashing algorithm to have high amortized space-time complexity because a brute-force attacker will need to evaluate the function on many different inputs (password guesses). Intuitively, cumulative memory complexity measures the product of the ***average*** space usage multiplied by the total running time while space-time complexity considers the ***maximum*** space usage. A bit more formally suppose we run an algorithm A on an input x and the algorithm terminates after t steps. Let σ_i denote the state of our program at time i, and the sequence $\sigma_1, \ldots, \sigma_t$ denotes the execution trace of A on input x. The space-time complexity of this execution trace is $t \cdot \max_{1 \leq i \leq t} |\sigma_i|$ and the cumulative memory cost is $\sum_{i=1}^{t} |\sigma_i|$. The s-sustained space complexity of the trace is $\left|\left\{i \ : \ |\sigma_i| \geq s\right\}\right|$ i.e., the number of rounds in which the space usage exceeds our threshold s. Intuitively, a function f has high space-time complexity if, for any parallel algorithm $A(x)$ evaluating $f(x)$, the execution trace of A has high cumulative memory cost. We can similarly define the space-time complexity or s-sustained space complexity for f.

Over the past decade there has been a great deal of progress towards constructing MHFs with provably high CMC e.g., [ABP17,ABH17,BHK+19,ACP+17,CT19]. However, while high CMC rules out amortization attacks it does not capture all of the properties we might desire of an MHF. For example, Scrypt [Per09] is a memory-hard function which can be evaluated in $O(N)$ sequential steps, and was proven to have maximum[1] possible CMC $\Omega(N^2)$ in the Parallel Random Oracle Model (PROM) [ACP+17]. However, for every space parameter $s \geq 2$, Scrypt can be evaluated using at most s space and N^2/s time. This is against our initial intuition for MHFs; we want any evaluation algorithm to necessarily lock up a large amount of memory for a large amount of time. Indeed, while Scrypt was intended to be ASIC resistant there are commercial ASIC Scrypt miners[scr25] for Scrypt based cryptocurrencies such as Litecoin [Ree17] or Dogecoin.

This motivated Alwen et al. [ABP18] to introduce the stronger goal of developing a MHF with high sustained space-complexity (SSC). Intuitively, the requirement that a function f have high sustained space-complexity is that any evaluation algorithm evaluating f needs to have many steps where the memory usage is high. The s-sustained space complexity of an execution trace measures

[1] $\Omega(N^2)$ is the best possible lower bound for the cumulative memory complexity of an MHF because we require that the honest evaluates the function in $O(N)$ sequential steps. Therefore, the honest evaluation algorithm has cumulative memory cost at most $O(N^2)$ as one can fill at most $O(N)$ memory blocks in sequential time $O(N)$.

the number of steps in which the space usage is at least s i.e., $\left|\left\{i \in [t] : |\sigma_i| \geq s\right\}\right|$. The goal of designing an MHF with high SSC aligns well with our intuitive goal for an MHF that any evaluation algorithm sustains $\Omega(N)$ space for N steps. Observe that any execution trace with s-SCC equal to $t' \leq t$ has CMC at least $\sum_i |\sigma_i| \geq s \cdot t'$. Thus, the requirement that a MHF have high SSC is even stronger than the requirement that a MHF have high CMC.

Data-Dependent vs Data Independent MHFs Broadly speaking there are two types of MHFs: data-independent memory-hard functions (iMHFs) and data-dependent memory-hard functions (dMHFs). iMHFs offer natural resistance to many side-channel attacks because the memory access pattern is ***independent*** of the initial input. On the one hand this side-channel resistance makes iMHFs attractive for applications like password hashing where we would like to protect sensitive inputs (passwords) against side-channel attacks. On the other hand it is possible to design dMHFs with stronger memory-hardness guarantees. For example, the dMHF Scrypt has cumulative memory complexity $\Omega(N^2)$ [ACP+17] while any iMHF has CMC at most $O(N^2 \log \log N / \log N)$ [AB16]. In response to pebbling attacks Argon2id now recommends using a hybrid "id" modes for password hashing where computation begins in data-independent mode or the first $N/2$ rounds of computation before switching over to a data-dependent mode[BDK16]. The idea is that a side-channel attack will ***at worst*** reduce security to that of the underlying iMHF and, in the optimistic scenario where there are no side-channel attacks, hybrid MHF modes can achieve stronger memory hardness guarantees than iMHFs.

Sustained Space Complexity for iMHFs Alwen et al. [ABP18] gave a theoretical construction of an iMHF in which any evaluation algorithm must sustain $\Omega(N/\log N)$ space for $\Omega(N)$ steps, which is asymptotically optimal (in the space parameter) due to Hopcroft's result [HPV77] mentioned above. While the construction of [ABP17] is theoretical, there has been work on developing practical iMHFs with strong SSC/CMC trade-offs. In particular, Blocki et al. [BHK+19] constructed a *practical* iMHF in which any evaluation algorithm either sustains $\Omega(N/\log N)$ space for $\Omega(N)$ steps, or has CMC $\omega(N^2)$.

Recall that our ideal goal is design an iMHF which provides the following SSC/CMC trade-off: any evaluation algorithm either (1) sustains $s = \Omega(N)$ space for $\Omega(N)$ steps, or (2) incurs a super-quadratic CMC penalty $\omega(N^2)$. Unfortunately, it is impossible for any iMHF to satisfy this strong SSC/CMC trade-off. Alwen and Blocki [AB16] gave a general pebbling attack which demonstrated that any iMHF has CMC at most $O(N^2 \log \log N / \log N)$. Because this attack has CMC at most $O(N^2 \log \log N / \log N)$, for any space parameter s there can be at most $O(N^2 \log \log N / (s \log N))$ rounds where the space usage of the attack exceeds s. In particular, for any space parameter $s = \omega\left((N \log \log N)/\log N\right)$ the pebbling attack of [AB16] simultaneously has (1) s-sustained space complexity $o(N)$, and (2) has sub-quadratic CMC at most $O(N^2 \log \log N / \log N)$.

Sustained Space Complexity for dMHFs Observing that the generic pebbling attacks of Alwen and Blocki [AB16] only apply to iMHFs, Blocki and Holman [BH22] investigated SCC/CMC trade-offs for *dMHFs* using the dynamic pebbling game. They gave a theoretical construction of a dMHF with strong SSC/CMC tradeoffs in the dynamic pebbling model. In particular, for any constant $\varepsilon > 0$, they proved that any dynamic pebbling strategy will (whp) either sustain $\Omega(N)$ space for $\Omega(N)$ steps, or has CMC $\Omega(N^{3-\varepsilon})$. The CMC penalty $\Omega(N^{3-\varepsilon})$ for dynamic pebbling strategies with lower sustained space complexity is essentially maximal. In particular, for any constant $c > 0$, any dMHF can be evaluated using ***at most*** cN space while incurring CMC cost ***at most*** $O(N^3)$ [BH22,LT79].

1.2 SSC/CMC Tradeoffs

As noted earlier, the ideal goal of designing an MHF that forces ***every*** evaluation algorithm to sustain $\Omega(N)$ space for $\Omega(N)$ steps is unattainable. If an MHF can be evaluated in sequential time $O(N)$, then there always exists an alternative evaluation strategy that uses space at most $s = O\left(\frac{N}{\log N}\right)$ [HPV77,BH22]. Although this alternative strategy requires exponential time, its s-sustained space complexity is zero for any $s = \omega(N/\log N)$ because its memory usage never exceeds s.

One way to circumvent this barrier is to consider trade-offs between sustained space complexity (SSC) and cumulative memory complexity (CMC). For example, the classical pebbling attacks of Hopcroft et al. [HPV77] have zero $s = \Omega(N)$-sustained space complexity, but they incur exponential cumulative memory cost. This observation suggests a natural relaxation of our ideal goal: instead of requiring maximal sustained space unconditionally, we can require that any algorithm failing to sustain $\Omega(N)$ memory for $\Omega(N)$ rounds must incur a steep cumulative memory penalty.

Accordingly, we seek an MHF that satisfies the following SSC/CMC tradeoff: any evaluation algorithm either (1) sustains $s = \Omega(N)$ space for $\Omega(N)$ steps, or (2) incurs a superquadratic cumulative memory cost $\omega(N^2)$, asymptotically exceeding the $\Theta(N^2)$ CMC of the honest sequential evaluation algorithm.

This goal is impossible for data-independent MHFs (iMHFs). In particular, any iMHF has CMC at most $O(N^2/\log N)$ [AB16], and therefore any s-sustained space lower bound t must satisfy $st = O(N^2/\log N)$. Alwen et al. [ABP18] gave a theoretical construction achieving $s = \Omega(N/\log N)$ sustained space for $\Omega(N)$ steps. While asymptotically optimal for iMHFs, this still falls short of the stronger goal of achieving sustained space when $s = \Omega(N)$.

For data-dependent MHFs (dMHFs), stronger trade-offs become possible. Nevertheless, even in this setting there are inherent limits: the strongest possible CMC penalty for algorithms that do not sustain $\Omega(N)$ space for $\Omega(N)$ steps is $\tilde{O}(N^3)$ [LT79,BH22], since there always exists an algorithm that evaluates a dMHF in time $\tilde{O}(N^2)$ using space at most $O(N/\log\log\log N)$. Despite this upper

bound, proving a penalty $c(N) \in \omega(N^2)$ remains highly meaningful: it guarantees that any deviation from maximal sustained space incurs cumulative cost asymptotically larger than the honest evaluation's $\Theta(N^2)$ cost, thereby making such attacks economically unattractive in practice.

Limitations of [BH22] At first glance, the work of Blocki and Holman [BH22] appears to nearly resolve the study of SSC/CMC trade-offs for dMHFs by establishing an essentially maximal CMC penalty of $\Omega(N^{3-\varepsilon})$ for any dynamic pebbling strategy that does not achieve $s = \Omega(N)$-sustained space complexity $\Omega(N)$. However, their results have two important limitations.

First, their asymptotically optimal construction is not suitable for practical deployment. It relies on expensive combinatorial objects such as ST-robust graphs [BC21]. To the best of our knowledge, there is only one known construction of ST-robust graphs [BC21], and this construction incorporates over a thousand copies of other moderately expensive combinatorial primitives, including superconcentrators [Pip77] and Grates [Sch83]. While [BH22] also proposed a more practical dMHF called Dynamic DRSample, its SSC/CMC trade-offs are substantially weaker. Specifically, they proved a CMC penalty only when the adversary fails to sustain $s = \Omega(N/\log N)$ space for $\Omega(N)$ rounds. In particular, for space parameters $s \in [\Omega(N/\log N), o(N)]$ (for example, $s = N \log\log N/\log N$), no superquadratic CMC penalty was established—even for strategies whose maximum space usage is s.

Second, and more fundamentally, all SSC/CMC guarantees in [BH22] are established solely within the dynamic pebbling model. While this model provides valuable intuition, it lacks a rigorous reduction to the Parallel Random Oracle Model (PROM). Consequently, dynamic pebbling lower bounds do not formally rule out the possibility of more efficient attacks in the PROM.

Pebbling Reductions for MHFs: Alwen and Serbinenko [AS15] provided a rigorous justification for using the parallel black pebbling game to analyze data-independent MHFs (iMHFs). In particular, their pebbling reduction shows that, in the Parallel Random Oracle Model (PROM)[2], the cumulative memory complexity of an iMHF f_G^H is tightly characterized by the cumulative pebbling cost of its underlying directed acyclic graph G. Moreover, analogous reductions show that stronger measures—such as sustained space complexity (and the closely related notion of bandwidth-hardness [RD17])—can also be characterized via pebbling on G [ABP18,BRZ18,BLRZ24]. As a result, a substantial body of work has used pebbling games to design and analyze iMHFs; see, e.g., [AB16,ABP17,ABH17,AB17,BLRZ24,BHK+19,RD17,AGK+18].

Unfortunately, no comparable reduction is known for data-dependent MHFs (dMHFs): there is currently no theorem justifying the dynamic pebbling game as a sound abstraction for PROM attackers. Indeed, Alwen et al. [ACP+17] identified concrete barriers to such a reduction by exhibiting a graph-labeling game in which the best dynamic pebbling strategy is outperformed by a simple PROM algorithm. More broadly, obtaining rigorous PROM lower bounds for dMHFs is

[2] Chen and Tessaro [CT19] extended the pebbling reduction of [AS15] to other idealized models, including the ideal permutation model and the ideal cipher model.

notoriously difficult. Alwen et al. [ACP+17] did manage to tightly lower bound the cumulative memory complexity of Scrypt [Per09] without relying on dynamic pebbling, but their proof is technically intricate[3] and the techniques do not appear to extend to stronger metrics such as sustained space complexity. In particular, we still lack general techniques for proving meaningful SSC/CMC trade-offs for dMHFs directly in the PROM.

Goals: Ideally, we seek a practical dMHF that achieves the following SSC/CMC trade-off: any evaluation algorithm either (1) sustains $s = \Omega(N)$ space for $\Omega(N)$ rounds, or (2) incurs a superquadratic cumulative memory cost $\omega(N^2)$. Moreover, we aim to establish this trade-off ***directly*** in the PROM, rather than relying on the heuristic dynamic pebbling model, which currently lacks a rigorous justification.

The Gap: At present, we do not have any practical dMHF construction that provably achieves the ideal sustained-space guarantee—namely, $s = \Omega(N)$ sustained space for $\Omega(N)$ rounds together with a superquadratic CMC penalty $\omega(N^2)$—even in the heuristic dynamic pebbling model. Establishing such a trade-off remains an open challenge. As we previously noted Blocki and Holman [BH22] proposed a practical dMHF called ***Dynamic DRSample***, but its SSC/CMC trade-offs are suboptimal: it guarantees either $s = \Omega(N/\log N)$-sustained space for $\Omega(N)$ rounds or cumulative memory complexity $\Omega(N^3/\log N)$, and these guarantees are established only in the dynamic pebbling model. While the $\Omega(N^3/\log N)$ CMC penalty is asymptotically strong, the primary objective is to maximize sustained space complexity. The role of the CMC penalty is largely as a deterrent—it should ensure that any algorithm avoiding maximal sustained space incurs cumulative cost significantly larger than the honest evaluation's $\Theta(N^2)$ baseline.

Crucially, achieving this deterrent effect does not require a $\Theta(N^3)$ penalty. A superquadratic bound such as $\Omega(N^{2.4})$ would already suffice to ensure that any deviation from $s = \Omega(N)$ sustained space results in asymptotically higher cumulative cost. From this perspective, a practical dMHF guaranteeing either $\Omega(N)$ sustained space for $\Omega(N)$ rounds or CMC $\Omega(N^{2.4})$ would be preferable to one offering the incomparable guarantee of $\Omega(N/\log N)$ sustained space or CMC $\Omega(N^3)$ [BH22].

1.3 Our Results

We establish the first SSC/CMC trade-offs in the Parallel Random Oracle Model for data-dependent memory-hard functions (dMHFs). Our results in the PROM are as follows:

– We develop a method for generically proving SSC/CC trade-offs for dMHFs in the PROM for a family of graphs that satisfy a new combinatorial property called ancestral robustness.

[3] By contrast, proving that any ***dynamic pebbling strategy*** for Scrypt has cumulative cost $\Omega(N^2)$ is comparatively straightforward [ACP+17].

– We construct a dMHF called DEGSample, that achieves maximal sustained space parameters: any PROM algorithm either sustains $\Omega(N)$ space for $\Omega(N)$ steps or has cumulative complexity $\Omega(N^{2.49})$.

Along the way, we develop new techniques to lower bound the dynamic pebbling complexity

1.4 Notation and conventions

We begin by outlining some basic notation to be used in this paper. For integers $a \geq b$, we let $[a : b] = \{a, a+1, \ldots, b-1, b\}$ denote the set of integers from a to b. Similarly, we let $[a] \coloneqq [1 : a]$ denote the set of all positive integers at most a.

For a distribution D, we let $x \sim D$ denote that x is a random variable x sampled according to D. If S is a set, we let $x \sim S$ denote sampling from the uniform distribution over S.

Graph Notation: We use $G = (V = [N], E)$ to denote a directed acyclic graph G with N nodes. We typically assume that nodes are given in topological order i.e., if $(i, j) \in E$ then $i < j$. Given a node $v \in V$ we use $\mathsf{Parents}(v, G) \coloneqq \{u : (u, v) \in E\}$ to denote the set of parents of node v in G. We use $\mathsf{indegree}(v, G) = \left|\mathsf{Parents}(v, G)\right|$ to denote the number of incoming edges and $\mathsf{indegree}(G) = \max_v \mathsf{indegree}(v, G)$ to denote the maximum indegree of the DAG. Given a subset $S \subseteq V$ of nodes we use $G - S$ to denote the DAG obtained by removing nodes in the set S and any incident edges. We use $\mathtt{depth}(v, G)$ to denote the length of the longest directed path ***ending*** at node v and $\mathtt{depth}(G) = \max_v \mathtt{depth}(v, G)$ to denote the depth of the graph. Given a subset $Z \subseteq V$ of nodes we use $\mathsf{Ancestors}(Z, G)$ to denote subgraph of G induced by all ancestors of nodes $v \in Z$. More formally, let $A_{Z,G} \coloneqq \{u \ : \exists v \in Z \text{ s.t. } \mathtt{path}(u, v, G) = 1\}$ where the predicate $\mathtt{path}(u, v, G) = 1$ if and only if there is a directed path from u to v in G then $\mathsf{Ancestors}(Z, G) = G - (V \setminus A_{Z,G})$. When $Z = \{v\}$ contains a single node v we will sometimes write $\mathsf{Ancestors}(v, G)$ for $\mathsf{Ancestors}(Z, G)$.

When we enumerate the vertices of a graph $G = (V, E)$, we will always do so in topological order (from least to greatest). For example if we say $V = \{v_1, \ldots, v_N\}$, then if $j > i$ then v_j is topologically greater than v_i. Moreover, if $V = [N]$, then the nodes are again sorted topologically i.e., we have $j > i$ for any directed edge (i, j).

1.5 Background: MHFs and the Parallel Random Oracle Model

Parallel Random Oracle Model (PROM): Computation in the Parallel Random Oracle Model takes place over rounds. At the beginning of round $i - 1$ the PROM algorithm $\mathcal{A}$ is given an input state σ_{i-1} the PROM algorithm $\mathcal{A}(\sigma_{i-1})$ performs ***arbitrary computation*** before generating an output $\overline{\sigma}_i = (\tau_i, \mathbf{q}_i)$ where τ_i is an arbitrary bit string and $\mathbf{q}_i = [q_i^1, \ldots, q_i^{z_i}]$ is a batch of z_i queries to be submitted to the random oracle H in parallel. The answers $H(\mathbf{q}_1) = [H(q_i^1), \ldots, H(q_i^{z_i})]$ are appended to τ_i to obtain a new state $\sigma_i = (\tau_i, H(\mathbf{q}_1))$. Then the PROM algorithm $\mathcal{A}$ resumes computation with state $\sigma_i = (\tau_i, H(\mathbf{q}_1))$

and again performs arbitrary computation before generating an output $\overline{\sigma}_{i+1} = (\tau_{i+1}, \mathbf{q}_{i+1})$. Computation ends at round T if $\mathbf{q}_T = \{\}$ in which case the final output is just τ_T. The initial state σ_0 encodes the initial input x to our PROM algorithm $\mathcal{A}$ on initial input x. We use $\mathsf{trace}(\mathcal{A}, H, X) = (\sigma_1, \ldots, \sigma_T)$ to denote the execution trace when we run $\mathcal{A}$ on input X using the random oracle H.

The space usage of a PROM algorithm $\mathcal{A}$ at round i is the bit-length $|\sigma_i|$ of the state σ_i and the running time is measured based on the number of rounds T of random oracle queries made by $\mathcal{A}$. Thus, we can define the cumulative memory cost of a trace as $\mathsf{cmc}(\mathsf{trace}(\mathcal{A}, H, x)) = \sum_{i=1}^{T} |\sigma_i|$. Similarly, the s-sustained space complexity of an execution trace is $s\text{-}\mathsf{ssc}(\mathsf{trace}(\mathcal{A}, H, X)) \doteq \left|\left\{i \ : \ |\sigma_i| \geq s\right\}\right|$. An astute reader will observe that we are ignoring intermediate computation that occurs in-between rounds when measuring cumulative memory cost or sustained space complexity. However, this only strengthens the SSC/CMC lower bounds that we prove in this paper.

iMHFs: An iMHF f_G^H can be defined by a random oracle H and a directed acyclic graph (DAG) $G = (V = [N], E)$ encoding static data-dependencies. We typically assume that the nodes are given in topological order i.e., if $(u, v) \in E$ then $u < v$. Given an input X to our iMHF we can define labels $X_1, \ldots, X_N$ for each node $v \in V$ as follows: for the source node we have $X_1 = H(1, x)$ and for an internal node $v > 1 \in [N]$ with δ incoming edges $(u_1, v), (u_2, v), \ldots, (u_\delta, v)$, sorted so that $u_1 < u_2 < \ldots < u_\delta$, we have $X_v = H(v, X_{u_1}, X_{u_2}, \ldots, X_{u_\delta})$. The output of the iMHF is $f_G^H(X) = X_N$ where X_N is the label of the final sink node N.

dMHFs and Dynamic Graphs: The primary distinguishing feature of a dMHF is that the data-dependencies may be dependent on the input X as well as the random oracle H e.g., we might define $X_v = H(v, X_{r(v)}, X_{v-1})$ where the incoming edge $r(v) = X_{v-1} \pmod{v-2} + 1$ depends on the label X_{v-1} and would not be known in advance. We can instead model a dMHF $f_{\mathcal{G}}^H$ using a distribution $\mathcal{G}$ over graphs. We refer to the distribution $\mathcal{G}$ as a dynamic graph and often refer to a sample $G \sim \mathcal{G}$ as a static graph. Once we fix our random oracle H and our input x we obtain a static sample $G_{H,x}$ from our dynamic graph $\mathcal{G}$. We call $G_{H,x}$ the ex-post facto (after the fact) graph and we can define $f_{\mathcal{G}}^H(x) = f_{G_{H,x}}^H(x)$ i.e., the output of $f_{\mathcal{G}}^H(x)$ will be the label of the final node in our ex-post facto pebbling graph $G_{H,x}$.

Parallel Black Pebbling Game: The parallel black pebbling game is a powerful tool to analyze iMHFs. A pebbling $P = (P_1, \ldots, P_t)$ of a DAG $G = (V, E)$ is a sequence of sets $P_i \subseteq V$ where P_i intuitively represents the set of nodes $v \in V$ whose label X_v is currently stored in memory. The rules of the pebbling game state that if node $v \in P_i \setminus P_{i-1}$ is newly pebbled during round i then $\mathtt{parents}(v, G) \coloneqq \{u : (u, v) \in E\} \subseteq P_{i-1}$ — we implicitly define $P_0 = \{\}$ so that only source nodes can be pebbled during round 1. The cumulative cost (cc) of a pebbling $P = (P_1, \ldots, P_t)$ is $\mathsf{cc}(P) \coloneqq \sum_{i=1}^{t} |P_i|$ and the s-sustained space complexity is $s\text{-}\mathsf{ssc}(P) \coloneqq \left|\left\{j : |P_j| \geq s\right\}\right|$. We can similarly de-

fine $\mathsf{cc}(G) := \min_P \mathsf{cc}(P)$ (resp. $s\text{-}\mathsf{ssc}(G) := \min_P s\text{-}\mathsf{ssc}(P)$) where the minimum is taken over all legal pebblings P of G.

The usage of the parallel black pebbling game to analyze iMHFs can be rigorously justified [AS15,ABP18]. In a bit more detail if A^H is a PROM algorithm then running $A^H(x)$ on input x generates a execution trace $\mathsf{trace}(A, H, X) = (\sigma_1, \ldots, \sigma_t)$ where σ_i denotes the state of the PROM algorithm after the ith batch of random oracle queries. Simplifying a bit, the pebbling reductions of [AS15,ABP18] show that if $A^H(x)$ correctly computes $f_G^H(x)$ then (whp) one can find an ex-post facto pebbling $P = (P_1, \ldots, P_t)$ of G such that (1) P is legal, and (2) for all $i \leq t$ we have $|P_i| \leq |\sigma_i| / (c_{\mathsf{word}} w)$ where c_{word} is a constant, and w is the length of a random oracle output i.e., $H(x) \in \{0,1\}^w$. This result tightly links the cumulative memory complexity (resp. sustained space complexity) of the function f_G^H to the cumulative pebbling complexity (resp. sustained space complexity) of the graph G.

Remark 1. In a recent breakthrough result Ryan Williams [Wil25] showed that time T multitape Turing Machines can be simulated with just $O(\sqrt{T \log T})$ space and that bounded fan-in circuits of size s can be evaluated on any input with space $\sqrt{s} \cdot \mathsf{poly}(\log s)$. Consider an iMHF $f_{G,H}$ and suppose that the hash function $H : \{0,1\}^{\delta w + \log N} \to \{0,1\}^w$ can be evaluated by a circuit of size $O(\mathsf{poly}(w))$ and there is $\mathsf{poly}(\log N)$ sized circuit C_G which computes the edges of G i.e., $C_G(v)$ outputs all of the parents $u_1, \ldots, u_v$ of node v. Under these assumptions there will be a bounded fan-in circuit of size $s = O(N(\mathsf{poly}(w) + \mathsf{poly}(\log N))$ which computes $f_{G,H}$. Applying [Wil25] this would imply that $f_{G,H}$ can evaluated with space at most $O(\sqrt{N}\mathsf{poly}(w) + \sqrt{N} \log N)$. In particular, the function $f_{G,H}$ cannot have high sustained space complexity. On the other hand, by instantiating the graph construction of [ABP18] with an explicit construction of an extremely depth-robust graph [BCLS22], one can obtain an explicit construction of a graph G with the property that (1) any pebbling of G requires $\Omega(N)$ rounds with $\Omega(N/\log N)$ pebbles, and (2) there is $\mathsf{poly}(\log N)$ sized circuit C_G which computes the edges of G — see [ABP18, Remark 3]. Because G has high sustained space complexity, the pebbling reductions of [AS15,ABP18] seems to imply that any PROM algorithm A^H computing $f_{G,H}$ has high sustained space complexity which seemingly contradicts our previous observation that $f_{G,H}$ cannot have high sustained space complexity. However, upon closer examination, the pebbling reductions of [AS15,ABP18] are allowed to fail if A^H makes exponentially many queries to the random oracle H. By contrast, the low space simulations of [Wil25] run in exponential time. While we do not obtain a contradiction, the results of [Wil25] are still quite informative. In particular, the goal of constructing a MHF $f_{G,H}$ with an ***absolute*** guarantee of high sustained space complexity guarantees appears to be unattainable. This further motivates the study of SSC/CMC tradeoffs.

Dynamic Pebbling For a dMHF the parents of node v may not be known until the label X_{v-1} has been computed for the first time. Thus, a dynamic pebbling strategy $\mathcal{S}$ for a dynamic graph $\mathcal{G}$ must adapt as these dynamic edges

are revealed, that is, some of the parents of node v may not be revealed until we place a pebble on node $v-1$. We can use $\mathcal{S}(G)$ to denote the final pebbling that is produced when we use the dynamic pebbling strategy $\mathcal{S}$ and the sampled graph is $G \sim \mathcal{G}$. Unlike iMHFs there is no rigorous justification for analyzing SSC/CMC trade-offs using the dynamic pebbling game. Ideally, a dynamic pebbling reduction should instead lower bound the SSC/CMC of $A^H(x)$ using the SSC/CC of $\mathcal{S}(G_{H,x})$ where $\mathcal{S}$ is the optimal dynamic pebbling strategy for $\mathcal{G}$. Unfortunately, there is no known dynamic pebbling reduction which shows this and Alwen et al. [ACP+17] identified several significant technical barriers to finding such a dynamic pebbling reduction. Thus, proving that a SSC/CMC trade-off holds for any dynamic pebbling strategy does not necessarily imply that the same SSC/CMC trade-off holds in the PROM.

Depth-Robustness Depth-robust graphs have proved to be a useful tool to design and analyze MHFs. A DAG $G = ([N], E)$ is (e, d)-depth robust if for any set $S \subseteq [N]$ of size at most $|S| \leq e$ we have $\mathsf{depth}(G - S) \geq d$. We say that G is (e, d, f)-fractionally depth robust [BZ17] if for any set $S \subseteq [N]$ of size at most $|S| \leq e$ we have $\left|\left\{v \in [N] \setminus S \ : \ \mathsf{depth}(v, G - S) \geq d\right\}\right| \geq fN$ i.e., there are fN nodes that are the endpoint of a directed path of length d in $G - S$. Observe that if G is (e, d)-depth robust then for any $S \subseteq [N]$ of size $|S| < e$ the remaining graph $G - S$ is still $(e - |S|, d)$-depth robust. [ABP17] proved that if G is (e, d)-depth robust then $\mathsf{cc}(G) > ed$ i.e., depth-robust graphs must have high cumulative pebbling complexity.

See Section A for a more formal description of the black pebbling game, dynamic pebbling game and parallel random oracle model as well as background on depth-robust graphs and pebbling.

2 Technical Overview

Our results have three primary components, culminating in the first dMHF to be proven secure with respect to sustained space complexity in the Parallel Random Oracle Model. We first introduce a generic procedure $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ (Definition 1) to transform a static graph G into a dynamic graph $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$. We then identify two useful combinatorial properties for the underlying static graph G to satisfy: fractional depth-robustness (Definition 6) and ancestral robustness (Definition 8). Here, ancestral robustness is a new property that we introduce.

Assuming that G satisfies both properties we can prove strong SSC/CMC trade-offs in the dynamic pebbling model. In particular, for any dynamic pebble strategy $\mathcal{S}$ the following holds with high probability over the selection of $G' \sim \mathsf{Dynamize}(G, N)$ either (1) $s\text{-}\mathsf{ssc}\left(\mathcal{S}(G')\right) = \Omega(N)$ for some $s = \Omega(N)$, or (2) $\mathsf{cc}(\mathcal{S}(G')) = \omega(N^2)$. See the overview in Section 2.1 or see Section B for full details.

Second, we develop techniques to prove SSC/CMC lower bounds directly in the PROM. We show that if $\mathcal{G} = \mathsf{Dynamize}(G, N)$ and the underlying graph

is fractionally depth-robust and ancestrally robust then the dMHF $f_{\mathcal{G}}^{H}$ satisfies strong SSC/CMC trade-offs. See the overview in Section 2.2 or see Section C for the full details.

Finally, we show how to construct a static graph G that satisfies both of the required properties: fractional depth-robustness and ancestral robustness. More specifically, in Theorem 10, we show that (with high probability) the graphs output by the randomized algorithm DRSample [ABH17] have high ancestral robustness. By combining DRSample with a family of fractionally depth-robust graphs called Grates [Sch83], we obtain a static constant-indegree graph EGSample (Definition 12) which satisfies both of our required properties. We then provide the following lower bounds for the corresponding dynamic graph/dMHF called DEGSample (Definition 13):

- **(Corollary 1)** In the pebbling model, any dynamic pebbling strategy on DEGS sustains $\Omega(N)$ pebbles for $\Omega(N)$ steps or, (with high probability) incurs CC $\Omega(N^{3-\varepsilon})$.
- **(Corollary 2)** In the PROM, any parallel algorithm evaluating DEGS sustains $\Omega(wN)$ memory for $\Omega(N)$ or, (with high probability) incurs CMC $\Omega(wN^{2.5-\varepsilon})$. Here, w is the word size for the random oracle $H : \{0,1\}^* \to \{0,1\}^w$.

See the overview in Section 2.4 or see Section E and Section D for full details.

2.1 Pebbling SSC/CC Trade-offs via Ancestral Robustness

We introduce a generic procedure $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ which transforms our static graph $G = ([N], E)$ into a dynamic graph with $N + N_{\mathsf{chal}}$ nodes (see Fig. 1 for an example). The dynamic graph is formed by appending a line graph with N_{chal} nodes $\ell_1 = N+1, \ldots, \ell_{N_{\mathsf{chal}}} = N+N_{\mathsf{chal}}$ to G, adding a directed edge (N, ℓ_1) from the sink node of G to the first node ℓ_1 in the line graph, then adding dynamic edges $(r(i), \ell_i)$ for each $i \leq N_{\mathsf{chal}}$. Intuitively, $r(i)$ is picked uniformly at random from $[N]$ using randomness from the previous label $X_{\ell_{i-1}}$ — for convenience we define $\ell_0 = N + 0$ so that ℓ_{i-1} is well defined when $i = 1$. In practice, this can be achieved by setting $r(i) = 1 + (X_{\ell_{i-1}} \mod N)$ since $X_{\ell_{i-1}}$ is a random oracle output and can thus be interpreted as a uniformly random integer between 0 and $2^w - 1$ — assuming that $N = 2^n$ with $n \leq w$ the value $X_{\ell_{i-1}} \mod N$ is uniformly random over the set $\{0, \ldots N-1\}$.

Definition 1 (Dynamize). *Fix a graph $G = ([N], E)$ and an integer $N_{\mathsf{chal}} \geq 1$. The dynamic pebbling graph $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ consists of the static graph $G' = ([N + N_{\mathsf{chal}}], E')$ with $N + N_{\mathsf{chal}}$ nodes and static edges $E' = E \cup \{(i, i+1) : N \leq i < N + N_{\mathsf{chal}}\}$ along with N_{chal} dynamic edges*

$$E_{\mathsf{Dynamic}} = \left\{(r(1), \ell_1), \ldots, (r(N_{\mathsf{chal}}), \ell_{N_{\mathsf{chal}}})\right\},$$

where $\ell_i = N + i$ denotes the ith node in the line graph node that we append to G and each $r(i)$ is chosen uniformly at random from the nodes $[N]$ of G.

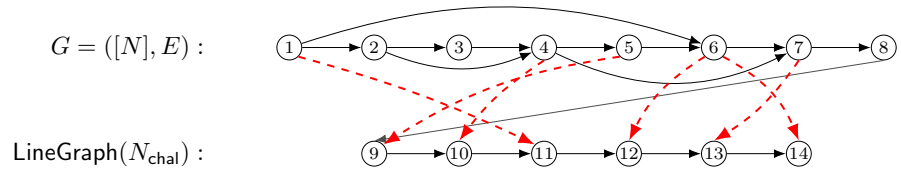

Fig. 1. Shown is a sample G' of the dynamic pebbling graph $\mathsf{Dynamize}(G = ([N], E), N_{\mathsf{chal}})$. In this case, $N = 8$ and $N_{\mathsf{chal}} = 6$. Each node of the bottom line graph has a dynamic (red, dashed) incoming edge. These dynamic predecessors are each chosen uniformly at random from $[N]$. In this example, if a dynamic pebbling strategy places a pebble on node 12 for the first time, it will be indicated that they must also pebble node 7 before placing a pebble on node 13.

Scrypt is a data-dependent memory-hard function that was notably shown to have maximal cumulative complexity. We note that the dynamic pebbling graph $\mathcal{G}_{\mathsf{scrypt}}$ for Scrypt is equal to $\mathcal{G}_{\mathsf{scrypt}} = \mathsf{Dynamize}(L_N, N)$ where $L_N = ([N], \{(i, i+1) : 1 \leq i < N\})$ denotes the line graph on N nodes. While $\mathcal{G}_{\mathsf{scrypt}}$ has cumulative pebbling complexity $\Omega(N^2)$, the dynamic graph does not offer strong SSC/CMC trade-offs. In particular, there is a dynamic pebbling strategy $\mathcal{S}$ which uses ***at most*** $s = 2$ pebbles and has CMC $O(N^2)$.

Our goal, then, is to find a family of constant indegree graphs G with respect to the parameter N in which any dynamic pebbling strategy $\mathcal{S}$ for $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ (with high probability) either keeps around $\Omega(N)$ pebbles for $\Omega(N_{\mathsf{chal}})$ steps, or has cumulative complexity $\omega(N^2)$. We stress that our primary goal is to achieve maximal sustained space parameters i.e., ensure that any pebbling strategy either has $s = \Omega(N)$-sustained space complexity $\Omega(N)$ or incurs cumulative complexity $\omega(N^2)$. The secondary goal is to ensure that the cumulative memory cost penalty for dynamic pebbling strategies that do not have maximum sustained space complexity is high enough to ensure that an attacker would prefer to follow a strategy with maximum sustained space complexity and CMC at most $O(N^2)$. We note that for any constant indegree DAG G and any constant $c > 0$, $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ can be pebbled using at most cN space and with cumulative complexity at most $O(N^2 N_{\mathsf{chal}})$ [BH22,LT79], Thus, if $s = \Omega(N)$ we cannot expect to achieve a cumulative complexity penalty $\omega(N^2 N_{\mathsf{chal}})$ for dynamic pebbling strategies which do not have s-sustained space complexity $\Omega(N)$. However, we would still like the cumulative pebbling cost penalty to be as close to $N^2 N_{\mathsf{chal}}$ as possible.

We show that if G satisfies two combinatorial properties, then $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ can achieve near-optimal SSC/CC trade-offs. These two combinatorial properties are *fractional-depth robustness* (Definition 6) and a new combinatorial property called *ancestral robustness* (Definition 8) that we introduce below.

Recall that a graph G is (e, d, f)-fractionally depth-robust if for any set of nodes $S \subseteq [N]$ of size $|S| \leq e$, the graph $G - S$ contains at least fN nodes with

depth at least d i.e., for all $S \subseteq [N]$ with $|S| \leq e$ we have $\left|\{v \ : \ \mathsf{depth}(v, G-S) \geq d\}\right| \geq fN$. Intuitively, if G is (e, d, f)-fractionally depth-robust and we are challenged to place a pebble on a uniformly random node starting from an initial pebbling configuration with at most e pebbles on the graph then, with probability at least f, it will take ***at least*** d pebbling rounds to respond to the challenge. In our constructions, we will rely on a family of constant indegree DAGs from [Sch83] that are (e, d, f)-fractionally depth-robust for $e = \Omega(N)$, $d = N^{1-\epsilon}$ and $f = \Omega(1)$.

Ancestral Robustness: We say G is (a, C, f')-ancestrally robust (Definition 8) if for every set $A \subseteq [N]$ of size a, there are $f'N$ nodes v in the remaining graph $G - A$ whose ancestors have cumulative complexity at least C:

$$\mathsf{cc}(\mathsf{Ancestors}(v, G - A)) \geq C.$$

More formally we have $\left|\{v : \ \mathsf{cc}(\mathsf{Ancestors}(v, G - A)) \geq C\}\right| \geq f'N$ for every set $A \subseteq [N]$ of size $|A| \leq a$. Intuitively, if G is (a, C, f')-ancestrally robust and we are challenged to place a pebble on a random node in G starting from an initial pebbling configuration with at most a pebbles on the graph then, with probability at least f', we will incur cumulative pebbling cost at least C to respond to the challenge.

If G is both (e, d, f)-fractionally depth-robust and (a, C, f')-ancestrally robust, then we show that any dynamic pebbling strategy $\mathcal{S}$ for $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ satisfies the following trade-off of sustained space and cumulative complexity.

Theorem 1 (Generic Pebbling SSC/CC Trade-off (Informal)). *If $G = ([N], E)$ is (e, d, f)-fractionally depth robust and (a, C, f')-ancestrally robust, then for any pebbling strategy $\mathcal{S}$ and except with negligible probability over $G' \sim \mathsf{Dynamize}(G, N_{\mathsf{chal}})$, either $\mathcal{S}(G')$ sustains e pebbles for $\Omega(N_{\mathsf{chal}})$ steps:*

$$e\text{-}\mathsf{ssc}\left(\mathcal{S}(G')\right) = \left|\left\{i : \ \left|\mathcal{S}(G')_i\right| \geq e\right\}\right| = \Omega(N_{\mathsf{chal}})$$

or $\mathcal{S}(G')$ has cumulative complexity

$$\mathsf{cc}(\mathcal{S}(G')) = \sum_i \left|\mathcal{S}(G)_i\right| \geq \Omega\left(N_{\mathsf{chal}} \cdot \min\{ad, C\}\right)$$

To illustrate the power of Theorem 1 suppose that G is (e, d, f)-fractionally depth robust and (a, C, f')-ancestrally robust for $e = \Omega(N)$ and $d = \Omega(N^{1-\epsilon/2})$, $a = \Omega(N^{1-\epsilon/2})$, $C = \Omega(N^{2-\epsilon})$ with $f, f' > 0$ constant — we will show later how to construct a constant indegree graph G with these properties. Now if we set $N_{\mathsf{chal}} = N$, then for any dynamic pebbling strategy $\mathcal{S}$ (with high probability) we either have (1) $\Omega(N)$ rounds with at least $\Omega(N)$ pebbles on the graph, or (2) cumulative pebbling cost at least $\mathsf{cc}(\mathcal{S}(G)) = \Omega(N^{3-\epsilon})$.

We will now explain how these ancestrally/fractionally depth robust graphs yield dMHFs with strong SSC/CC trade-offs. While we end up with the same

SSC/CC trade-off as the theoretical construction of [BH22], the proof is quite different since the properties of ancestral robustness and fractional robustness are somewhat weaker than the combinatorial property called ST-robustness used in the theoretical construction of [BH22]. As it turns out, this new dynamic pebbling proof technique also turns out to be more amenable to adapt to directly prove SSC/CMC trade-offs in the parallel random oracle model.

Fix a graph $G' = \big([N], E\big)$ that is (e, d, f)-fractionally depth-robust and (a, C, f')-ancestrally robust, with $a < e$. Let us consider a dynamic pebbling strategy $\mathcal{S}$ for $\mathcal{G} = \mathsf{Dynamize}(G', 2N_{\mathsf{chal}})$. We let $P = \mathcal{S}(G) = (P_1, \ldots, P_T)$ be the resulting pebbling with $G \sim \mathcal{G}$. It is convenient to group the last $2N_{\mathsf{chal}}$ challenge nodes into consecutive pairs $\ell_i = (\ell_i^1, \ell_i^2) = (N + 2i - 1, N + 2i)$ for each $i \leq N_{\mathsf{chal}}$. We'll refer to ℓ_i as the ith challenge pair. For $b \in \{1, 2\}$ we let $s_b(i) \doteq \min\{j : \ell_i^b - 1 \in P_j\}$ denote the first round where we place a pebble on node $\ell_i^b - 1$. Intuitively, the dynamic parent $r(2i-2+b)$ of node $\ell_i^b = N+2i-2+b$ is unknown before round $s_b(i)$. Finally, we let $t_b(i) = \min\{t : r(2i-2+b) \in P_{s_b(i)+t}\}$ denote the time between the round $s_b(i)$ where the challenge $r(2i - 2 + b)$, the dynamic parent of ℓ_i^b, is revealed and the round where the challenge is answered by placing a pebble on node $r(2i - 2 + b)$.

For the ith challenge pair, we aim to lower bound the cost of the pebbling sequence

$$P^i = \Big(P_{s_1(i)}, \ldots, P_{s(i)+t_1(i)}, P_{s_2(i)}, \ldots, P_{s_2(i)+t_2(i)}\Big)$$

starting when the first challenge $r(2i - 1)$ is discovered and ending when the second challenge is answered by placing a pebble on node $r(2i)$. If the pebbling configuration $P_{s_1(i)}$ at the start of the challenge has size at least e, then the subsequence P^i contributes at least 1 to the e-sustained space cost. If this this is true for $\Omega(N_{\mathsf{chal}})$ indices i, i.e., if $\Big|\{i \ : \ \Big|P_{s_1(i)}\Big| \geq e\}\Big| = \Omega(N_{\mathsf{chal}})$, then the e-sustained space cost is $\Omega(N_{\mathsf{chal}})$ and we are immediately done.

However, it is possible that $\Big|\{i \ : \ \Big|P_{s_1(i)}\Big| \geq e\}\Big| = o(N_{\mathsf{chal}})$. Thus, we must consider the more difficult case, where $\Big|P_{s_1(i)}\Big| < e$. Because G is (e, d, f)-fractionally depth-robust and the initial pebbling configuration has at most $\Big|P_{s_1(i)}\Big| < e$ pebbles there are at least fN nodes v such that $\mathtt{depth}(v, G - P_{s_1(i)} \geq d$. Thus, $\Pr[\mathtt{depth}(r(2i - 1), G - P_{s_1(i)}) \geq d] \geq f$ where the probability is taken over the uniformly random selection of the challenge $r(2i - 1) \sim [N]$. Thus, with probability at least f we will have $t_1(i) \geq d$ i.e., it will take at least d rounds to respond to the first challenge $r(2i - 1)$.

If $t_1(i) \geq d$ and $|P_j| \geq a$ for each round $s_1(i) \leq j \leq s_1(j) + t_1(i)$ then we the cumulative complexity incurred is ***at least***

$$\mathsf{cc}(P^i) = \sum_{j \in [s_1(i):s_2(i)+t_2(i)]} \Big|P_j^i\Big| \geq \sum_{j \in [s_1(i):s_1(i)+t_1(i)]} \Big|P_j^i\Big| \geq ad.$$

Otherwise, if $|P_j| < a$ for some round $s_1(i) \leq j \leq s_1(j) + t_1(i)$ we can appeal to the (a, C, f') ancestral robustness G to argue that with probability

at least f' the pebbling will incur cost cumulative pebbling cost ***at least*** C between round j and $s_2(i) + t_2(i)$. In particular, we have $\sum_{z=j}^{s_2(j)+t_2(i)} |P_z| \geq \mathsf{cc}\left(\mathsf{Ancestors}(r(2i), G - P_j)\right)$ and because the challenge $r(2i)$ is picked uniformly at random and not revealed by round j we have $\Pr[\mathsf{cc}\left(\mathsf{Ancestors}(r(2i), G - P_j)\right) \geq C] \geq f'$.

To summarize, we say that the ith challenge pair is *costly* if P^i falls into one of these categories:

1. **High Space:** $\left|P_{s_1(i)}\right| \geq e$,
2. **Medium Space, High Depth:** $\mathsf{depth}(r(2i-1), G - P_{s_1(i)}) \geq d$ and $a \leq \left|P_j\right|$ for all $s_1(i) \leq j \leq s_1(i) + t_1(i)$, or
3. **Low Space, High CC:** $\mathsf{depth}(r(2i-1), G - P_{s_1(i)}) \geq d$ and $\left|P_j\right| < a$ for some $j \in [s_1(i), s_1(i) + t_1(i)]$ and $\mathsf{cc}(\mathsf{Ancestors}(r(2i), G - P_j)) \geq C$

Let Z_i be an indicator random variable indicating whether or not the ith challenge pair is *costly*. We can show that for any prior outcomes $z_1, \ldots, z_{i-1}$ we have $\Pr[Z_i = 1 | (Z_1, \ldots, Z_{i-1}) = z_1, \ldots, z_{i-1}] \geq ff'$ — see Lemma 5. Intuitively, if we assume that $|P_{s_1(i)}| < e$ (otherwise we already have $Z_i = 1$) then ff' lower bounds the probability of a costly event $Z_i = 1$ since of the event that $\mathsf{depth}(r(2i-1), G - P_{s_1(i)}) \geq d$ occurs with probability ***at least*** f. If $\mathsf{depth}(r(2i-1), G - P_{s_1(i)}) \geq d$ we can assume wlog that $\left|P_j\right| < a$ for some $j \in [s_1(i), s_1(i) + t_1(i)]$ (otherwise we already have $Z_i = 1$ as we are in the Medium Space, High Depth case) and we have $\mathsf{cc}(\mathsf{Ancestors}(r(2i), G - P_j)) \geq C$ with probability f'. We can apply concentration bounds to argue that the total number of costly rounds is $Z = Z_1 + \ldots, Z_{N_{\mathsf{chal}}} \geq ff' N_{\mathsf{chal}}/2$ with high probability.

Thus, either we will have $\Omega(N_{\mathsf{chal}})$ high space challenges i with $\left|P_{s_1(i)}\right| \geq e$, or we will have $\Omega(N_{\mathsf{chal}})$ challenges where the subsequence P^i either incurs cumulative cost $\mathsf{cc}(P^i) \geq ad$ or $\mathsf{cc}(P^i) \geq C$. In particular, in this case there are $\Omega(N_{\mathsf{chal}})$ challenges where the subsequence P^i incurs cumulative cost at least $\mathsf{cc}(P^i) \geq \min\{C, ad\}$ and the total cumulative pebbling cost will be $\Omega\left(N_{\mathsf{chal}} \min\{C, ad\}\right)$. Theorem 1 follows immediately from these observations.

2.2 SSC/CMC Trade-offs in the Parallel Random Oracle Model

Theorem 1 implies that our dynamic graph $\mathcal{G} = \mathsf{Dynamize}(G, N)$ will have strong SSC/CC trade-offs in the dynamic pebbling model provided that the underlying graph G is (a, C, f')-ancestral robust and (e, d, f)-fractional depth-robust for suitable parameters a, C, f', e, d and f. However, as we previously noted we lack rigorous justification for the use of the heuristic dynamic pebbling model when analyzing SSC/CMC trade-offs. In this section we aim to directly analyze SSC/CC trade-offs for the dMHF $f_{\mathcal{G}}^H$ in the parallel random oracle model (PROM).

To prove the first SSC/CMC trade-offs for dMHFs in the PROM model we combine techniques of Alwen et al. [ACP+17] with the notion of an ex-post facto pebbling [AS15] of an ex-post-facto graph to establish PROM bounds for

MHFs. Taken individually both of these techniques fail to establish meaningful SSC/CMC trade-offs, but taken in combination we are able to draw powerful conclusions.

2.3 Summary of SSC/CMC Trade-offs in the PROM

We again consider an (e, d, f)-fractionally depth robust and (a, C, f')-ancestrally robust graph $G = ([N], E)$ and its corresponding dynamic graph $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$. Theorem 2 establishes a strong SSC/CMC trade-off for our dMHF construction $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ provided that the underlying graph is (e, d, f)-fractionally depth robust and (a, C, f')-ancestrally robust.

Theorem 2 (Generic PROM SSC/CMC Trade-off (Informal)). *Fix a graph $G = ([N], E)$ that is (e, d, f)-fractionally depth robust and (a, C, f')-ancestrally robust graph G, let $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$, and fix any $X \in \{0,1\}^w$. For any PROM algorithm $\mathcal{A}$ with access to the random oracle $H : \{0,1\}^* \to \{0,1\}^w$ that computes $f_{\mathcal{G},H}$ on X, one of the following is true except with negligible probability:*

1. *$\mathcal{A}$ sustains $\Omega(w \cdot e)$ space for $\Omega(N_{\mathsf{chal}})$ steps. For some $s = \Omega(we)$ we have :*

$$s\text{-}\mathsf{ssc}(\mathsf{trace}(\mathcal{A}, H, X)) = \Omega(N_{\mathsf{chal}}).$$

2. *$\mathcal{A}$ has cumulative memory complexity at least*

$$\mathsf{cmc}(\mathsf{trace}(A, H, X)) = \sum_i |\sigma_i| = \tilde{\Omega}\left(wN_{\mathsf{chal}} \cdot \min\left\{ad, \frac{C}{a \log N}\right\}\right).$$

In Fig. 2, we present a side-by-side comparison of the PROM and the dynamic pebbling bound described previously. In the PROM analysis, there is an extra factor of w for the space parameters; this reflects the fact that storing a single label for a node requires w bits, where w is the "word length" corresponding to a random oracle $H : \{0,1\}^* \to \{0,1\}^w$.

2.3.1 Lower Bounding the Number of Rounds Between Challenges A result of Alwen et al. [ACP+17] allows us to make the following PROM argument when the underlying static graph $G = ([N], E)$ in $\mathsf{Dynamize}(G, N)$ is (e, d, f)-fractionally depth robust. Suppose that we are given an initial state σ_i of size $|\sigma_i| < we/2$ and then challenged to compute the label X_v of a uniformly random node $v \in [N]$ where v is sampled ***after*** the initial state σ_i is fixed. Then with probability ***at least*** $f - \mathtt{negl}(w)$ it will take $\mathcal{A}$ at least d rounds to compute X_v e.g., see Lemma 21. This is a useful observation, but it is not clear how to use this observation alone to prove strong SSC/CMC trade-offs.

Given an execution trace $\mathsf{trace}(A, H, X)$ we can let s_i denote the first round where the label $X_{\ell_i - 1}$ appears as an output to one of the random oracle queries i.e., the first round where the dynamic edge $(r(i), \ell_i = N + i)$ is revealed. We

Pebbling vs. PROM Trade-off Comparison

Analysis Model	dMHF	Space Sustained for $\Omega(N)$ Steps	CMC Penalty if Lower Sustained Space
Dynamic Pebbling	Generic	e	$\Omega\left(N \min\{ad, C\}\right)$
PROM	Generic	$\Omega(w \cdot e)$	$\Omega\left(Nw \cdot \min\left\{ad, \frac{C}{a \log N}\right\}\right)$
Dynamic Pebbling	DEGSample	$\Omega(N)$	$\Omega\left(N^{3-\varepsilon}\right)$
PROM	DEGSample	$\Omega(Nw)$	$\Omega\left(N^{2.5-\varepsilon}\right)$

Fig. 2. This table contains the (asymptotic) SSC/CMC trade-offs for dMHFs stemming from the generic dMHF construction $\mathsf{Dynamize}(G, N)$, where G is (e, d, f)-fractionally depth-robust and (a, C, f')-ancestrally robust as well as DEGSample. Here, $\varepsilon > 0$ is a chosen constant. For the rows corresponding to the PROM, w is the "word length," the bit-length of the label for a node.

can let $t_i \geq 0$ denote the time to respond to the challenge i.e., the number of rounds that pass before the label $X_{r(i)}$ appears as an input to a random oracle query in a future round $s_i + t_i$. Theorem 5 implies that if the execution trace $\mathsf{trace}(A, H, X)$ does not have asymptotically optimal sustained space complexity then there will be a set B of $|B| = \Omega(N)$ challenges i such that $t_i \geq d$ for each $i \in B$. Once again this observation alone does not imply a that there is a large cumulative memory cost penalty. In particular, it could be that $\left|\sigma_j\right|$ is small for all intermediate rounds $j \in [s_i, s_i + t_i]$ in which case the contribution $\sum_{j=s_i}^{s_i+t_i} \left|\sigma_j\right|$ to the overall cumulative memory cost might also be smaller. For a bird's-eye view of how we prove SSC/CC trade-off lower bounds in the PROM, see Fig. 5.

2.3.2 Ex-Post Facto Pebbling of Ex-Post Facto Graph We also observe that, given an execution trace $\mathsf{trace}(A, H, X)$, in which $f_G^H(X)$ is computed correctly we can extract an ***ex post facto*** pebbling $P = (P_1, \ldots, P_t)$ of the ***ex post facto*** graph $G_{H,X}$. We can further argue that, except with negligible probability, that (1) P is a legal pebbling of $G_{H,X}$ and (2) for all $i \in [T]$ we have $|P_i| \leq |\sigma_i| / (c_{\mathsf{word}} w)$ for some constant $c_{\mathsf{word}} > 0$. See Theorem 6 for the formal claim. This argument is very similar to prior pebbling arguments [AS15,ABP18] except that the final graph $G_{H,X}$ is not fixed a priori. Taken alone this observation will not allow us to establish strong SSC/CMC trade-offs because $G_{H,X}$ is a static graph with constant indegree we have $\mathsf{cc}(G_{H,X}) = O(N^2 \log\log N / \log N)$ [AB16]. We cannot achieve strong SSC/CMC trade-offs without exploiting dynamic edges.

Even if $\mathcal{S}$ is an optimal dynamic pebbling strategy for our dynamic graph $\mathcal{G}$ we cannot necessarily conclude that $\mathcal{S}(G_{H,x})$ will be the optimal pebbling for the static graph $G_{H,x}$ because $\mathcal{S}$ does not know $G_{H,x}$ in advance e.g., we may have $\mathsf{cc}(\mathcal{S}(G_{H,x})) \gg \mathsf{cc}(G_{H,x})$. As a concrete example, if $\mathcal{G}_{Scrypt}$ denotes the dynamic

pebbling graph underlying Scrypt [Per09] then ***any*** graph G in the support of $\mathcal{G}_{Scrypt}$ we have $\mathsf{cc}(G_{H,x}) = O(N^{1.5})$ [4] while for any dynamic pebbling strategy $\mathcal{S}$ we will have $\mathsf{cc}(\mathcal{S}(G)) = \Omega(N^2)$ with high probability [ACP+17]. Ideally, we would like to have a dynamic pebbling reduction which lower bounds the SSC/CMC of a PROM algorithm $A^H(x)$ using the the SSC/CC of the best dynamic pebbling strategy $\mathcal{S}(G_{H,x})$ instead of the SSC/CC of the ex-post facto graph $G_{H,x}$. Unfortunately, no such dynamic pebbling reduction is known and Alwen et al. [ACP+17] identified several significant technical barriers to finding such a dynamic pebbling reduction.

Lower Bounding The Pebbling Cost of a Constrained Pebbling Based on the above discussion it may seem pointless to consider ex-post facto pebblings of the ex-post facto graph. Our key insight is that when the execution trace $\mathsf{trace}(A, H, X)$ has low sustained space complexity we can often impose additional constraints on the ex-post facto pebbling by leveraging techniques that Alwen et al. [ACP+17] used to lower bound the cumulative memory complexity of Scrypt in the PROM. For example, suppose that the underlying static graph G in our dynamic graph $\mathsf{Dynamize}(G, N)$ is (e, d, f)-fractionally depth robust with $e = \Omega(N)$, $d = \Omega(N^{1-\epsilon})$ and $f = \Omega(1)$. Then it will be possible to show that, except with negligible probability, there will be $\Omega(N)$ challenge nodes $i \in [N]$ where $s(i+1) - s(i) - 1 \geq d$ i.e., at least d intermediate rounds elapse between the first round $s(i)$ where we place a pebble on node $N+i-1$ and the first round $s(i+1)$ where we place a pebble on node $N+i$. If the underlying graph G is also (a, C, f')-ancestrally robustness then we show that (whp) any legal pebbling of the ex-post facto graph $G_{H,X}$ that respects the above constraints ***must*** have high cumulative pebbling complexity.

Let $B = \{i : s(i+1) - s(i) - 1 \geq d\}$ denote the challenge nodes where at least d intermediate pebbling rounds elapse between the first round where we place a pebble on node $N + i - 1$ and the first round where we place a pebble on node $N + i$. Fixing a challenge $i \in B$ observe that if $|P_j| \geq a$ for all intermediate rounds $j \in [s(i), S(i+1)-1]$ then we trivially have $\sum_{j=s(i)}^{s(i+1)-1} |P_j| \geq ad$ i.e., the cumulative pebbling cost incurred the while advancing pebble from node $N + i - 1$ to node $N + i$ for the first time is large. Now if we let $D = \left\{i \in B : |P_j| \geq a \forall j \in [s(i), S(i+1) - 1]\right\} \subseteq B$ then we trivially have $\mathsf{cc}(P) \geq \sum_{i \in D} \sum_{j=s(i)}^{s(i+1)-1} |P_j| \geq |D|ad$. If $|D| \geq |B|/2$ then we have $\mathsf{cc}(P) = \Omega(adN)$ i.e., $\mathsf{cc}(P) = \Omega(N^{2.5-\epsilon})$ if $a = \Omega(\sqrt{N})$ and $d = \Omega(N^{1-\epsilon})$.

The slightly more challenging case arises when $|D| < |B|/2$. If $i \in B \setminus D$ then by definition we have $|P_j| < a$ for some round $j \in [s(i), S(i+1) - 1]$. Our key insight here is to leverage (a, C, f')-ancestrally robustness of G. We will be able to argue that for some $k = \Theta(a \log N)$ (whp) ***any*** legal pebbling of the static ex-post facto graph $G_{H,X}$ must incur cost C between rounds j and $s(i+k)-1$ for all $i \in B \setminus D$ — see Lemma 6. It follows that $\sum_{j=s(i)}^{s(k)-1} |P_i| \geq C$ and

[4] Lemma 5 in [AS15] shows that any graph that satisfies the property of being a "sandwich graph" has $\mathsf{cc}(G) = O(N^{1.5})$. Any DAG G in the support of $\mathcal{G}_{Scrypt}$ satisfies this property.

$\mathsf{cc}(P) \geq \sum_{i \in B \setminus D} C/k \geq |B|C/(2k) = \Omega(NC/(a \log N))$ i.e., $\mathsf{cc}(P) = \tilde{\Omega}(N^{2.5-\epsilon})$ if $a = \sqrt{N}$ and $C = \Omega(N^{1.5-\epsilon})$.

2.3.3 Combining the Techniques to Prove Theorem 2 We would like to perform a similar case analysis as we did for dynamic pebblings. If $|\sigma_{s_i}| \geq we/2$ then this challenge contributes to sustained space complexity. Otherwise, with probability $\approx f$ we will have $t_i \geq d$. Thus, far the analysis can be adapted to the PROM. Next we want to argue that either (1) we have $\left|\sigma_{s_i+j}\right| \geq aw/2$ for all $j \leq d$ so that we incur cumulative memory cost ***at least*** $\sum_{j=1}^{d} \left|\sigma_{s_i+j}\right| \geq daw/2$ responding to this one challenge, or (2) we have $\left|\sigma_{s_i+j}\right| < aw/2$ for some $j \leq d$ and with probability at least $f' - \mathsf{negl}(w)$ we will incur cumulative memory cost at least $\Omega(Cw)$ before round $s_{i+1} + t_{i+1}$ when we finish responding the next challenge for node ℓ_{i+1}. Because G is (a, C, f')-ancestrally robust, in case 2 it seems intuitive to expect that given an initial state σ_{s_i} of size $|\sigma_{s_i}| < aw/2$ and a uniformly random challenge $v \in [N]$ (selected after σ_{s_i} is fixed) that with probability $f' - \mathtt{negl}(w)$ our PROM algorithm $\mathcal{A}$ will incur cumulative memory cost at least $\Omega(Cw)$ to compute label X_v. While this conjecture seems plausible, it does not appear possible to prove the conjecture holds using current PROM analysis techniques.

However, we can still define $s(i) = \min\{j : \ell_i - 1 \in P_j\}$ as the first round when we place a pebble on node $\ell_i - 1 = N + i - 1$. In the PROM trace we will have $s(i) = s_i$ where s_i denotes the first round where the label $X_{\ell_i - 1}$ appears as an output to one of the random oracle queries i.e., the first round where the challenge $r(i)$ is revealed. We can also define $t(i) = \min\{j \ : \ r(i) \in P_{s(i)+j}\}$ as the time to respond to challenge $r(i)$ by placing a pebbling on node $r(i)$ — in the PROM trace this will be the first round $s(i) + j$ where the label $X_{r(i)}$ is generated as the output to a random oracle query.

Our argument now proceeds as follows: If $\left|\sigma_{s(i)}\right| \geq ew/2$ then this challenge contributes to the sustained space complexity. Otherwise, if $\left|\sigma_{s(i)}\right| < ew/2$ then with probability at least $f' - \mathtt{negl}(w)$ the time to respond to the ith challenge $r(i)$ is at least $t(i) \geq d$. We now consider two subcases when $t(i) \geq d$. **Subcase 1:** the time to respond to the ith challenge $r(i)$ is at least $t(i) \geq d$ and $\left|\sigma_{i+j}\right| \geq aw/2$ for all $j \leq d$. In this subcase we incur cumulative memory cost at least $daw/2$ to respond to this one challenge. **Subcase 2:** the time to respond to the ith challenge is at least d, but $\left|\sigma_{i+j}\right| < aw/2$ for some $j \leq d$. In this subcase we have $\left|P_{i+j}\right| \leq a$. In this subcase we can argue that the cumulative memory cost incurred while responding to the next k challenges will be lower bounded by $w \cdot \mathsf{cc}\left(\mathsf{Ancestors}([N+i+1, N+i+k+1], G - P_{i+j})\right)/2$.

It would be tempting to assert that if $j < s_i$ then P_j (the configuration of the ex post facto pebbling at time j) and $r(i)$ (the ith challenge) are independent because the ith $r(i)$ is random and unknown during round j. If this were true then we could appeal to (a, C, f')-ancestral robustness to argue that, except with probability $\approx (1 - f')^k$, we will have $\mathsf{cc}\left(\mathsf{Ancestors}(r(i+z), G - P_j)\right) \geq C$ for at least one of the next k challenges $1 \leq z \leq k$. Unfortunately, this intuition

is incorrect for a subtle, but important, reason. The PROM attacker $\mathcal{A}$ can adapt its behavior after the ith challenge $r(i)$ is revealed in round s_i and the definition of the set P_j is dependent on the random oracle queries made in future rounds after rounds j or s_i. Therefore, the key analysis task in subcase 2 is to prove that, for a suitable parameter k, the ex-post facto graph $G_{H,X}$ will (with high probability) satisfy the following property: for all subsets $S \subseteq V$ of size $|S| \leq a$ and all length k intervals $[N+j, N+j+k-1]$ we have $\mathsf{cc}\left(\mathsf{Ancestors}([N+j, N+j+k-1], G-S)\right) \geq C$.

Because we cannot assume that our particular set P_j is independent of the next k challenges we instead union bound over all $\binom{N}{a}$ possible sets subsets S of size a as well as all $i \leq N_{\mathsf{chal}}$. Thus, except with probability $\binom{N}{a}(1-f')^k$, we will have $\mathsf{cc}\left(\mathsf{Ancestors}([N+i, N+i+k-1], G_{H,X}-S)\right) \geq C$ for ***any*** $i \leq N_{\mathsf{chal}}$ and ***any*** subset S of size $|S| \leq a$ — see Lemma 6. To ensure that this property holds with high probability we need to pick $k = \Theta(a \log N)$ such that $fk \geq a \log N + \log N_{\mathsf{chal}}$. If this property holds then in subcase 2 we can argue that the cumulative memory cost incurred while responding to the next k challenges will be $\Omega(wC)$. The amortized cost per challenge is $\Omega(wC/k)$ leading to a CMC penalty of $\Omega(Nw \cdot \min\{da, C/k\})$ for computations that do not have maximum sustained space complexity.

Equipped with both ex post facto pebblings and the time-space trade-offs for dynamic edges, we may prove the SSC/CC trade-off for dynamic pebbling graphs $\mathcal{G} = \mathsf{Dynamize}(G, 2N_{\mathsf{chal}})$, where $G = ([N], E)$ is (e, d, f)-fractionally depth-robust and (a, C, f')-ancestrally robust. Among the N_{chal} challenges, we can show that (with high probability) there are at least $N_{\mathsf{chal}}{}' = (f - \varepsilon')N_{\mathsf{chal}}$ challenge nodes $i_1, \ldots, i_{N_{\mathsf{chal}}{}'}$ in which ***at least*** one of the following hold:

1. **High Space:** The space usage at the start s_{i_j} of challenge i_j is at least $\left|\sigma_{s_{i_j}}\right| \geq c_{\mathsf{word}} we$.
2. **Long Time:** The time t_{i_j} $\mathcal{A}$ takes to answer challenge i_j is at least $t_{i_j} \geq d$.

See Theorem 5 for a more formal statement.

Now we let $\mathtt{costly} = \{i_1, \ldots, i_{N_{\mathsf{chal}}{}'}\}$ denote the set of costly challenges and let $Z_1 \subseteq \mathtt{costly}$ denote the set $Z_1 = \{j \ : \ \left|\sigma_{s_{i_j}}\right| \geq c_{\mathsf{word}} we\}$. If $|Z_1| \geq N_{\mathsf{chal}}{}'/3$ (i.e., the first case holds for at least $N_{\mathsf{chal}}{}'/3$ challenges) then the sustained space complexity is large. In particular, there are ***at least*** $|Z_1| = \Omega(N_{\mathsf{chal}})$ rounds where the space usage is at least $c_{\mathsf{word}} we$. Otherwise, if $|Z_1| < N_{\mathsf{chal}}{}'/3$ then second case holds for at least $2N_{\mathsf{chal}}{}'/3$ rounds. Let $Z_2 = \{j \ : \ t_{i_j} \geq d \wedge \left|\sigma_{s_{i_j}+r}\right| \geq aw/2 \forall r \leq d\}$ denote the subset of of challenge nodes i_j for which $t_{i_j} \geq d$ and we sustain space $aw/2$ while responding to that challenge. Thus, we have

$$\mathsf{cmc}(\mathsf{trace}(\mathcal{A}, H, x)) = \sum_{i=1}^{t} |\sigma_i| \geq \sum_{i \in Z_2} \sum_{j=s_i}^{s_i+d} \left|\sigma_{s_i+j}\right| \geq \sum_{i \in Z_2} \sum_{j=s_i}^{s_i+d} \frac{aw}{2} \geq \frac{|Z_2| aw}{2} \ .$$

If $|Z_2| \geq N_{\mathsf{chal}}{}'/3$ then the cumulative pebbling cost is $\mathsf{cmc}(\mathsf{trace}(A, H, X)) = \Omega(N_{\mathsf{chal}} daw)$. Finally, if $|Z_1| < N_{\mathsf{chal}}{}'/3$ and $|Z_2| < N_{\mathsf{chal}}{}'/3$ then there are at least

$|Z_3| \geq N_{\mathsf{chal}}'/3$ remaining challenges in $Z_3 = \texttt{costly} \setminus (Z_1 \cup Z_2)$. For each $i_j \in Z_3$ we must have $t_{i_j} \geq d$ (since $i_j \notin Z_1$) and $\left|\sigma_{s_{i_j}+r}\right| \leq aw/2$ for some $r \leq d$ (since $i_j \notin Z_2$) i.e., we drop below space $aw/2$ while responding to this challenge. We can now find a subset $Y \subseteq Z_3$ of $|Y| \geq |Z_3|/k$ challenges such that $\left|c - c'\right| > k$ for all challenges $c, c' \in Y$ i.e., the challenges in Y are well spread apart so that the intervals $[c, c+k]$ and $[c', c'+k]$ do not overlap. Thus, if $|Z_3| \geq N_{\mathsf{chal}}/3$ we now have

$$\begin{aligned}
\mathsf{cmc}(\mathsf{trace}(A,H,X)) &\geq \sum_{i \in Y} \sum_{j=s_{i+1}}^{s_{i+k}+t_{i+k}} \left|\sigma_j\right| \geq \sum_{i \in Y} \sum_{j=s_{i+1}}^{s_{i+k}+t_{i+k}} \left|P_j\right| c_{\mathsf{word}} w \\
&\geq c_{\mathsf{word}} w |Y| \min_{S,i:|S| \leq a \wedge i \leq N_{\mathsf{chal}}} \mathsf{cc}\left(\mathsf{Ancestors}([N+i, N+i+k-1, G-S)\right) \\
&\geq c_{\mathsf{word}} w |Y| C \geq c_{\mathsf{word}} w \frac{N_{\mathsf{chal}}'}{3k} C = \Omega\left(\frac{w N_{\mathsf{chal}} C}{a \log N}\right) .
\end{aligned}$$

If $|Z_1| \leq N_{\mathsf{chal}}'/3$ then the CMC penalty is at least

$$\mathsf{cmc}(\mathsf{trace}(A,H,X)) = \Omega\left(N_{\mathsf{chal}} w \cdot \min\left\{da, \frac{C}{a \log N}\right\}\right) .$$

Otherwise, if $|Z_1| \geq N_{\mathsf{chal}}'/3 = \Omega(N_{\mathsf{chal}})$ the sustained space-complexity is asymptotically optimal.

2.4 Our Construction: Extremely Grate Sample

Now that we know that we can use ancestrally/fractionally robust graphs G to construct dMHFs $f_{G,\mathsf{Dynamize}(G,N_{\mathsf{chal}})}$ which satisfy strong sustained space/cumulative complexity trade-offs in both the dynamic pebbling model as well as in the parallel random oracle model, we may proceed in constructing explicit examples of such dMHFs.

Our static component $\mathsf{EGSample}(N, \varepsilon)$ has two components:

1. An $(N, N^{1-\varepsilon}, f(\varepsilon))$-fractionally depth-robust graph on N nodes called $\mathsf{Grates}(N, \varepsilon)$ [Sch83] for any chosen $0 < \varepsilon < 1$. Here, $f \geq 0$ is a constant that depends only on ε.
2. An $(N/m, N^2/m, f')$-ancestrally robust graph DRS on N nodes sampled from the randomized algorithm $\mathsf{DRSample}(N)$ [ABH17] for all $m = \Omega(\log^2 N)$ sufficiently large, except with negligible probability.

To get the graph EGSample, we first simply take the union of DRSample and Grates. The problem, however, is that it's ideal in practice that the graphs defining our MHFs have indegree two. Union of DRSample and Grates has indegree three, so we must apply a standard technique called *indegree reduction*. See Fig. 3 for a summary of the EGSample construction. Our analysis of $\mathsf{DRSample}(N)$ (Definition 14) involves showing that when we sample a

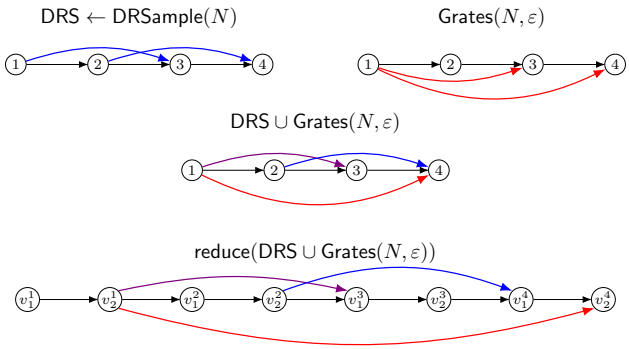

Fig. 3. The figure above describes how to construct EGS_N, where $N = 4$. First, we sample DRS from the randomized algorithm $\mathsf{DRSample}(N)$ (see Definition 14). The next component is Grates (see Theorem 11). We then take their union, which we then prove (see Theorem 9) has the desired robustness properties. The cross edges of DRSample are blue and are red for Grates. In their union, we color the edge $(1, 3)$ purple since they have it in common. Finally, we perform indegree reduction (see Definition 15). Here, we have kept the color-coding of the edges. For example, the edge (v_2^1, v_2^4) corresponds to the edge $(1, 4)$ in Grates.

static graph $G \sim \mathsf{DRSample}(N)$ that the corresponding "metagraph" (Definition 10) G_m of G is "extremely depth-robust" with high probability. Thus, we call our (static) construction "Extremely Grate Sample" or "EGSample" because it combines grates with the extremely depth-robust metagraphs from DRSample. We will defer the analysis of $\mathsf{DRSample}(N)$ to Section 2.5. In the end, we describe a randomized algorithm $\mathsf{EGSample}(N, \varepsilon)$ (Definition 12) that, except with negligible probability, outputs a static graph G that satisfy both (up to constant factors) $(N, N^{1-\varepsilon}, f)$-fractionally depth-robust and $(N/m, N^2/m, f')$-ancestrally robust.

In Definition 13, we present the randomized algorithm $\mathsf{DEGSample}(N, N_{\mathsf{chal}}, \varepsilon)$. This randomized algorithm outputs the dynamic pebbling graph $\mathsf{DEGS} = \mathsf{Dynamize}(\mathsf{EGS}, 2N_{\mathsf{chal}}, \varepsilon)$, where the static graph $\mathsf{EGS} \leftarrow \mathsf{EGSample}(N, \varepsilon)$ is sampled from EGSample. Naturally, we call the dynamic pebbling graphs output by $\mathsf{DEGSample}$ "dynamic extremely grate sample" or "DEGSample."

Using Theorem 1, we can easily use our above results for EGSample to prove strong SSC/CC bounds for DEGSample in the dynamic pebbling model. We will assume that $N_{\mathsf{chal}} = N$. Let $\mathsf{DEGS} \leftarrow \mathsf{DEGSample}(N, N, \varepsilon/2) = \mathsf{Dynamize}(\mathsf{EGS}, 2N)$. Since EGS is $(\Omega(N/m), \Omega(N^2/m), f)$-ancestrally depth-robust (with high probability) when $m = \Omega(\log^2 N)$ Theorem 9), we can easily choose the most suitable assignment for our analysis. For the pebbling analysis, we will consider $m = O\left(N^{\varepsilon/2}\right)$. Combining Theorem 1 with these robustness properties of EGSample, we have that except with negligible probability, any dynamic pebbling strategy on DEGS either:

– sustains $\Omega(N)$ pebbles for $\Omega(N)$ steps, or
– has cumulative complexity $N \cdot \min\left\{N/m \cdot N^{1-\varepsilon/2}, N^2/m\right\} = \Omega(N^{3-\varepsilon})$.

This is near-optimal as the CC penalty cannot exceed $O(N^3)$ [BH22].

For the PROM SSC/CMC trade-off, we will apply Theorem 2. This time, however, we will consider $\mathsf{DEGS} \leftarrow \mathsf{DEGSample}(N, 2N, \varepsilon'/2)$, where $\varepsilon' = 0.99\varepsilon$, for example. Observe that ancestral robustness is a monotone property in the sense that if a graph G is (a', C, f')-ancestrally robust then it is also (a, C, f')-ancestrally robust for any $a \leq a'$. Thus, when $m = \Omega(\log^2 N)$ the DAG EGS is $(a, N^2/m, f)$-ancestrally depth-robust for any $a \leq N/m$ (with high probability). In particular, we can set $a = \sqrt{N}$ and $m = \Theta(N^\epsilon)$ and $d = N^{1-\epsilon}$ so that the CMC penalty is $\Omega\left(\min\{Ndaw, \frac{NwC}{a \log N}\}\right) = \Omega\left(N^{2.5-\epsilon} w / \log N\right)$.

2.5 Constructing Ancestrally Robust Graphs

So far, we have shown that if we can find families of highly ancestrally robust graphs, then we can construct secure dMHFs in both the dynamic pebbling model as well as the stronger parallel random oracle model. We show that a previous construction called DRSample [ABH17] satisfies this property. DRSample is a randomized algorithm which outputs a DAG $G \leftarrow \mathsf{DRSample}(N)$ with N nodes and maximum indegree 2. Alwen et al. proved that (with high probability) the DAG G will be $(c_1 N/\log N, c_2 N)$-depth robust i.e., for all subsets S containing at most $|S| \leq c_1 N/\log N$ nodes the graph $G - S$ still contains a directed path of length d. This immediately implies that $\mathsf{cc}(G) \geq c_1 c_2 N^2/\log N$ because because Alwen et al. [ABP17] proved that $\mathsf{cc}(G) \geq ed$ for any (e, d)-depth robust graph G. While the DRSample [ABH17] construction is not new, we do require a novel analysis of this construction to establish ancestral robustness as this property is stronger than depth-robustness.

We show that graphs output from the randomized algorithm DRSample are almost maximally ancestral robust with high probability. In particular, except with negligible probability, $\mathsf{DRS} \leftarrow \mathsf{DRSample}(N)$ is $(\Omega(N/m), \Omega(N^2/m), f')$-ancestrally robust when $m = \Omega(\log^2 N)$ Theorem 10. To show that graphs output from DRSample are ancestrally robust, we consider a combinatorial property called *local expansion* [EGS75] (see Definition 9), which quantifies the connectivity of a single node in relation to the entire graph. Intuitively a node v in G is a δ local expander if, for all k, every δk-sized set of predecessors $A \subseteq [v-k-1 : v-1]$ and successors $B \subseteq [v : v+k]$ has an edge from a node in A to a node in B in G. We say that G is a δ-local expander if every node in G is a δ-local expander. Using some known facts about local expanders, it is relatively straightforward to prove that δ-local expansion implies ancestral robustness, although we will see that this will not suffice for our applications.

Lemma 1 (Local Expander Facts (informal [ABP18])). *If $G = ([N], E)$ is a δ-local expander for some $\delta < 1/10$, then*

1. *G is $(cN, (1 - c - 10\delta)N)$-depth robust for all $c > 0$ (see Definition 11), meaning it has cumulative complexity $c^2 N^2$ (see Lemma 2).*

2. *For any set $S \subseteq [N]$, there exists a subset of nodes U of size at most $N - 2|S|$ in $G - S$ such that for every $u < v \in U$ there is a path from u to v.*

Now, suppose $G = ([N], E)$ is a δ-local expander with $\delta < 1/10$ and fix a set $S \subset [N]$ of size at most pN, where $0 < p < 1/2$ is some constant to be chosen later. By Lemma 1, we see that there is a set of $(1-2p)N$ nodes $U = \{v_1, \ldots, v_{(1-2p)N}\}$ such that for all $i < j \leq (1-2p)N$ there is a directed path from v_i to v_j. This means that for all $j \leq (1-2p)N$ we have $\mathsf{Ancestors}(v_j, G-S) \supset \{v_1, \ldots, v_{j-1}\}$. Let $L_r = \{v_{(1-2p)N-r+1}, \ldots, v_{(1-2p)N}\}$ denote the final r nodes in U and let $F_r = \{v_1, \ldots, v_r\}$ denote the topologically first r nodes in U. Note that for each $v \in L_{c'N}$ we have $\mathsf{Ancestors}(v, G-S) \supset F_{(1-2p)N-c'N}$. Now we can argue that the graph $H = G - \left(V \setminus F_{(1-2p)N-c'N}\right)$, obtained by removing all nodes except for those in $F_{(1-2p)N-c'N}$, is itself depth robust. To see this note that $\left|V \setminus F_{(1-2p)N-c'N}\right| = N - ((1-2p)N - c'N) = 2pN + c'N$. Recall that the original graph G is $(cN, (1-c-10\delta)N)$-depth robust and we have only removed $2pN+c'N$ nodes to obtain the subgraph H. As long as $2pN+c'N \leq cN/2$ we can still remove up to $cN - 2pN - c'N \geq cN/2$ additional nodes from H and there will still be a directed path of length $(1-c-10\delta)N$ i.e., for any $S' \subseteq V(H)$ of size $|S'| \leq cN/2$ we have $\mathsf{depth}(H - S') \geq (1-c-10\delta)N$. Because H is $(cN/2, (1-c-10\delta)N)$-depth robust it follows that $\mathsf{cc}(H) \geq N^2 c(1-c-10\delta)$. This implies that for all $v \in L_{c'N}$ we have

$$\mathsf{cc}\left(\mathsf{Ancestors}(v, G-S)\right) \geq \mathsf{cc}(H) \geq N^2 c(1-c-10\delta) = \Omega(N^2) \ .$$

Therefore, as long as we assign p and c' small enough with respect to our other chosen parameter c (which is always possible when $\delta < 1/10$), we have shown that a δ-local expander is an $\big(\Omega(N), \Omega(N^2), \Omega(1)\big)$-ancestrally robust!

Unfortunately, it turns out that δ-local expansion is simply too strong of a property for DRSample to hold; Indeed, it is impossible to construct a δ-local expander with maximum indegree $O(1)$. Instead we analyze a graph G_m called the *metagraph* (Definition 10) of $G \leftarrow \mathsf{DRSample}(N)$ and argue that (with high probability) G_m is a δ-local expander when metanode size parameter $m = \Omega(\log^2 N)$; see Lemma 13 for details. We show that to achieve strong ancestral robustness, it suffices for the *metagraph* G_m of G to be a local expander (see Theorem 3). Together, these results allow us to argue that the original graph is (a, C, f)-ancestrally robust with $a = \Omega(N/m)$, $C = \Omega(N^2/m)$ and $f = \Omega(1)$ in Theorem 10.

The metagraph G_m has $N' = N/m$ nodes, and each node $v \in V(G_m)$ corresponds to the interval of nodes $[v(m-1)+1 : vm]$ from $V(G) = [N]$. See Fig. 4 for more intuition or Definition 10 for the formal definition. There is an edge (u, v) in G_m if and only if there is an edge in G from a node in the "last part" of the interval corresponding to u to a node in the "first part" of the interval corresponding to v (see Definition 10 for a formal definition). Thus, nodes in G_m may have indegree $O(m)$. The goal is to prove strong combinatorial properties of the "stronger" G_m and then relate these properties to the sparser G. The

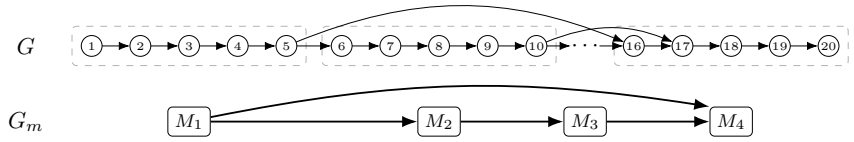

Fig. 4. Shown is a graph $G = ([20], E)$ and its metagraph G_m with $m = 5$. Since $m = 5$ we have that $M_1^{\mathsf{first}} = \{1\}$ and $M_1^{\mathsf{last}} = \{5\}$, and similarly for the three remaining metanodes. Notice that G has two cross edges $(5, 16)$ and $(10, 17)$. The corresponding metagraph G_m has an edge (M_1, M_4) because there $10 \in M_1^{\mathsf{last}}$ and $16 \in M_4^{\mathsf{first}}$. The other edge, $(10, 17)$ does *not* induce a meta edge because, while 10 is in the last part of M_2, the node 17 is not in the last part of M_4.

authors of [ABH17] used metagraphs to show that DRSample (which is a component of our EGSAmple) has cumulative complexity $\Omega(N^2/\log N)$. In particular, they showed that (with high probability) the metagraph G_m satisfied a weaker version of δ-local expansion when $m = \Theta(\log N)$ — the version required that ***most*** (not all) nodes $v \in V(G_m)$ satisfy δ-local expansion around that node. This was enough argue that G_m is $(\Omega(N/m), \Omega(N/m))$-depth robust. If G_m is $(\Omega(N/m), \Omega(N/m))$-depth robust, this implies that that G is $(\Omega(N/m), \Omega(N))$-depth robust and therefore $\mathsf{cc}(G) = \Omega\left(N^2/m\right)$.

Unfortunately, the properties implied by these weak local expanders and the connections between metagraphs and graphs appear to be insufficient towards our goal of showing that DRSample (and other graphs) are ancestrally robust. For this reason, we do not use any prior results on metagraphs and instead establish tighter relationship between a graph G and it's metagraph G_m.

We show that when $m = \Omega(\log^2 N)$ that (with high probability) the metagraph G_m satisfies δ-local expansion for ***all*** nodes in the metagraph. We then prove the following relationship between G_m and G; see Theorem 8 for a more detailed statement.

Theorem 3 (Meta local Expansion to Ancestral Robustness (Informal)). *If a metagraph $G_m = ([N/m], E_m)$ of a graph $G = ([N], E)$ is a δ-local expander with $\delta < 1/10$, then G is $(\Omega(N/m), \Omega(N^2/m), \Omega(1))$-ancestrally robust.*

Lemma 11 lays out the three core connections between graphs and their metagraphs that are used to prove Theorem 3. The most fundamental connection is that for every set of nodes $S \subseteq [N]$, there is a unique set $S' \subseteq [N/m]$ of metanodes in which $G \ominus S = G_m \ominus S'$[5] The set S' is precisely the set of meta nodes that "overlap" with the nodes in S. With this association, it is therefore possible to map paths in $G_m - S'$ to paths in $G_m - S$. With this granularity, we can work back and forth between graph and metagraph. Since G_m is a local expander, by the analysis at the beginning of this section, there are $c'N/m$ nodes $v_1, \ldots, v_{c'N/m}$ in $G_m \ominus S'$ whose ancestors are $(\Omega(N/m), \Omega(N/m))$-depth robust

[5] here, $G \ominus S$ denotes the graphs G, except without any incoming or outgoing edges of the nodes in S.

i.e., for all $j \leq c'N/m$ the graph $\mathsf{Ancestors}(v_j, G_m \ominus S')$ is $(\Omega(N/m), \Omega(N/m))$-depth robust. By the association in Lemma 11, it follows that for each of these $c'N/m$ nodes in $G_m \ominus S'$, there are $\Omega(m)$ unique nodes in $G - S$ whose ancestors are $(\Omega(N/m), \Omega(N))$-depth robust. In particular, the node v_j in G_m corresponds to the interval $[v_j(m-1)+1 : v_j m]$ in $V(G) = [N]$ and for any $x \in [v_j m - m/5 : v_j m]$ the graph $\mathsf{Ancestors}(v, G-S)$ will be $(\Omega(N/m), \Omega(N))$-depth-robust. Thus, there are $\Omega(mN/m)$ total nodes $v \in V(G-S)$ such that $\mathsf{Ancestors}(v, G-S)$ is $(\Omega(N/m), \Omega(N))$-depth-robust. Finally, by Lemma 2, we have that the ancestors of these nodes have cumulative complexity $\mathsf{cc}(\mathsf{Ancestors}(v, G-S)) = \Omega(N^2/m)$, proving Theorem 3.

2.6 Analysis of Argon2id

Argon2 [BDK16] was selected the winner of the password hashing competition in 2015[pas15], and has been deployed in cryptographic libraries such as Libsodium [Den]. Currently, the hybrid mode Argon2id is recommended for password hashing. Oversimplifying slightly (see Remark 2 in Section I) we can view Argon2id as $\mathsf{Dynamize}(\mathsf{Argon2i}_{N/2}, N/2)$ where the static graph $\mathsf{Argon2i}_{N/2}$ is sampled using a priori fixed salt values. Adapting analysis from [BZ17] we show that if $m = \Omega(\sqrt{N \log N})$ that the metagraph G_m of $\mathsf{Argon2i}_{N/2}$ is the complete graph (see Lemma 41). This allows us to immediately apply our prior analysis (Lemma 12) to conclude that $\mathsf{Argon2i}_{N/2}$ is $(N/(4m), 3N^2/(5 \cdot 4^2 m), 1/5)$-ancestrally robust. Blocki and Zhou [BZ17] already proved that (whp) $\mathsf{Argon2i}_{N/2}$ is $(e, \ell_N \frac{N^3}{e^3}, f = O(1))$-fractionally depth robust for any $e = \Omega(N^{2/3})$ — see Theorem 15. Thus, we can immediately apply our main results in Theorem 4 and Theorem 7 to obtain SSC/CMC tradeoffs in the dynamic pebbling model and in the PROM (see Theorem 14). In particular, for any constants $2/3 \leq \beta < 1$ and $\varepsilon > 0$ any dynamic pebbling strategy will (whp) either (1) sustains N^β pebbles for $\Omega(N)$ steps or (2) have cumulative pebbling cost penalty $\Omega(N^{4.5-3\beta-\epsilon})$. Suppose we set $\beta = 3/4$ and $\varepsilon = 0.05$. Then we get (up to polylog factors) that any strategy sustains $N^{0.75}$ pebbles for N steps, or incurs CC $N^{2.2}$. In the PROM model we have to further constrain $\beta < \frac{5}{6} - \epsilon$ and the cumulative memory cost penalty $\Omega\left(N^{3.25-3\beta/2-\varepsilon/2}\right)$ is a bit lower — see Theorem 13. Again, using the example of $\beta = 3/4$ and $\varepsilon = 0.05$, we see that any PROM algorithm computing the MHF corresponding to $\mathsf{Argon2iDyn}_N$ either (up to polylog factors) sustains $wN^{0.75}$ space for N steps, or has cumulative complexity $wN^{2.1}$. [BH22] previously presented a SSC/CMC tradeoff for Argon2id in the dynamic pebbling model, but their analysis had a subtle bug.[6] Thus, we give the first non-trivial SSC/CMC for Argon2id in both the dynamic pebbling model and the PROM.

[6] The bug in [BH22] relates to the fact that the static graph of Argon2id is depth-robust along a curve $(e, d(e))$, but they implicitly assumed this for values of e outside of the bounds for which this is true.

References

AB16. Joël Alwen and Jeremiah Blocki. Efficiently computing data-independent memory-hard functions. In Matthew Robshaw and Jonathan Katz, editors, ***Advances in Cryptology – CRYPTO 2016, Part II***, volume 9815 of ***Lecture Notes in Computer Science***, pages 241–271. Springer, Berlin, Heidelberg, August 2016.

AB17. Joël Alwen and Jeremiah Blocki. Towards practical attacks on Argon2i and balloon hashing. In ***2017 IEEE European Symposium on Security and Privacy***, pages 142–157. IEEE Computer Society Press, April 2017.

ABH17. 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: 24th Conference on Computer and Communications Security***, pages 1001–1017. ACM Press, October / November 2017.

ABP17. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Depth-robust graphs and their cumulative memory complexity. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, ***Advances in Cryptology – EUROCRYPT 2017, Part III***, volume 10212 of ***Lecture Notes in Computer Science***, pages 3–32. Springer, Cham, April / May 2017.

ABP18. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Sustained space complexity. In Jesper Buus Nielsen and Vincent Rijmen, editors, ***Advances in Cryptology – EUROCRYPT 2018, Part II***, volume 10821 of ***Lecture Notes in Computer Science***, pages 99–130. Springer, Cham, April / May 2018.

ACP+17. Joël Alwen, Binyi Chen, Krzysztof Pietrzak, Leonid Reyzin, and Stefano Tessaro. Scrypt is maximally memory-hard. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, ***Advances in Cryptology – EUROCRYPT 2017, Part III***, volume 10212 of ***Lecture Notes in Computer Science***, pages 33–62. Springer, Cham, April / May 2017.

AGK+18. Joël Alwen, Peter Gazi, Chethan Kamath, Karen Klein, Georg Osang, Krzysztof Pietrzak, Leonid Reyzin, Michal Rolinek, and Michal Rybár. On the memory-hardness of data-independent password-hashing functions. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, ***ASIACCS 18: 13th ACM Symposium on Information, Computer and Communications Security***, pages 51–65. ACM Press, April 2018.

AS15. Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, ***47th Annual ACM Symposium on Theory of Computing***, pages 595–603. ACM Press, June 2015.

BC21. Jeremiah Blocki and Mike Cinkoske. A new connection between node and edge depth robust graphs. In James R. Lee, editor, ***ITCS 2021: 12th Innovations in Theoretical Computer Science Conference***, volume 185, pages 64:1–64:18. Leibniz International Proceedings in Informatics (LIPIcs), January 2021.

BCLS22. Jeremiah Blocki, Mike Cinkoske, Seunghoon Lee, and Jin Young Son. On Explicit Constructions of Extremely Depth Robust Graphs. In Petra Berenbrink and Benjamin Monmege, editors, ***39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)***, volume

219 of ***Leibniz International Proceedings in Informatics (LIPIcs)***, pages 14:1–14:11, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

BDK16. Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. Argon2: New generation of memory-hard functions for password hashing and other applications. In ***2016 IEEE European Symposium on Security and Privacy***, pages 292–302. IEEE Computer Society Press, March 2016.

BH22. Jeremiah Blocki and Blake Holman. Sustained space and cumulative complexity trade-offs for data-dependent memory-hard functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, ***Advances in Cryptology – CRYPTO 2022, Part III***, volume 13509 of ***Lecture Notes in Computer Science***, pages 222–251. Springer, Cham, August 2022.

BHK+19. Jeremiah Blocki, Benjamin Harsha, Siteng Kang, Seunghoon Lee, Lu Xing, and Samson Zhou. Data-independent memory hard functions: New attacks and stronger constructions. In Alexandra Boldyreva and Daniele Micciancio, editors, ***Advances in Cryptology – CRYPTO 2019, Part II***, volume 11693 of ***Lecture Notes in Computer Science***, pages 573–607. Springer, Cham, August 2019.

BHZ18. Jeremiah Blocki, Benjamin Harsha, and Samson Zhou. On the economics of offline password cracking. In ***2018 IEEE Symposium on Security and Privacy***, pages 853–871. IEEE Computer Society Press, May 2018.

BK16a. Alex Biryukov and Dmitry Khovratovich. Egalitarian computing. In Thorsten Holz and Stefan Savage, editors, ***USENIX Security 2016: 25th USENIX Security Symposium***, pages 315–326. USENIX Association, August 2016.

BK16b. Alex Biryukov and Dmitry Khovratovich. Equihash: Asymmetric proof-of-work based on the generalized birthday problem. In ***ISOC Network and Distributed System Security Symposium – NDSS 2016***. The Internet Society, February 2016.

BLRZ24. Jeremiah Blocki, Peiyuan Liu, Ling Ren, and Samson Zhou. Bandwidth-hard functions: Reductions and lower bounds. ***Journal of Cryptology***, 37(2):16, April 2024.

BRZ18. Jeremiah Blocki, Ling Ren, and Samson Zhou. Bandwidth-hard functions: Reductions and lower bounds. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, ***ACM CCS 2018: 25th Conference on Computer and Communications Security***, pages 1820–1836. ACM Press, October 2018.

BS25. Jeremiah Blocki and Nathan Smearsoll. Provably memory-hard proofs of work with memory-easy verification. TCC 2025 (to appear), 2025.

BZ17. Jeremiah Blocki and Samson Zhou. On the depth-robustness and cumulative pebbling cost of Argon2i. In Yael Kalai and Leonid Reyzin, editors, ***TCC 2017: 15th Theory of Cryptography Conference, Part I***, volume 10677 of ***Lecture Notes in Computer Science***, pages 445–465. Springer, Cham, November 2017.

CT19. Binyi Chen and Stefano Tessaro. Memory-hard functions from cryptographic primitives. In Alexandra Boldyreva and Daniele Micciancio, editors, ***Advances in Cryptology – CRYPTO 2019, Part II***, volume 11693 of ***Lecture Notes in Computer Science***, pages 543–572. Springer, Cham, August 2019.

Den. Frank Denis. Libsodium documentation: Password hashing. `https://doc.libsodium.org/password_hashing` (Retrieved 10/1/2025).

EGS75. Paul Erdoes, Ronald L. Graham, and Endre Szemeredi. On sparse graphs with dense long paths. Technical report, Stanford, CA, USA, 1975.
HPM15. George Hatzivasilis, Ioannis Papaefstathiou, and Charalampos Manifavas. Password hashing competition - survey and benchmark. Cryptology ePrint Archive, Report 2015/265, 2015.
HPV77. John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. ***Journal of the ACM (JACM)***, 24(2):332–337, 1977.
Lee11. Charlie Lee. Litecoin, 2011.
LT79. Thomas Lengauer and Robert Endre Tarjan. Upper and lower bounds on time-space tradeoffs. In ***Proceedings of the eleventh annual ACM symposium on Theory of computing***, pages 262–277, 1979.
pas15. Password hashing competition, 2015. `https://www.password-hashing.net/`.
Per09. Colin Percival. Stronger key derivation via sequential memory-hard functions, 2009.
Pip77. Nicholas Pippenger. Superconcentrators. ***SIAM Journal on Computing***, 6(2):298–304, 1977.
RD17. Ling Ren and Srinivas Devadas. Bandwidth hard functions for ASIC resistance. In Yael Kalai and Leonid Reyzin, editors, ***TCC 2017: 15th Theory of Cryptography Conference, Part I***, volume 10677 of ***Lecture Notes in Computer Science***, pages 466–492. Springer, Cham, November 2017.
Ree17. Jeff Reed. ***Litecoin: An introduction to litecoin cryptocurrency and litecoin mining***. CreateSpace Independent Publishing Platform, 2017.
Sch83. G. Schnitger. On depth-reduction and grates. In ***2013 IEEE 54th Annual Symposium on Foundations of Computer Science***, pages 323–328, Los Alamitos, CA, USA, nov 1983. IEEE Computer Society.
scr25. The best scrypt miners, 2025. `https://www.hashrate.no/c/The_best_Scrypt_miners` (Retrieved 9/30/2025).
Wie04. Michael J. Wiener. The full cost of cryptanalytic attacks. ***Journal of Cryptology***, 17(2):105–124, March 2004.
Wil25. R. Ryan Williams. Simulating time with square-root space. In Michal Koucký and Nikhil Bansal, editors, ***57th Annual ACM Symposium on Theory of Computing***, pages 13–23. ACM Press, June 2025.

A Helpful Background

We introduce the necessary background on pebbling, graph properties to aid in pebbling analysis, and finally PROM algorithms.

A.1 Graph Pebbling

We now formally define dynamic pebbling graphs, which are a specific kind of distribution over DAGs.

Definition 2 (Dynamic Pebbling Graph [BH22]). *A dynamic pebbling graph $\mathcal{G}$ is a distribution of DAGs $G = ([N], E)$ with edges $E \supseteq \{(i,j) : 1 \leq i < j \leq N\}$. We say that an edge (i,j) is* ***static*** *if for all DAGs $G = (V, E)$ in the support of $\mathbb{G}$ we have $(i,j) \in E$ and we let $E_{\mathsf{static}} \subseteq \{(i,j) : 1 \leq i < j \leq N\}$*

denote the set of all ***static edges****. Some nodes* $D \subseteq [N]$ *have* ***dynamic edges****, which are determined by some random variables* $r(j)$ *for* $j \in D$ *over the set* $[j-1]$*. The* ***dynamic edges*** *of* G *are* $E^j_{\mathsf{Dynamic}} = \{(r(j), j) \mid j \in D\} = E \setminus E_{\mathsf{static}}$*. A node with a dynamic edge is called a* ***dynamic node****.*

The authors in [BH22] give a more general definition of dynamic pebbling graphs, where each node can have more than one dynamic edge. Since there are currently no dMHFs (including those we define in this paper) that take advantage of this property, we use Definition 2 for simplicity.

procedures for pebbling dynamic graphs are called *dynamic pebbling strategies*; such a procedure makes decisions what to pebble/unpebble based on outcomes of the dynamic edges.

Definition 3 (Dynamic Pebbling Strategy [BH22]). *A* ***dynamic pebbling strategy*** $\mathcal{S}$ *is a function that takes as input*

1. *an integer* $i \leq N$*,*
2. *an initial pebbling configuration* $P^i_0 \subseteq [i]$ *with* $i \in P^i_0$*, and*
3. *a partial graph* $G_{\leq i+1}$*,*

where the partial graph $G_{\leq i}$ *is the subgraph of* G *induced by the nodes* $1, \ldots, i$*. The output of* $\mathcal{S}(i, P^i_0, G_{\leq i+1})$ *is a legal sequence of pebbling moves* $P^i_1, \ldots, P^i_r$ *that will be used in the next phase to place a pebble on node* $i+1$*, so that* $i+1 \in P^i_{r_i} \subseteq [i+1]$*. Given* $G \sim \mathbb{G}$*, we let* $\mathcal{S}(G)$ *denote the sequence of pebbling moves* $\left\langle P^0_1, \ldots, P^0_{r0}, P^1_1, \ldots, P^{N-1}_{r_1}, \ldots, P^{N-1}_{r_{N-1}} \right\rangle$*. Here,* $P^i_1, \ldots, P^i_{r_i} = \mathcal{S}\left(i, P^i_0, G_{\leq i+1}\right)$*,* $P^i_0 = P^{i-1}_{r_{i-1}}$*, and* $P^0_0 = \emptyset$*. We call* $\mathcal{S}(G)$ *a* ***pebbling*** *(for* G*.)*

The constructions in this paper will all fall into the category of what we call *standard dynamic pebbling graphs*. A standard dynamic pebbling graph $\mathcal{G}$ has two static components: a graph $G = ([N], E)$ that is "high cost" in some way and a line graph $\mathsf{LineGraph}(N_{\mathsf{chal}})$ on N_{chal} nodes, with an edge connecting the last node of G to the first node of $\mathsf{LineGraph}(N_{\mathsf{chal}})$. The dynamic nodes of $\mathcal{G}$ consist of the nodes in the line graph, each with an incoming dynamic edge chosen uniformly at random from $[N]$, the nodes of G. We let $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ denote the standard dynamic pebbling graph for G with a line graph on N_{chal} nodes, as we are "making" G dynamic (see Definition 1).[7] For $j \in [N_{\mathsf{chal}}]$, we let $r(j)$ denote the node uniformly chosen from $[N]$ that forms a dynamic edge going to the jth node (topologically) in the line graph. We call $r(j)$ a "challenge" or "the jth challenge." The challenge is sampled and relayed to the pebbling strategy after the strategy pebbles all static parents of jth line graph node. To pebble the jth line graph node, the pebbling strategy must also pebble the challenge node $r(j)$.

Now we formally define the cost metrics we will be using to measure the memory-hardness of our pebblings.

[7] Almost all data-dependent memory-hard functions can be represented as $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ for some parameters G and N_{chal}. A notable exception is Argon2d/Argon2id [BZ17].

Definition 4 (Pebbling Complexity). *Let $\mathbb{G}$ be a dynamic pebbling graph, G be a graph in the sample space of $\mathbb{G}$, $\mathcal{P}$ be the set of legal pebblings of G, and $\mathcal{S}$ be the set of pebbling strategies for $\mathbb{G}$. We define the* ***cumulative complexity*** *of a*

- *pebbling $P = \langle P_1, \ldots, P_T \rangle \in \mathcal{P}$ as $\mathsf{cc}(P) = \sum_{i=1}^{T} |P_i|$,*
- *a sequence of pebbling moves $\langle P_i, \ldots, P_j \rangle$ as $\mathsf{cc}(P, i, j) = \sum_{k=i}^{j} |P_k|$*
- *graph G as $\mathsf{cc}(G) = \min_{P \in \mathcal{P}} \{\mathsf{cc}(P)\}$, and*

Likewise, we define the ***s-sustained space complexity*** *of a*

- *pebbling P as $\mathsf{ssc}(P, s) = \left|\{i \,||P_i| \geq s, i \in [T]\}\right|$,*
- *graph G as $\mathsf{ssc}(G, s) = \min_{P \in \mathcal{P}} \{\mathsf{ssc}(P, s)\}$, and*

A.2 Graph Properties

we now discuss some graph properties that are key to proving pebbling lower bounds and constructing MHF. The most fundamental property is *depth-robustness*. Intuitively, a graph has high depth-robustness if, even after removing many nodes from the graph, there is always a long path in the remaining graph.

Definition 5 (Depth-Robustness). *A DAG $G = ([N], E)$ is (e, d)-depth robust, if for any set $S \subseteq [N]$ of size at most e, $G - S$ contains a path of length d.*

Depth-robustness is useful because high depth-robustness implies high cumulative complexity.

Lemma 2 (Depth-Robustness and Cumulative Complexity [ABP17]). *If a DAG G is (e, d)-depth-robust, then $\mathsf{cc}(G) > ed$.*

An idea that we will use heavily is that an (e, d)-depth robust graph is still depth-robust after removing $< e$ nodes from the graph.

Lemma 3. *If G is (e, d)-depth robust, then for any set of nodes S, $G - S$ is $(e - |S|, d)$-depth robust.*

While depth-robustness says that there is always a path of length d in $G - S$, it's often useful to know that there are many nodes of depth at least d. This is called *fractional depth-robustness*. This property will be integral to our constructions.

Definition 6 (Fractional Depth-Robustness [BZ17]). *A DAG $G = [N]$ is (e, d, f)-fractionally depth-robust if for any set $S \subseteq [N]$ of size at most e, $G - S$ contains at least fN nodes with depth at least d.*

A.3 PROM Algorithms

In this section we'll overview our notion of PROM algorithms, which is essentially the same as [ACP+17]. The PROM algorithm A is deterministic (without loss of generality) and operates in rounds. Let $H : \{0,1\}^u \to \{0,1\}^w$ be an oracle for some large u. A state is a pair $(\tau, \mathbf{s})$ of data τ and tuple of strings $\mathbf{s}$. On round 0, A receives an initial state $\sigma_0 = \emptyset$, marking the start of round 1. At the end of round i, A produces an output state $\bar{\sigma}_i = (\tau_i, \mathbf{q}_i)$. $\mathbf{q}_i = [q_i^1, \ldots, q_i^{z_i}]$ is a batch of z_i queries to be made in parallel, and τ_i can be thought of as the ending configuration of A at round i. Round $i+1$ starts when A receives as input the pair $\sigma_i = (\bar{\sigma}_i, H(\mathbf{q}_i))$, where $H(\mathbf{q}_i) = [H(q_i^1), \ldots, H(q_i^{z_i})]$. As in [ACP+17], we require that A terminates. If A terminates after T queries, then its output at time T is $\bar{\sigma}_T = (\tau_i, \emptyset)$, and τ_i is taken as A^H's output on input X, denoted $A^H(X)$. The list $\mathsf{trace}(A, H, X) = (\sigma_1, \ldots, \sigma_t)$ is called the *trace* of A with oracle access to H on input X. With this definition, we can define cumulative memory complexity and sustained memory complexity in the PROM, corresponding to cumulative complexity and sustained space complexity in the pebbling model.

Definition 7 (PROM Algorithm Trace Cost). *Let A^H be a PROM algorithm with access to an oracle H. Let $(\sigma_1, \ldots, \sigma_t)$ be the trace of A^H on X. The cumulative memory complexity of A^H on X is*

$$\mathsf{cmc}(A, H, X) \coloneqq \sum_{1 \le i \le t} |\sigma_i| \,.$$

The s-sustained memory complexity (SMC) of A^H on X is

$$s\text{-}\mathsf{ssc}(A, H, X) = \left| \left\{ i \in [t] : \, |\sigma_i| \ge s \right\} \right|$$

For $a \le b \in [t]$, we let

$$\mathsf{cmc}(A, H, X, a, b) = \sum_{a \le i \le b} |\sigma_i|$$

and

$$s\text{-}\mathsf{ssc}(A, H, X, a, b) = \left| \left\{ i \in [a:b] \mid \, |\sigma_i| \ge s \right\} \right|$$

A.3.1 Computing Memory-Hard Functions Fix a (static) pebbling graph $G' = ([N], E)$. Such graphs G' (along with an oracles H) define data-independent memory-hard functions $f_{G'}^H$(iMHFs) by encoding data dependencies. Each node has a label; we essentially use the labeling convention of [AS15]. The label for node 1 is $H(X)$. The label for node v is the hash of v and the labels of its parents: $X_v = H(v \| X_{\mathsf{Parents}(v,G')})$, where for a set $S = \{s_1, \ldots, s_\ell\} \subseteq [N]$ (s_i sorted numerically), X_S denotes the concatenation $X_{s_1} \| \cdots \| X_{s_\ell}$. Finally, $f_{G'}^H(X) = X_N$.

Now we will move on to data-dependent memory-hard functions (dMHFs) which are defined with respect to dynamic pebbling graphs. When defining dMHFs, we have to be careful about how we are converting our dynamic pebbling graphs into functions with respect to oracles $H : \{0,1\}^u \to \{0,1\}^w$, because the dynamic edges are going to be determined by a string $s \sim \{0,1\}^w$. In this paper, we will only need to consider dynamic pebbling graphs $\mathsf{Dynamize}(G, N_{\mathsf{chal}})$ (see Definition 1) in which challenge $r(i) \sim [N]$ are sample uniformly at random. To implement this distribution with H, we simply define $r(i)$ as $H(\mathsf{Parents}(i, G))$ mod N.

For a graph $G = ([N], E)$, let $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$. The data-dependent memory-hard function (dMHF) $f_{\mathcal{G}}^{H}$ is defined as follows. We still have static labels X_v. The prelabels are defined as $\mathsf{prelabS}(v) = v \| 0 \| X_{\mathsf{Parents}(v,G)}$ and $X_v = H(\mathsf{prelabS}(v))$. This 0 after v helps us distinguish static labels from dynamic labels. If node v as a incoming dynamic edge $r(v)$ drawn uniformly at random from some set of nodes $S = \{s_1, \ldots, s_\ell\}$, then the dynamic prelabel for a node v is $\mathsf{prelabD}\,(v) = v \| 1 \| X_{\mathsf{Parents}(v,G)} \| X_{s_{(X_v \mod \ell)+1}}$. The dynamic prelabel of v is $T_v = \mathsf{prelabD}(v)$, and the dynamic label of v is $H(T_v)$. Intuitively, we use the randomness of the label X_v (that was just computed) to figure out which additional node we need to complete the (dynamic) label of v. If v has no dynamic incoming edges, then we let $\mathsf{prelabD}(v) \coloneqq \mathsf{prelabS}(v)$.

B The Generic Construction and Ancestral Robustness

In this section, we show that the dynamic pebbling graph $\mathsf{Dynamize}(G, 2N_{\mathsf{chal}})$ satisfies strong sustained space and cumulative complexity trade-offs as long as the underlying graph G is fractionally depth-robust and satisfies a new property called *ancestral robustness* — see definition 8. Later, we will instantiate with a DAG G that satisfies both properties to obtain a dynamic graph that we call *Extremely Grate Sample* (EGSample) — the name is a reference to the combinatorial constructions used to build G: extremely depth-robust graphs [ABP18] and Grates [Sch83]. In particular, EGSample exhibits near optimal sustained space/cumulative complexity trade-offs: any pebbling strategy sustains $\Omega(N)$ space for $\Omega(N)$ steps or has cumulative complexity $\Omega(N^{3-\varepsilon})$.

Definition 8 (Ancestral Robustness). *A graph G on N nodes is (e, C, f)-**Ancestrally Robust** if for any set S of size at most e, there exist fN nodes whose ancestors in $G - S$ have cumulative complexity at least C. That is, for any set $S \subseteq V(G)$ of size at most $|S| \leq e$, we have*

$$\left| \left\{ v \ : \ \mathsf{cc}\left(\mathsf{Ancestors}(v, G - S)\right) \geq C \right\} \right| \geq fN \ .$$

We will prove the following sustained space/cumulative complexity trade-off for $\mathsf{Dynamize}(G, 2N_{\mathsf{chal}})$.

Theorem 4 (Generic SSC/CC Trade-offs via Robustness Properties). *Fix any dynamic pebbling strategy $\mathcal{S}$ and any graph $G = ([N], E)$ that is (e, d, f)-fractionally depth-robust and (e', C, f')-ancestrally robust and let $(P_1, \ldots, P_t) = S(G')$ denote the actual pebbling of a graph $G' \sim \mathsf{Dynamize}(G, 2N_{\mathsf{chal}}))$ sampled from the dynamic edge distribution. Then for any constants $0 < p, \varepsilon < 1$ and $0 < p_{\mathsf{Unlucky}} \coloneqq ff'$, with probability at least $1 - \exp(-2\varepsilon^2 N_{\mathsf{chal}})$, over the selection of $G' \sim \mathsf{Dynamize}(G, N_{\mathsf{chal}}))$, we either have*

$$\mathsf{cc}(\mathcal{S}(G')) = \sum_{i=1}^{t} |P_i| \geq (1-p)(p_{\mathsf{Unlucky}} - \varepsilon) \min\left\{e' d N_{\mathsf{chal}}, C N_{\mathsf{chal}}\right\} \ , \quad \textit{or}$$

$$\left|\left\{i : |P_i| \geq e\right\}\right| \geq p(p_{\mathsf{Unlucky}} - \varepsilon) N_{\mathsf{chal}} \ .$$

Towards proving Theorem 4 fix a graph G on N nodes that is (e, d, f)-fractionally depth-robust and (e', C, f')-ancestrally robust. Recall that $\mathsf{Dynamize}(G, 2N_{\mathsf{chal}})$ consists of G followed by a line graph with $2N_{\mathsf{chal}}$ nodes. Let ℓ_j denote the jth node in the line graph. We will partition the $2N_{\mathsf{chal}}$ nodes into N_{chal} "challenge pairs." The ith "challenge pair" is to pebble the dynamic parents $r_1(i)$ and $r_2(i)$ of nodes ℓ_{2i+1} and ℓ_{2i+2}, respectively where $r_1(i)$ and $r_2(i)$ are sampled uniformly at random from $V(G)$. In this section we'll show that any pebbling strategy for $\mathsf{Dynamize}(G, 2N_{\mathsf{chal}})$ either sustains e pebbles for $\Omega(N_{\mathsf{chal}})$ steps or has CC at least $\Omega(\min\left\{e' d N, C\right\})$, assuming f and f' are constants.

A pebbling strategy $\mathcal{S}$ must answer N_{chal} challenge pairs. The first phase of the ith challenge pair starts on round $s_1(i)$, the step in which the static parents of $r_1(i)$ are first pebbled (equivalently, when the dynamic edge $(r_1(i), \ell_{2i+1})$ is revealed). If $\left|P_{s_1(i)}\right| \geq e$, then we can count this step towards the e-sustained space complexity. On the other hand, suppose $\left|P_{s_1(i)}\right| < e$. Then by fractional depth robustness, the depth of $r_1(i)$ in $G - P_{s_1(i)}$ is at least d with probability at least f. This means that, with probability at least f, the number of rounds $t_1(i)$ to pebble $r_1(i)$ is at least d. If the pebbling strategy $\mathcal{S}$ maintains at least e' pebbles on the graph during these $t_1(i) \geq d$ rounds, then the strategy incurs cumulatively complexity at least $e'd$ during the first phase.

The second phase starts on the round $s_2(i)$, the first step in which all static parent of ℓ_{2i+2} are pebbled (equivalently, the step in which $r_2(i)$ is revealed). The goal of this phase is punish strategies that fail to keep at least e' pebbles on the graph for the duration of phase one. Let j be a step during phase one in which $\left|P_j\right| < e'$. Since G is (e', C, f')-ancestrally robust, pebbling $r_2(i)$ incurs CC at least C with probability at least f'.

More specifically, for $G' \sim \mathsf{Dynamize}(G, 2N_{\mathsf{chal}})$, we'll say that $\mathcal{S}$ is *lucky* on challenge i if either

1. P has less than e pebbles on the graph at time $s_1(i)$ but the depth of $r_1(i)$ in $G - P_{s_1(i)}$ is less than d or

2. the depth of $r_1(i)$ in $G - P_{s_1(i)}$ is at least d, there is a step $j \in [s_1(i) : s_1(i) + t_1(i)]$ during phase one in which $\left|P_j\right| < e'$, and the cumulative complexity of the ancestors of $r_2(i)$ in $G - P_j$ is less than C.

Let Lucky_i be the indicator random variable for the event that $\mathcal{S}$ is lucky on challenge i. if $\mathsf{Lucky}_i = 0$ we will say $\mathcal{S}$ is *unlucky*. We'll show that getting lucky isn't too likely. There's one small technicality we must address, however. The random variable Lucky_i may be dependent on the outcomes of the prior events $\mathsf{Lucky}_1, \ldots, \mathsf{Lucky}_{i-1}$. Fortunately, we can still bound the conditional probability of $\mathsf{Lucky}_i = 1$ given *any* prior outcome of $\mathsf{Lucky}_1, \ldots, \mathsf{Lucky}_{i-1}$. In particular, $\Pr[\mathsf{Lucky}_i \mid \mathsf{Lucky}_1 = b_1, \ldots, \mathsf{Lucky}_{i-1} = b_{i-1}] \leq p_{\mathsf{unlucky}} \coloneqq ff'$ for any prior outcomes $b_1, \ldots, b_{i-1} \in \{0, 1\}$ — see Lemma 4. Then we can use a Hoeffding-type bound (see Lemma 39) to account for the dependence and lower bound the number of times $\mathcal{S}$ is unlucky: $\sum_i (1 - \mathsf{Lucky}_i) \geq (p_{\mathsf{unlucky}} - \epsilon) N_{\mathsf{chal}}$ with high probability.

We now proceed in bounding the probability that $\mathcal{S}$ is lucky.

Lemma 4 (Lucky Challenge Probability). *Fix any* $b_1, \ldots, b_{i-1} \in \{0, 1\}$. *Then* $[\Pr[\mathsf{Lucky}_i = 0 \mid \mathsf{Lucky}_1 = b_1, \ldots, \mathsf{Lucky}_{i-1} = b_{i-1}] \geq p_{\mathsf{unlucky}} \coloneqq ff'$.

Proof. We have that $\mathsf{Lucky}_i = 0$ if ***at least*** one of the following holds

1. **High Space:** $\left|P_{s_1(i)} \geq e\right|$.
2. **Medium Space, Long Time:** $\mathsf{depth}(r_1(i), G - P_{s_1(i)}) \geq d$ and $\left|P_j\right| \geq e'$ for all $j \in [s_1(i) : s_1(i) + t_1(i)]$.
3. **Low Space, High CC:** There is some $j \in [s_1(i) : s_1(i) + t_1(i)]$ in which $\left|P_j\right| < e'$ and $\mathsf{cc}(\mathsf{Ancestors}(r_2(i), G - \left|P_j\right|)) \geq C$.

We'll use the fact that $r_1(i)$ and $r_2(i)$ are both drawn independently and uniformly at random from G.

If $\left|P_{s(i)}\right| < e$, then by fractional depth robustness, an f-fraction of nodes in the remainder of G have depth d. Then $r_1(i)$ has depth at least d with probability at least f. Suppose this is the case. Then either the space usage of P is above e' for all of phase one (in which case we have $\mathsf{Lucky}_i = 0$ by case 2), or there is some index $j \in [s_1(i) : s_1(i) + t_1(i)]$ in which $\left|P_j\right| < e'$. If the latter holds, then by the ancestral depth-robustness of G, then the ancestors of $r_2(i)$ in $G - P_j$ is at least C with probability at least f'. So, the probability that 1), 2), or 3) hold is at least ff'. □

Now we'd like to understand how costly it is to be unlucky. Fortunately, this is straightforward due to the definition of Lucky_i. If the depth of $r_1(i)$ is at least d, then P incurs cumulative complexity $e'd$ as long as it sustains e' pebbles for all of the $t(i) \geq d$ steps. Otherwise, if there is some $j \in [s_1(i) : s_1(i) + t(i)]$ in phase one in which $\left|P_j\right| < e'$, then P incurs CC at least C.

Lemma 5 (The Cost of Being Unlucky). *If* $\mathsf{Lucky}_i = 0$*, then either* $\left|P_{s_1(i)}\right| \geq e$ *or*

$$\mathsf{cc}(P, s_1(i), s_2(i) + t_2(i)) = \sum_{s_1(i) \leq j \leq s_2(i)+t_2(i)} \left|P_j\right| \geq \min\left\{e'd, C\right\}.$$

Proof. We have that $\mathsf{Lucky}_i = 0$ if ***at least*** one of the following holds

1. **High Space:** $\left|P_{s_1(i)} \geq e\right|$.
2. **Medium Space, Long Time:** $\mathsf{depth}(r_1(i), G - P_{s_1(i)}) \geq d$ and $\left|P_j\right| \geq e'$ for all $j \in [s_1(i) : s_1(i) + t_1(i)]$.
3. **Low Space, High CC:** There is some $j \in [s_1(i) : s_1(i) + t_1(i)]$ in which $\left|P_j\right| < e'$ and $\mathsf{cc}(\mathsf{Ancestors}(r_2(i), G - P_j)) \geq C$.

We'll use the fact that $r_1(i)$ and $r_2(i)$ are both drawn independently and uniformly at random from G.

If $\left|P_{s(i)}\right| < e$, then by fractional depth robustness, an f-fraction of nodes in the remainder of G have depth d. Then $r_1(i)$ has depth at least d with probability at least f. Suppose this is the case. Then either the space usage of P is above e' for all of phase one (in which case we have $\mathsf{Lucky}_i = 0$ by case 2), or there is some index $j \in [s_1(i) : s_1(i) + t_1(i)]$ in which $\left|P_j\right| < e'$. If the latter holds, then by the ancestral depth-robustness of G, then the ancestors of $r_2(i)$ in $G - P_j$ is at least C with probability at least f'. So, the probability that 1), 2), or 3) hold is at least ff'. □

Now that we know that with constant probability, P either has a large amount of pebbles on the graph or incurs high CC, we just need to bound cost of all N_{chal} challenges.

We can now combine prove Theorem 4, the SSC/CC trade-off for EGS.

Proof of Theorem 4. Recall that $G = ([N], E)$ is (e, d, f)-fractionally depth-robust (Definition 6) and (e', C, f')-ancestrally robust (Definition 8). Let $N' = (p_{\mathsf{Unlucky}} - \varepsilon)N\mathsf{chal}$, where p_{Unlucky} is defined with respect to G as in Lemma 4. By Lemma 4 and Lemma 39, with probability at least $\exp\left(-2\varepsilon^2 N_{\mathsf{chal}}\right)$, there are distinct challenges $i_1, \ldots, i_{N'} \in [N_{\mathsf{chal}}]$ in which $\mathsf{Lucky}_{i_j} = 0$. If there are at least pN' challenges i_j such that $\left|\mathcal{S}(G')_{s(i_j)}\right| \geq e$ then we are done. Otherwise, by Lemma 5 there are at least $(1-p)N'$ challenges in which $\mathcal{S}(G')$ incurs CC at least $\min\left\{e'dN_{\mathsf{chal}}, CN_{\mathsf{chal}}\right\}$, meaning $\mathsf{cc}(\mathcal{S}(G')) \geq (1-p)N' \min\left\{e'dN_{\mathsf{chal}}, CN_{\mathsf{chal}}\right\}$. □

C Generic Trade-offs in the Parallel Random Oracle Model

Fix some dynamic pebbling graph $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$, where $G = ([N], E)$. Previously, we showed that if G satisfies strong fractional depth robustness and

ancestral robustness properties, then any dynamic pebbling strategy for $\mathcal{G}$ either has high sustained space complexity or must pay a large cumulative complexity penalty of $\omega(N^2)$. In this section, we show that any *algorithm* A evaluating $f_{\mathcal{G}}^H$ with respect to a random oracle $H : \{0,1\}^u \to \{0,1\}^w$ must either have high sustained space complexity or have cumulative memory complexity $\omega(N^2 w)$ (see Section A.3 for formal definitions of PROM algorithms and their complexity). Our proof is divided into two separate parts, which later come together to prove the trade-off. These two results are straightforward generalizations of the Scrypt analysis in [ACP+17] and the pebbling reduction in [AS15], which we include in **??** for completeness. Alone, neither of these techniques are sufficient to analyze SSC/CMC trade-offs for dMHFs. To obtain these lower bounds, show how to combine the two techniques as to take advantage of the ancestral robustness underlying $f_{\mathcal{G}}^H$.

First, we show that if G is (e, d, f)-fractionally depth robust and $\mathcal{A}$ utilizes less than $c_{\mathsf{word}} ew$ space when a challenge is issued, then with probability roughly f (over the choice of H), A must take d steps to respond to the challenge. This observation follows PROM analysis from [ACP+17], and, thus fair aligns with our dynamic pebbling analysis in Section B. Now, we would like to be able to say that if G is (a, C, f')-ancestrally robust and $\mathcal{A}$ takes d steps to respond to a challenge, then $\mathcal{A}$ either sustains $c_{\mathsf{word}} wa$ memory for d steps or, with probability $\approx f'$, incurs cumulative memory cost at least $c_{\mathsf{word}} wC$ to respond to the next challenge. This is challenging because we cannot make any direct inferences about the CMC incurred by A during those steps because the state σ_i of A does not necessarily correspond to a fixed set of pebbles; The idea of a challenge being "hard" cannot be directly defined as in Section B. Instead, we show that we can use the trace $\mathsf{trace}(A, H, X)$ of A to extract a cost-equivalent "ex post facto" (static) pebbling P of the "ex post facto" static graph $\mathcal{G}_{H,X}$, the sample from $\mathcal{G}$ that is fixed when H and X are fixed. With this result, we can show that, with high probability over the choice of H, A will eventually incur a CMC penalty of $c_{\mathsf{word}} wC$ no matter what its configuration was when it dropped below a space.

Before we formally prove these two claims relating the robustness properties of G to the cost of a PROM algorithm A evaluating $f_G H$, we setup some familiar notation and definitions. Let $\ell_1, \ldots, \ell_{N_{\mathsf{chal}}}$ denote the last N_{chal} nodes of $\mathcal{G}$, ordered topologically. These nodes are those with dynamic neighbors in $\mathcal{G}$.

Recall that we are concerned with the trace of A with respect to an oracle H on input X:

$$\mathsf{trace}(A, H, X) = [(H(\mathbf{q}^1), \sigma_1), (H(\mathbf{q}^2), \sigma_2), \ldots, (H(\mathbf{q}^T), \sigma_T)],$$

where $\sigma_i = (\tau_i, \mathbf{q}^{i+1})$ for all $i < T$. The space usage of A^H on X at time i is $|\sigma_i|$. Here, $\mathbf{q}^i$ is the parallel query made at time i; at step $i+1$, A is given σ_i and $H(\mathbf{q}^i)$.

In previous sections, for a dynamic pebbling strategy, we defined $r(i)$, $s(i)$, and $t(i)$ to denote the ith challenge node, the step in which $r(i)$ is discovered, and the number of steps it took to pebble $r(i)$ after it was discovered, respectively. In this section, we will carefully define analogous values r_i, s_i, and t_i for the

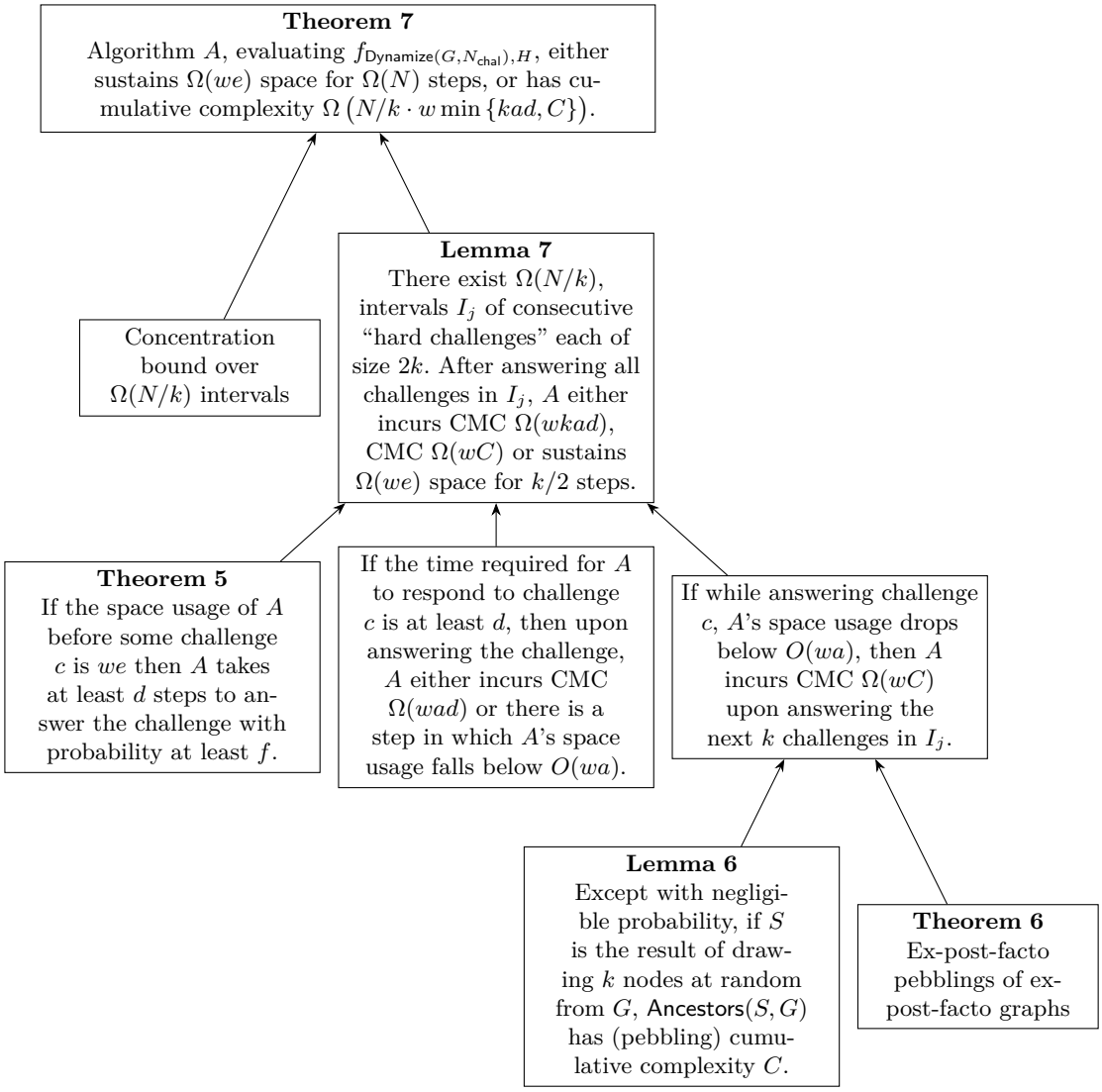

Fig. 5. This tree shows the structure of our proof of Theorem 7. In particular, we consider a PROM algorithm A evaluating the memory-hard function $f_{\mathsf{Dynamize}(G,N_{\mathsf{chal}})}$ on some input X, where $G = ([N], E)$ is an (e, d, f)-fractionally depth-robust and (a, C, f')-ancestrally robust DAG. Here, $f, f' > 0$ are constants. If a node in the tree does not have a reference, then the corresponding statement is proven directly in the proof of Theorem 4.

evaluation algorithm A^H on X. Formally, we define r_i, s_i, and, t_i to be the following implicit functions of A, H, and X:

- r_i is the node in $\mathcal{G}$ chosen uniformly using the randomness of H on $\mathsf{prelabS}(\ell_i)$. More formally, $r_i = 1 + \big(H(\mathsf{prelabS}(\ell_i)) \mod N\big)$.
- $s_i = j$ if j is the first step in which A queries the static prelabel of ℓ_i or equivalently, the step in which r_i is discovered. That is, j is the smallest integer such that $\mathsf{prelabS}(\ell_i) \in \mathbf{q}^j$ and $s_i = \infty$ if no such integer exists.
- t_i is the time it takes A to answer the challenge r_i. In our case, if $s_i, s_{i+1} < \infty$ then $t_i = s_{i+1} - s_i - 1$ for $i \in [N_{\mathsf{chal}} - 1]$ and $t_{N_{\mathsf{chal}}} = T - s_{N_{\mathsf{chal}}} - 1$. Otherwise, $t_i = \infty$.

Our proof has two main parts. First, we show that PROM algorithms that evaluate $f_{\mathcal{G}}^H$ satisfy the required time-space trade-off. In particular, if the space usage at $s(i)$ is at most e, then it takes at least d steps to compute the label for node $r(i)$.

Theorem 5. *Let A be any q-query PROM machine with input X and let $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$ for some (e, d, f)-fractionally depth-robust, constant indegree graph G on N nodes. Assume A^H on X outputs $f_{\mathcal{G}}^H(X)$ with probability χ, where the probability is taken over the choice of $H : \{0,1\}^{\delta w} \to \{0,1\}^w$. There exists a constant $c_{\mathsf{word}} \geq 0$ such that for any $\varepsilon > 0$ and $q \geq 2$, with probability at least*

$$\chi - p_{\mathsf{Dyn}}(N, N_{\mathsf{chal}}, q, w, \varepsilon) \coloneqq \chi - \frac{2N^2 + \delta q({N_{\mathsf{chal}}}^3 N + 1)}{2^w} - e^{-2\varepsilon^2 N_{\mathsf{chal}}}$$

the following hold:

1. *intervals $[s_k : s_k + t_k]$ are disjoint and finite for all $1 \leq k \leq N_{\mathsf{chal}}$, and*
2. *there exist indices $i_1, \ldots, i_{(f-\varepsilon)N}$ such that each j, either*

$$\left|\sigma_{s_{i_j}}\right| \geq e \cdot c_{\mathsf{word}} w \quad , \text{ or } \quad t_{i_j} \geq d \ .$$

Second, we show that an algorithm that drops below some space-threshold a must (with at constant probability) recompute the labels for a large portion of nodes in G, incurring high CMC. For this part, we show that we can extract a corresponding "after-the-fact" (static) pebbling of the (static) graph that is fixed when the oracle H is fixed. If A fails to compute P, then P is just a legal pebbling sequence. If A correctly computes P, then with high probability, P is in fact a legal pebbling.

Theorem 6 (Algorithms to After-the-Fact Pebblings). *Fix any input X for an algorithm A^H and let $\mathcal{G}$ be a dynamic pebbling graph and $H : \{0,1\}^* \to \{0,1\}^w$ be chosen uniformly at random. Except with probability at most $p_{\mathsf{EPF}}(N, w) = \frac{1+q+q^2}{2^w}$ over the choice of H, there exists a sequence $P = (P_0, \ldots, P_T)$, obtained from $\mathsf{trace}(A, H, X)$ in which the following hold:*

1. *P is a legal pebbling sequence for $\mathcal{G}_{H,X}$ (the graph static obtained from $\mathcal{G}$ when H is fixed) with $P_0 = \emptyset$,*
2. *For all $i \in [T]$, a node $v \in P_i$ is pebbled for the first time in P (for all $j < i$, $v \notin P_j$) if and only if the label of node v is generated for the first time at time i in A^H's computation ($\mathsf{prelabD}(v) \in \mathbf{q}^i$ but $\mathsf{prelabD}(v) \notin \mathbf{q}^j$ for all $j < i$).*
3. *For each $i \in [T]$, $c_{\mathsf{word}} w \cdot |P_i| \leq |\sigma_i|$.*
4. *If A^H succeeds in outputting $f_{\mathcal{G}}^H(X)$, then P is a legal pebbling of $\mathcal{G}_{H,X}$. In particular, P_T contains the sink of $G_{H,X}$.*

The proof of Theorem 6 is in Appendix G. An important consequence of Theorem 6 is that we can "link up" pebbling configurations from the ex post facto pebblings P from Theorem 6 and the states from $\mathsf{trace}(A, H, X)$ from Theorem 5. In particular, the times $s(i)/s_i$ in which the ith challenge is issued in the dynamic pebbling/PROM evaluation are equal. In fact we have that $s(i) = s_i$, $r(i) = r_i$ and $t(i) = t_i$ for all $i \in [N_{\mathsf{chal}}]$.

Now that we have shown that we can map the trace of the evaluation algorithm A to a cost-equivalent static pebbling of the ex-post-facto graph, we want to take advantage of the ancestral robustness of G to show that if A is sufficiently low space for some consecutive challenges $i_1, \ldots, i_k$, then A must incur CMC $\Omega(wC)$. The idea is that for sufficiently high choices of k, except with negligible probability, the ancestors of $i_1, \ldots, i_k$ in $G - S$ *for all* sets of size S.

Lemma 6. *Fix any (e, C, f)-ancestrally robust graph $G = (V = [N], E)$ on N nodes and sample k nodes $i_1, \ldots i_k$ uniformly at random from $[N]$ and let $\mathtt{LOWCC}$ denote the event that there exists a subset $S \subseteq [N]$ of size $|S| \leq e$ such that $\mathsf{cc}\left(\mathsf{Ancestors}\left(I, G - S\right)\right) < C$ where $I = \{i_1, \ldots i_k\}$. The probability of the event $\mathtt{LOWCC}$ is at most $\Pr[\mathtt{LOWCC}] \leq p_{\mathsf{lowCC}}(N, e, f, k) \coloneqq \exp\left(e \log N - fk\right)$.*

Proof. Fix a set $S \subseteq [N]$ of size e and for each $j \leq k$ let $\mathtt{B}_{S,j}$ denote the event that $\mathsf{cc}\left(\mathsf{Ancestors}\left(\{i_j\}, G - S\right)\right) < C$. Let $\mathtt{B}_S \coloneqq \bigcap_{j=1}^{k} \mathtt{B}_{S,j}$ and note that if $\mathtt{B}_S$ does not occur then there exists some $j \leq k$ for which $\mathtt{B}_{S,j}$ does not occur and it immediately follows that $\mathsf{cc}\left(\mathsf{Ancestors}\left(I, G - S\right)\right) \geq \mathsf{cc}\left(\mathsf{Ancestors}\left(\{i_j\}, G - S\right)\right) \geq C$. Thus, it suffices to upper bound the probability of the event $\mathbb{B}_S$ and union bound over all subsets $|S| \leq e$

Because G is (e, C, f)-ancestrally robust it follows that for each $j \leq k$ we have $\Pr[\mathtt{B}_{S,j}] = Pr_{i_j \sim [N]}[\mathsf{cc}\left(\mathsf{Ancestors}\left(\{i_j\}, G - S\right)\right) < C] \leq 1 - f$. The events $\mathtt{B}_1, \ldots \mathtt{B}_k$ are independent since we pick $i_1, \ldots, i_k$ independently. Thus, the probability of the event $\mathtt{B}_S \coloneqq \bigcap_{j=1}^{k} \mathtt{B}_{S,j}$ is at most

$$\Pr[\mathtt{B}_S] = \prod_{j=1}^{k} \Pr[\mathtt{B}_{S,j}] \leq (1 - f)^k \leq \exp\left(-fk\right)$$

Observe that if $S' \subseteq S$ and the event $\mathtt{B}_S$ occurs, that the event $\mathtt{B}_{S'}$ will also occur. Thus, it is sufficient to union bound over all $\binom{N}{e} \leq N^e = \exp\left(e \log N\right)$ subsets $S \subseteq [N]$ with size exactly $|S| = e$. We have $\Pr[\exists S \ : \mathtt{B}_S \wedge |S| = e] \leq \exp\left(e \log N - fk\right)$. □

Lemma 6 is useful because it allows us to bound the cumulative complexity of pebbling the challenges issued by k of the N_{chal} challenge nodes in $\mathcal{G}$. It shows that for any fixed set of k challenges, if the pebbling P extracted from $\mathsf{trace}(A, H, X)$ has less than e pebbles before these challenges are issued, then, with high probability, P will incur CC at least C before answering each challenge. This implies that A^H also incurs this high CC penalty. Still, this is not enough for our applications. We need Lemma 6 to hold for every possible interval of length k, so that the next k challenges are "hard" no matter when the adversary drops below the predetermined space threshold.

Lemma 7. *Fix any (e, C, f)-ancestrally robust graph $G = (V = [N], E)$ on N nodes, $k > 0$, and let $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$. Consider $\mathcal{G}_{G,X}$ for some uniformly chosen oracle H and fixed string X. With probability at most*

$$(N_{\mathsf{chal}} - k)p_{\mathsf{lowCC}}(N, e, f, k) \leq \exp\left(e \log N + \log N_{\mathsf{chal}} - fk\right)$$

over the choice of H, for every $j \in [N_{\mathsf{chal}} - k]$ and set $S \subseteq [N]$ of size at most e, the interval $I_j = [j : j + k]$ satisfies $\mathsf{cc}(\mathsf{Ancestors}(I_j, G - S)) \geq C$.

Proof. Define the event LOWCC as in Lemma 6. We need to bound the probability LOWCC occurs not just for a single set of k nodes, but rather for all k-length contiguous intervals of the N_{chal} challenge nodes in the line graph of $\mathcal{G}$. By the union bound, the probability that every I_j satisfies LOWCC is at most $(N_{\mathsf{chal}} - k)\Pr[\mathtt{LOWCC}] \leq (N_{\mathsf{chal}} - k)p_{\mathsf{lowCC}}(N, e, f, k) \leq \exp\left(e \log N + \log N_{\mathsf{chal}} - fk\right)$ by Lemma 6. □

Putting these two theorems together, we obtain the following security bounds for dMHFs in the parallel random oracle model.

Theorem 7 (SSC/CMC Trade-offs in the Parallel Random Oracle Model). *Define p_{Dyn} from Theorem 5, p_{EPF} from Theorem 6, p_{lowCC} from Lemma 6, let G be an (e, d, f)-fractionally depth-robust and (a, C, f')-ancestrally robust graph on N nodes. Now define any constant $0 < \varepsilon < 1$ and integer $N_{\mathsf{chal}} := N_{\mathsf{chal}}(N) \geq 2$. Lastly, fix a q-query PROM algorithm A^H with oracle access to H that succeeds in computing $f_{\mathcal{G}}^H$ on some input X with probability χ over the choice of H with trace $\mathsf{trace}(A, H, X) = (\sigma_1, \ldots, \sigma_T)$.*

Then with probability at least

$$\chi - p_{\mathsf{ROM}} = \chi - O\left(\frac{q{N_{\mathsf{chal}}}^2 N}{2^w} - N_{\mathsf{chal}} \exp(-a \log N)\right)$$

(over the choice of H), both of the following hold:

1. *A^H correctly computes $f_{\mathcal{G}}^H$ on X.*
2. *A^H either sustains $c_{\mathsf{word}} w \cdot e$ space for $\frac{f-\varepsilon}{2} N_{\mathsf{chal}}$ steps:*

$$c_{\mathsf{word}} w \cdot e\text{-}\mathsf{ssc}(A, H, X) = \left|\left\{|\sigma_i| : \ i \in [T]\right\}\right| \geq \frac{f - \varepsilon}{2} N_{\mathsf{chal}}$$

or

$$\mathsf{cmc}(A,H,X) = \sum_{i\in[T]} |\sigma_i| \geq c_{\mathsf{word}} w \cdot (f-\varepsilon) N_{\mathsf{chal}} \min\left\{ad/2, \frac{f'C}{4a\log N}\right\}$$

where

$$\begin{aligned} p_{\mathsf{ROM}} &\coloneqq p_{\mathsf{ROM}}(N, N_{\mathsf{chal}}, q, e, f', a, w, \varepsilon) \\ &\coloneqq p_{\mathsf{Dyn}}(N, N_{\mathsf{chal}}, q, w, \varepsilon) - p_{\mathsf{EPF}}(N, w) - N_{\mathsf{chal}} \cdot p_{\mathsf{lowCC}}(N, e, f', 2a\log N) \end{aligned}$$

Proof. Let $(\sigma_1, \ldots, \sigma_T) = \mathsf{trace}(A, H, X)$, where H is chosen uniformly at random. We define our "interval length" to be $k = \frac{2a\log N}{f'}$. Let $G' = \mathcal{G}_H$ and $\ell_1, \ldots, \ell_{N_{\mathsf{chal}}}$ denote the last N_{chal} nodes of G' consisting of the underlying line graph. For $N' = (f-\varepsilon)N_{\mathsf{chal}}$ Theorem 5 implies that with probability p_{Dyn}, at least $\chi - p_{\mathsf{Dyn}}(G, N_{\mathsf{chal}}, w)$, A^H successfully computes $f_{\mathcal{G}}^H(X)$, and there exist nodes $\mathsf{costly} = \left\{\ell_{i_1}, \ldots, \ell_{i_{N'}}\right\}$ (sorted topologically with respect to $j \in [N']$) such that for all $v \in \mathsf{costly}$, the memory usage of A^H at the start of challenge v is large ($|\sigma_{s_v}| \geq e \cdot c_{\mathsf{word}} w$) or A^H takes a long time to answer the challenge ($t_v \geq d$). Furthermore by Theorem 6, except with probability at most p_{EPF}, there exists a pebbling $P = [P_1, \ldots, P_T]$ such that $v \in P_i$ for the first time if and only if $\mathsf{prelabD}(v)$ is first seen as an input to H on round i, and $|\sigma_i| \geq c_{\mathsf{word}} w |P_i|$ for all $i \in [T]$.

Suppose there are less than $N'/2$ nodes $v \in \mathsf{costly}$ for which $\left|\sigma_{s_{\ell_v}}\right| \geq c_{\mathsf{word}} we$ (otherwise we are done because the sustained space complexity is sufficiently high). Let $\mathsf{costly}' \subseteq \mathsf{costly}$ be the set of nodes $v = \ell_{i_j}$ in which the space usage at time s_v is less than $c_{\mathsf{word}} we$. This means that $\left|P_{s(v)}\right| < e$ as well. Let $N'_k \coloneqq \frac{N'}{2k}$. Define intervals $I_1, \ldots, I_{N'_k}$, where $I_j = \{\ell_{i_m} : (j-1)k+1 \leq m \leq jk\}$. We further split each interval I_j into two equally sized intervals I_j^1 and I_j^2. The interval I_j^1 has the first half (topologically) of the challenges and I_j^2 has the second half. We let $\mathsf{first}(I_j^b)$ denote the step s_v, where v first node of I_j^b. Similarly, we $\mathsf{last}(I_j^b)$ denote the step before the last node of I_j^b is pebbled.

We now have two cases for each $j \in [N'_k]$. Either $|\sigma_m| \geq c_{\mathsf{word}} wa$ for all $\mathsf{first}(I_j^1) \leq m \leq \mathsf{last}(I_j^2)$, or there is a step m in which $|\sigma_m| < c_{\mathsf{word}} wa$. In the first case, we know that

$$\begin{aligned} \mathsf{cmc}(A, H, X, \mathsf{first}(I_j^1), \mathsf{last}(I_j^1)) &= \sum_{\mathsf{first}(I_j^1) \leq m \leq \mathsf{last}(I_j^1)} |\sigma_m| \\ &\geq c_{\mathsf{word}} w \cdot adk. \end{aligned}$$

Otherwise, if $|\sigma_m| < c_{\mathsf{word}} wa$, then

$$\begin{aligned} \mathsf{cmc}(A, H, X, \mathsf{first}(I_j^1), \mathsf{last}(I_j^2)) &\geq \mathsf{cmc}(A, H, X, \mathsf{first}(I_j^1), \mathsf{last}(I_j^1)) \\ &\geq \min_{|S|<a} c_{\mathsf{word}} w \cdot \mathsf{cc}(\mathsf{Ancestors}(I_j, G' - S), G'). \end{aligned}$$

By Lemma 7, $\mathsf{cc}(\mathsf{Ancestors}(I_j, G' - S, G')) \geq C$ for all j with probability at least $1 - N_{\mathsf{chal}} p_{\mathsf{lowCC}}(N, a, f', k)$ over the choice of H. Then

$$\mathsf{cmc}(A, H, X, \mathsf{first}(I_j^1), \mathsf{last}(I_j^2)) \geq \min\{adk, C\}.$$

Summing over all N'_k intervals, if the above events occur, then the CMC of A^H on X is at least

$$\begin{aligned}
\mathsf{cmc}(A, H, X) &\geq c_{\mathsf{word}} w \cdot N'_k \min\{adk, C\} \\
&= c_{\mathsf{word}} w \cdot \frac{(f-\varepsilon)N_{\mathsf{chal}}}{2k} \min\{adk, C\} \\
&= c_{\mathsf{word}} w \cdot (f-\varepsilon)N_{\mathsf{chal}}/2 \cdot \min\{ad, C/k\} \\
&= c_{\mathsf{word}} w \cdot (f-\varepsilon)N_{\mathsf{chal}} \min\left\{ad/2, \frac{f'C}{4a \log N}\right\}
\end{aligned}$$

□

D How to Construct Ancestrally Robust Graphs

In Section B, we showed that ancestrally robust graphs can be used to construct dynamic pebbling graphs with high sustained space and cumulative complexity trade-offs. In this section, we will give sufficient conditions for a graph to be ancestrally robust. In particular, to construct an ancestrally robust graph, it suffices to find a graph that satisfies a property called *local expansion*. Intuitively, a node v in a graph G is a δ-local expander if there are δk edges between the k proceeding nodes of (and including) v and the k succeeding nodes of v for every $k > 0$ in which the intervals both exist.

Definition 9 (Local Expansion [ABP18]). *Let $I_v(k) = [k - v - 1 : v] \cap [N]$ and $I_v^\star(k) = [v + 1 : v + k] \cap [N]$. For any $\delta > 0$, a node v of $G = ([N], E)$ is a **δ-local expander** if for any $1 \leq k \leq \min v, N - v$, $A \subseteq I_v(k)$, and $B \subseteq I_v^\star(k)$ each of size at least $|A|, |B| \geq \delta k$, we have*

$$A \times B \cap E \neq \emptyset.$$

We say that G is a δ-local expander if every node in G is a δ-local expander.

In some sense, local expansion is too strong of a requirement. For practical memory-hard functions, we need our graphs to have constant in-degree [BH22]. However, if a graph $G = ([N], E)$ is a δ-local expander, then G has CC $\Omega(N^2)$ [EGS75]. Every constant indegree graph has CC at most $O\left(\frac{N \log\log N}{\log N}\right)$, so there are no constant indegree δ-local expanders, which is necessary for the construction of practical MHFs. Fortunately, we show that it suffices for a graph related to G, called its *metagraph*, to be a δ-local expander instead. This is a much easier

requirement, and we'll show in Section E that practical graphs already satisfy this property.

The metagraph $G_m = ([N'], E_m)$ of a graph $G = ([N], E)$, where $N' = \lfloor N/m \rfloor$ is defined as follows. Each node $v \in [N']$ of G corresponds to the interval $I_v^\star(m) \coloneqq [(v-1)m + 1 : vm]$. Thus, we call these nodes *metanodes*. The first $m/5$ nodes of $I_v^\star(m)$ is the *first part* of v and the last $m/5$ nodes are called the *last part* of v. The edge set E_m consists of edges (u, v), where there is an edge in G from some node in the last part of u to a node in the first part of v.

Definition 10 (Metagraph [ABH17]). *Let $G = ([N], E)$ be a graph on N nodes, $m \in [N]$ and $N' = \lfloor N/m \rfloor$. The metagraph $G_m = ([N'], E_m)$ of G is defined as follows. The ith metanode of G_m represents the nodes $M_i = [(i-1)m+1 : im]$ in G. We denote the "first part" of the meta-node i, with respect to an implicit parameter, as $M_i^F = \left[(i-1)m + 1 : \lceil (i - 1 + 1/5)m \rceil\right]$. The last part of i is $M_i^L = \left[\lfloor (i + 1 - 1/5)m + 1 \rfloor : im\right]$. The edgeset E_m for G_m is defined as follows. We have for all $i < j \in [N']$, $(i, j) \in E_m$ if $E \cap M_i^{\mathsf{last}} \times M_j^{\mathsf{first}} \neq \emptyset$ and graphs $M_i \cap G$ and $M_j \cap G$ both contain the line graphs on m nodes.* [8]

We will prove the following relationship between the ancestral robustness of a graph, and the local expansion of its metagraph.

Theorem 8 (Meta Local Expansion to Ancestral Robustness). *For a graph G on N nodes and $m \geq 0$, if G_m is a δ-local expander, then G is*

$$\left(pN/m, 3{\delta_{\mathsf{DRA}}}^2 \frac{N^2}{5m}, \frac{4f}{5}\right) \textit{-ancestrally robust}$$

for all $0 < p, f < 1$ satisfying $0 < 2p + f \leq 1 - 10\delta$, with $\delta_{\mathsf{DRA}} \coloneqq \delta_{\mathsf{DRA}}(f, p, \delta) = (1 - 10\delta - 2p - f)/2$

To prove Theorem 8, we first show that δ-local expanders are ancestrally robust. In fact, we show the stronger property, that the corresponding ancestors are depth-robust (see Lemma 10). Theorem 8 is then implied by Lemma 12, which states that if a meta graph G_m satisfies this stronger ancestral property, then the original graph G is ancestrally robust.

D.1 Depth-Robustness of Ancestors of Local Expander Nodes

Recall that δ-local expanders $G([N], E)$ are $(\Omega(N), \Omega(N))$-depth robust. We will show an even stronger property of δ-local expanders. If we remove pN nodes, there are still fN remaining nodes whose *ancestors* are $(\Omega(N), \Omega(N))$-depth robust. This is useful because, as we will later see, this property is transferable from metagraphs to their original graphs.

[8] [ABH17] also defined the notion of metagraphs while implicitly assuming that G was a supergraph of the line graph on N nodes. For our analysis, we will need to define metagraphs for graphs without this property (See Lemma 11).

These δ-local expanders satisfy strong depth-robustness properties for all reasonable choices of the e parameter. This property is called ε-extreme depth-robustness.

Definition 11 (Extreme Depth-Robustness [ABP18]). *A graph $G = ([N], E)$ is ε-extremely depth-robust for $\varepsilon > 0$ if for any $e + d \leq N(1-\varepsilon)$, G is (e, d)-depth robust.*

As Lemma 8 asserts, local expansion implies extreme depth-robustness.

Lemma 8 (Local Expansion to Extreme Depth-Robustness [ABP18]). *Any δ-local expander is 10δ-extremely depth-robust.*

While Lemma 8 is not directly stated in [ABP18] it is implicit in their proof of Theorem 3. One useful property of ε-extreme depth-robust graphs is that any moderately large subgraph H of G (e.g., containing more than εN nodes) will itself be highly depth-robust (e.g., (e, d) depth-robust with $e = d = (|H| - \epsilon N)/2$) and consequently have high CC (e.g., $\mathsf{cc}(H) \geq (|H| - \epsilon N)^2/4$). To supplement this fact, we also will also show that if you remove e nodes from a local expander, there are at least $N - 2e$ leftover nodes that are all connected to each other. Lemma 9 follows immediately from results of [ABP18]. We include the proof in Appendix H for the sake of completeness.

Lemma 9 ([ABP18]). *If a graph $G = ([N], E)$ is a δ-local expander with $\delta < 1/6$, then for any set S, the graph $G - S$ contains a a set of nodes $C \subseteq [N] - S$ of size at least $N - 2|S|$, with a path from u to v in $G \cap C$ for all $u < v \in C$.*

Using Lemma 9 and Lemma 8, we can show that, even upon removing many nodes from the graph, there are still many nodes whose ancestors are highly depth-robust.

Lemma 10 (Local Expansion to Depth-Robust Ancestors). *Let G be a δ-local expander on N' nodes with $\delta < 1/10$. Then for any set S of size at most pN', there are fN' nodes v in $G - S$ such that $\mathsf{Ancestors}(v, G - S)$ is $(\delta_{\mathsf{DRA}} N', \delta_{\mathsf{DRA}} N')$-depth robust for any constants $0 < f, p < 1$ satisfying $0 < 2p + f \leq 1 - 10\delta$ and $\delta_{\mathsf{DRA}} = \delta_{\mathsf{DRA}}(p, f, \delta) \coloneqq (1 - 10\delta - 2p - f)/2$.*

Proof. Fix any subset $S \subseteq [N']$ of size at most pN'. By Lemma 9, $G - S$ contains a directed path of length $(1 - 2p)N'$. That means that the last fN' nodes of this directed path have at least $(1 - 2p - f)N'$ ancestors. Let L' denote the set of these final fN' nodes. If $v \in L'$ then $\mathsf{Ancestors}(v, G - S)$ has at least $(1 - 2p - f)N'$ nodes. We claim that any sufficiently large subgraph of G must be depth-robust. By Lemma 8, we know that G is 10δ-extremely depth-robust, meaning that for any constant $\alpha > 0$ the graph G is $\big((\alpha + 2p + f)N', \beta N'\big)$-depth-robust with $\beta = (1 - 10\delta - 2p - f - \alpha)$. Thus, $\mathsf{Ancestors}(v, G - S)$ is $(\alpha N', \beta N')$-depth robust. The quantity $\alpha\beta$ is maximized when $\alpha = (1 - 10\delta - 2p - f)/2$ so that $\alpha = \beta$. □

D.2 Metagraph Local Expansion Implies Ancestral Robustness

In this section we show that the if a metagraph G_m satisfies ancestor depth-robustness properties as stated in Lemma 10, then G is ancestrally robust. From the above results, this shows that if G_m is a local expander, then G is ancestrally robust.

For these proofs, we define a new operator "$\ominus$" between graphs G and sets S as $G \ominus S$. The difference between "$-$" and "$\ominus$" is only that "$-$" removes the nodes in S while "$\ominus$" simply removes all incoming and outgoing edges of S. This operation will be convenient in connecting properties of $G - S$ with its metagraph G_m, because it preserves the m-length intervals in G corresponding to each metanode. We first prove some basic facts about the relationship between G and G_m. Lastly, we define $\mathsf{toMeta}_m(v)$ denote the meta node corresponding to v ($v \in [(v-1)m + 1 : vm]$.

Lemma 11 (Meta Properties). *Let $G = ([N], E)$ be a graph and $G_m = ([N'], E)$ be its metagraph.*

1. *For any set $S \subseteq [N]$, $(G \ominus S)_m = G_m \ominus \mathsf{toMeta}_m(S)$.*
2. *For any path from node u to node v in G_m of length d, there is a path of length $3md/5$ from the first node in the interval M_u to every node in $M_v^{\mathsf{mid}} \cup M_v^{\mathsf{last}}$.*
3. *If $\mathsf{Ancestors}(v, G_m)$ is (e, d)-depth robust, then $\mathsf{Ancestors}(u, G)$ is $(e, 3md/5)$-depth robust for every $u \in M_v^{\mathsf{last}}$.*

Proof.

1. From Definition 10, we know that if an interval M_v does not contain a line graph on m nodes, then $\mathsf{toMeta}_m(v)$ has no incoming or outgoing edges in G_m. Thus, for every node v in S, $\mathsf{toMeta}_m(v)$ will have no incoming or outgoing edges in $(G \ominus S)_m$. Furthermore, if there is an edge (u, v) in G_m with $u, v \notin \mathsf{toMeta}_m(S)$, then M_v and M_u form line graphs in $G \ominus S$ and there is an edge from some node in M_u^{last} to some node in M_v^{first}. Thus, (u, v) is an edge in $(G \ominus S)_m$, meaning that $(G \ominus S)_m = G_m - \mathsf{toMeta}_m(S)$.
2. Let $u = u_0 \to u_1 \to \ldots \to u_d = v$ be a path of length d in G_m. Then from Definition 10, we see that M_{u_i} forms a line graph in G for all $0 \leq i \leq d$ and for every $0 \leq i < d$, there exists a nodes $\mathsf{out}_i \in M_{u_i}^{\mathsf{last}}$ and $\mathsf{in}_{i+1} \in M_{u_{i+1}}^{\mathsf{first}}$ such that $(\mathsf{out}_i, \mathsf{in}_{i+1})$ is an edge in G. Since in_{i+1} and out_{i+1} correspond to the same metanode in G_m, there is also a path from in_{i+1} to out_{i+1}. In G, any path from in_i to out_i has length at least $3m/5$, as the path must go through every node in $M_{u_{i+1}}^{\mathsf{mid}}$. Then there is a path that intersects every node $\mathsf{out}_0, \mathsf{in}_1, \mathsf{out}_1, \mathsf{in}_2, \ldots, \mathsf{in}_d$ of length at least $\sum_{1 \leq i < d} \left|M_{u_i}^{\mathsf{mid}}\right| = 3(d-1)m/5$. Additionally, there's a path from the first node of M_u to out_0, which contributes additional length of least $4m/5$. Thus, there is a path of length $3md/5$ from the first node in the interval M_u to w.
3. Fix a metanode $v \in [N']$ in which $\mathsf{Ancestors}(v, G_m)$ is (e, d)-depth robust, a node $w \in M_v^{\mathsf{mid}} \cup M_v^{\mathsf{last}}$, a set $S \subseteq \mathsf{Ancestors}(w, G)$ of size at most e, and let $S' = \mathsf{toMeta}_m(N)$. To prove the depth-robustness of the ancestors of w, we'll

use the previous two properties of G_m to prove that there is always a length $3md/5$ path in $\mathsf{Ancestors}(w, G) - S$. By the given (e, d)-depth robustness of $\mathsf{Ancestors}(v, G_m)$, there's a u-v path of length at least d in $G_m - S'$ and thus in $G_m \ominus S'$. By Item 1, $G_m \ominus S'$ is the m-metagraph for $G \ominus S$. Then by Item 2, there is a path of length at least $3md/5$ from the first node of M_u^{first} to w. Thus, $\mathsf{Ancestors}(w, G - S)$ is $(e, 3md/5)$-depth robust.

□

Now we show that if a metagraph G_m of a graph $G = ([N], E)$ has this ancestor depth-robustness property, then G is ancestrally depth-robust.

Lemma 12 (Meta Depth-Robust Ancestors to Ancestral Robustness). *Let G be a graph on N nodes. For $m < N$, let $N' = \lfloor N'/m \rfloor$. If every set A of size at most a, the graph $G_m - A$ has at least fN' nodes v whose ancestors in $G_m - A$ are (e, d)-depth robust, then G is $(a, 3med/5, 4f/5)$-ancestrally robust.*

Proof. Let $A' = \mathsf{toMeta}_m(A)$, and fix $0 < f < 1$. By Item 1 of Lemma 11, $(G \ominus A)_m = G_m \ominus A'$. Since $\left|A'\right| \leq |A|$, there are at least fN' metanodes whose ancestors are (e, d)-depth robust in $G_m - A'$. By Item 3 in Lemma 11, it follows that there are $4fN'm/5 = 4fN/5$ nodes in $G \ominus A$ (and thus $G - A$) whose ancestors are $(e, 3md/5)$-depth robust, meaning G is $(e, 3emd/5, 4f/5)$-ancestrally robust. □

Now we can prove Theorem 8, concluding that if G_m is a local expander, then G is ancestrally robust.

Proof of Theorem 8. Let $N' = \lfloor N/m \rfloor$. Since G_m is a δ-local expander, it follows from Lemma 10 that for constants $p, f > 0$ satisfying $2p + f \leq 1 - 10\delta$ and any set $S \subseteq [N]$, of size at most pN', $G_m - \mathsf{toMeta}_m(S)$ has at least fN' nodes $v_1, \ldots, v_{fN'}$ in G_m whose ancestors $\mathsf{Ancestors}(v_j, G_m - \mathsf{toMeta}_m(S))$ are $(\delta_{\mathsf{DRA}} N', \delta_{\mathsf{DRA}} N')$-depth robust, where $\delta_{\mathsf{DRA}} = \delta_{\mathsf{DRA}}(p, f, \delta)$ defined in Lemma 10. Thus, by Lemma 12, G is $\left(\frac{pN', 3\delta_{\mathsf{DRA}}{}^2 N^2}{5m}, 4f/5\right)$-ancestrally robust. □

E EGSample: Towards Practical Sustained Space and Cumulative Complexity Trade-offs

From Theorem 4, we know that if we can find constant indegree, fractionally depth robust, and ancestrally robust graphs, then we can construct a dynamic pebbling graph (and thus dMHF) with high sustained space and cumulative complexity trade-offs. Then Theorem 8 shows that it suffices to find a graph whose metagraph is a local expander. In this section we will show that graphs output by the randomized algorithm called *DRSample* [ABH17] already satisfies this property with high probability. Alwen et al. [ABH17] introduced $\mathsf{DRSample}(N)$ and proved that with high probability a randomly sampled graph $\mathsf{DRS} \sim \mathsf{DRSample}(N)$ is $\left(\Omega\left(N/\log N\right), \Omega(N)\right)$-depth-robust. We

will show that DRS is (asymptotically) $(N/m, N^2/m, f)$-ancestrally robust for all $m = \Omega(\log^2 m)$. To do this we just need the following bounds on the edge distribution of DRSample.

It is unknown whether DRSample satisfies the required fractional depth-robustness; however, we show that another graph family called *Grates* [Sch83] does. It turns out that we can combine these graphs to retain both properties without increasing the indegree. We take the union of Grates and DRSample G and reduce the resulting graph's indegree using somewhat standard techniques to obtain $\mathsf{reduce}(G)$, which is an indegree two, ancestrally robust and fractionally depth-robust graph with high probability.

Definition 12 (Extremely Grate Sample). *Define the randomized algorithm* $\mathsf{DRSample}$ *from Definition 14, the graph family* Grates *from Theorem 11, and* reduce *from Definition 15. For* $N \geq 2$ *and* $0 < \varepsilon < 1$*, the randomized algorithm* $\mathsf{EGSample}(N, \varepsilon)$ *is the result of sampling* $\mathsf{DRS} \leftarrow \mathsf{DRSample}(N)$ *and returning* $\mathsf{reduce}(\mathsf{DRS} \cup \mathsf{Grates}(N, \varepsilon))$*, an indegree two graph on* $2N$ *nodes.*

Theorem 9 (The Robustness of EGSample). *For any constants* $0 < p, f, c < 1$ *satisfying* $\varepsilon' \coloneqq 1 - 2p - 5f/12 - \sqrt{20c/9} \geq 0$*. There exists constants* $\gamma_1, \gamma_2 \geq 0$ *such that for all integers* $N \geq 2$ *and* $m \coloneqq m(N)$*,* $G \leftarrow \mathsf{EGSample}(N, \varepsilon)$ *is*

$$(pN/m, cN^2/m, f)\text{-}ancestrally\ robust$$

and

$$(\gamma_1 N, \gamma_2 N^{1-\varepsilon}, \gamma_1)\text{-}fractionally\ depth\text{-}robust,$$

except with probability at most $p_{\mathsf{DAR}}(N, m, p, c, f) \leq 21N^2 e^{-\frac{\varepsilon' m}{2500 \log N}}$*, where* p_{DAR} *is defined explicitly in Theorem 10.*

Furthermore, G *has* $3N$ *nodes and* $\mathsf{indegree}(G) = 2$

Definition 13 (Dynamic Extremely Grate Sample). ***Dynamic Extremely Grate Sample (DEGSample)*** *is the randomized algorithm which samples from the distribution* $\mathsf{DEGSample}(N, N_{\mathsf{chal}}, \varepsilon)$*, defined by first sampling* $\mathsf{EGS} \leftarrow \mathsf{EGSample}(N, \varepsilon)$ *and then outputting* $\mathsf{Dynamize}(\mathsf{EGS}, 2N_{\mathsf{chal}})$*.*

Now we know that, with high probability, dynamic pebbling graphs output by DEGSample satisfy our required robustness properties (Theorem 9), it follows from our generic pebbling trade-off (Theorem 4) that any pebbling strategy either sustains $\Omega(N)$ pebbles for $\Omega(N_{\mathsf{chal}})$ steps, or has cumulative complexity at least $\Omega(N^{3-\varepsilon})$.

Corollary 1 (DEGSample Pebbling SSC/CC Trade-off). *Fix any dynamic pebbling strategy* $\mathcal{S}$ *along with constants* $< \gamma_1, \gamma_2, p, c, f, \varepsilon > 0$*, where* γ_1 *and* γ_2 *depend only on* ε*. Then for all* N *sufficiently large and* $N_{\mathsf{chal}} = \frac{2N}{f\gamma_1 - \varepsilon/2}$*, with probability at least* $1 - p_{\mathsf{DAR}}(N, m, p, c, f)$ *over the choice of* $\mathsf{DEGS} \sim \mathsf{DEGSample}(N, N_{\mathsf{chal}}, \varepsilon/2)$*, either*

$$\mathsf{cc}(\mathcal{S}(\mathsf{DEGS})) \geq \gamma_2 f N^{3-\varepsilon}$$

or

$$\gamma_1 N\text{-}\mathsf{ssc}(\mathcal{S}(\mathsf{DEGS})) \geq N.$$

The function p_{DAR} *is from Theorem 10 and is negligible in* m *when* $m = \Omega(\log^2 N)$.

Again, from the robustness of EGSample Theorem 9, we may apply Theorem 7. To avoid dealing with the extra $1/\log N$ factor as seen in Theorem 7, we simply pick the parameter ε in $\mathsf{DEGSample}(N, N_{\mathsf{chal}}, \varepsilon)$ to be slightly smaller than needed, absorbing these subpolynomial factors.

Corollary 2 (DEGSample SSC/CC Trade-off in the Parallel Random Oracle Model). *Define* DEGSample *from Definition 13. For some constants* $\alpha_1, \alpha_2, \alpha_3, \varphi_1, \varphi_2, \varphi_3, \varepsilon_1, \varepsilon_2 > 0$ *such that for all sufficiently large* N, $N\mathsf{chal} = \frac{2N}{(\varphi_1 - \varepsilon_2)}$, *and* $m = N^{1/2+\varepsilon_1/2}$, *the following hold. Fix any* q*-query PROM algorithm* A^H *with trace* $\mathsf{trace}(A, H, X) = (\sigma_1, \ldots, \sigma_T)$ *on input* X *that succeeds in computing* f^H_{DEGS}, *where* $\mathsf{DEGS} \leftarrow \mathsf{DEGSample}(N, N_{\mathsf{chal}}, 0.99\varepsilon_1/2)$. *Then, except with probability*

$$p_{\mathsf{DEGS}}(N, \alpha, \varphi, \varepsilon) = O\left(\frac{qN^4 + q^2}{2^w} + \exp(-\sqrt{N})\right).$$

both of the following hold:

1. A^H *correctly computes* f^H_{DEGS} *on* X.
2. A^H *either sustains* $c_{\mathsf{word}}\varphi_1 w \cdot N$ *space for* $(\varphi_3 - \varepsilon)N_{\mathsf{chal}}$ *steps:*

$$c_{\mathsf{word}}\varphi_1 w \cdot N\text{-}\mathsf{ssc}(A, H, X) = \left|\left\{|\sigma_i| : \ i \in [T]\right\}\right| \geq N$$

or

$$\mathsf{cmc}(A, H, X) = \sum_{i \in [T]} |\sigma_i| \geq c_{\mathsf{word}} w \cdot N^{2.5-\varepsilon_1}.$$

E.1 DRSample is Ancestrally Robust

For the rest of this section, we mainly focus on the analysis of DRSample before combining DRSample and Grates. We will analyze DRSample via its edge distribution.

Definition 14 (DRSample [ABH17]). *For an integer* $N > 0$, $\mathsf{DRSample}(N)$ *is a randomized algorithm which outputs an indegree two (static) graph* $G = \big([N], E\big)$ *such that*

$$E = \big\{(i, i+1) : i \in [N-1]\big\} \cup \big\{(r(i), i) : i \in [2:N]\big\},$$

where for $u < v - 1$ *we have*

$$\Pr[r(v) = u] \geq \frac{1}{(v-u)\log v}.$$

As in Theorem 8, to show that a graph $\mathsf{DRS} \sim \mathsf{DRSample}(N)$ sampled from DRSample is ancestrally robust, it suffices to show that a metagraph DRS_m of DRS is a δ-local expander with high probability. We show that for any constant $\delta > 0$ and $m \coloneqq m(N) = \Omega(\log^2 N)$, the metagraph DRS_m is a δ-local expander except with negligible probability in N. The proof is similar to that of [ABH17], so it is left to Section H.

Lemma 13 (Meta-DRSample Local Expansion). *Let $G \leftarrow \mathsf{DRSample}(N)$. For any constant $\delta > 0$, G_m is a δ-local expander except with probability at most*

$$p_{\mathsf{LE}}(N, m, \delta) \coloneqq \exp\left(2\log N + 3 - \frac{m\delta^2}{25\log N}\right)$$

By straightforwardly applying Theorem 8 and Lemma 13, we obtain the following bound on the ancestral robustness of DRSample.

Theorem 10 (Ancestral Robustness of DRSample). *Fix constants $0 < p, f, c < 1$ satisfying $1 - 2p - 5f/4 - \sqrt{20c/3} > 0$ and let $\mathsf{DRS} \leftarrow \mathsf{DRSample}(N)$. With probability at most*

$$p_{\mathsf{DAR}}(N, m, p, c, f) \coloneqq p_{\mathsf{LE}}\left(N, m, \frac{1 - 2p - 5f/4 - \sqrt{20c/3}}{10}\right),$$

DRS is $(pN/m, cN^2/m, f)$-ancestrally robust, where p_{LE} is defined in Lemma 13.

Proof. By Theorem 8, for DRS to be $(pN/m, cN^2/m, f)$-ancestrally robust, it suffices for DRS_m to be a δ-local expander with $\delta = \frac{1-2p-5f/4-\sqrt{20c/3}}{10}$. We have that $\delta > 0$ whenever the inequality in the theorem statement is satisfied, so $\delta > 0$ for all such assignments p, f, and c. Finally, from Lemma 13, DRS is $(pN/m, cN^2/m, f)$-ancestrally robust with probability at least $1 - p_{\mathsf{LE}}(N, m, \delta)$. □

E.2 Combining DRSample and Grates

Now that we have shown that DRSample is ancestrally robust with high probability. All that is left is to combine the properties of DRSample and grates. The grates construction is a family of $(\Omega(N), \Omega(N^{1-\varepsilon}))$-depth robust graphs.

Theorem 11 (Grates [Sch83]). *For every $N \geq 2$ and $0 < \varepsilon < 1$, there exists a graph $\mathsf{Grates}(N, \varepsilon)$ on N nodes that is $(\gamma N, \gamma' N^{1-\varepsilon})$ depth-robust and has indegree two.*

Grates was not shown to have the fractional depth-robust that we need, but we prove a folklore claim that shows that if our graph is $(\gamma N, \gamma' N^{1-\varepsilon})$-depth robust, it is $(\gamma N/2, \gamma', \gamma/2)$-fractionally depth robust.

Lemma 14 (Depth-robustness to Fractional Depth-robustness). *If $G = ([N], E)$ is (e, d)-depth robust with $e > d$, then G is $\left(e/2, d, \frac{e}{2N}\right)$-fractionally depth-robust.*

Proof. Let S be a set of size $e/2$. Let $V = \emptyset$, and consider the procedure where we find a path $u_1, \ldots, u_d$ in $G - S - V$, and then add $V \leftarrow V \cup \{u_d\}$. Such a path must exist for $e/2$ executions of the procedure (by normal depth robustness), so there are at least $e/2$ nodes of depth at least d in $G - S$. □

Now that we have two graphs that have all of the necessary combinatorial properties, we'd like to combine them. Intuitively, we could simply sample $\mathsf{DRS} \leftarrow \mathsf{DRSample}(N)$ and let EGS be $\mathsf{DRS} \cup \mathsf{Grates}(N, \varepsilon)$. The problem is that some nodes would have indegree more than three, while in practice it's important for them to have indegree at most two. A standard technique for addressing this problem is *indegree reduction*. For example, in [ABP17], they show that one can reduce the indegree from some constant δ to two, roughly by replacing each node v with a line graph $v_1 \to v_2 \ldots$, having an incoming edge to v_i corresponding to the ith parent of v in G. We define a similar process for indegree reduction, and show that it preserves ancestral robustness and depth-robustness.

Definition 15 (Indegree Reduction). *Let $G = (V = [N], E)$ be a supergraph of $\mathsf{LineGraph}(N)$ with indegree $\delta + 1$. The graph $\mathsf{reduce}(G) = (V_{\mathsf{Red}} = [\delta N], E_{\mathsf{Red}})$ is an indegree two graph defined by*

- *For $i \in [N]$ and $k \in \delta$, let $v_k^i = \delta(i-1) + k$. The vertex set V_{Red} of size δN and consists of nodes $v_1^i, v_2^i, \ldots, v_\delta^i$ for every $i \in V$.*
- *The edge set E_{Red} contains the edges of the line graph $\mathsf{LineGraph}(\delta N)$ as well as the following non-line graph edges. For each node $i \in V$ with (non-predecessor) parents $j_1, \ldots, j_k \in V \setminus \{i-1\}$, E' contains $(v_\delta^j, v_1^i), \ldots, (v_\delta^j, v_k^i)$.*

Showing that the reduced version of a graph retains depth-robustness is a straightforward application of the definition of indegree reduction and depth-robustness.

Lemma 15 (Reduced Depth Robustness). *If a graph $G([N], E)$ with indegree δ is (e, d)-depth robust, then $G' = \mathsf{reduce}(G)$ is $(e, \delta d)$-depth robust.*

Proof. Let $S \subseteq [\delta N]$ of size at most e. Define $S \subseteq [N]$ such that $S' = \left\{ s \mid \text{for some } j, \text{ the node } v_j^s \in S \right\}$. By construction $\left|S'\right| \leq e$. By the (e, d)-depth robustness of G, there exists a path $s = s^1 \to s^2 \to \ldots s^d = t$ in $G - S'$. We show that there is a length δd path P from v_1^s to v_δ^t in $\mathsf{reduce}(G) - S$. By Definition 15, the path P contains the intervals $v_1^{s^i} \to v_2^{s^i} \to \cdots \to v_\delta^{s^i}$ for $i \in [d]$. Likewise, the path P also contains the edges $v_\delta^{s^j} \to v_1^{s^{j+1}}$ for each $j \in [d-1]$, connecting the d intervals. Since S contains no nodes in the path, the path P is contained in $\mathsf{reduce}(G) - S$. □

The next property of reduced graph is more challenging. We want to show that for any graph G, $\mathsf{cc}(\mathsf{reduce}(G)) = \Omega(\mathsf{cc}(G))$. This will help us show later

that G a and $\mathsf{reduce}(G)$ satisfy ancestral robustness with similar parameters. We prove this lemma by providing a mapping from pebblings of $\mathsf{reduce}(G)$ to pebblings of G.

Theorem 12 (Indegree Reduction Preserves Cumulative Complexity). *For any DAG G, $\mathsf{cc}(\mathsf{reduce}(G)) \geq \mathsf{cc}(G)/\delta$.*

Proof. Let P' be a CC-minimizing pebbling for $\mathsf{reduce}(G)$. Our goal is to construct a similar cost pebbling P for G. The main hurdle is the fact that to place a pebble on node $v \in V$, we need to have pebbles on $\mathsf{Parents}(v, G)$; for P' to place a pebble on any node v_k^j, there need not be a pebble on all intervals corresponding to each parent in $\mathsf{Parents}\,(v, G)$. Therefore, we will keep around these pebbles to insure that P is legal. To aid in our proof, we define $\mathsf{orig}(v_k^j)$ to be the node $j \in G$. We let

$$\mathsf{parent}^\star\left(v_k^j, \mathsf{reduce}(G)\right) = \mathsf{Parents}\left\{v_k^j, \mathsf{reduce}(G)\right\} \setminus \left\{v_{k-1}^j\right\}$$

denote the incoming edge to v_k^j which corresponds to some $u \in \mathsf{Parents}(j, G)$. Similarly, we let $\mathsf{Parents}^\star\left(j, \mathsf{reduce}(G)\right) = \bigcup_{i\in[\delta]} \mathsf{parent}^\star\left(v_k^j, \mathsf{reduce}(G)\right)$.

The pebbling $P = (P_1, \ldots, P_T)$ is constructed as follows:

1. $P_1 \coloneqq \mathsf{orig}(P_1')$.
2. For $i > 1$, let
$$P_i^{\mathsf{add}} \coloneqq \left\{j \in \mathsf{orig}\left(P_i' \setminus P_{i-1}'\right) \mid \mathsf{Parents}(j, G) \subseteq P_{i-1}\right\}$$
denote the nodes in G corresponding to the pebbled nodes pebbled by P' in round i, except for those that are still illegal to pebble in G. This happens when P' has placed a pebble on some v_k^j with $k < \delta$, but hasn't yet placed a pebble on v_δ^j.
3. For $i > 1$, let
$$P_i^{\mathsf{del}} \coloneqq \left\{j \mid v_1^j, \ldots, v_\delta^j \notin P_i'\right\} \setminus \mathsf{orig}\left(\mathsf{Parents}(P_i', \mathsf{reduce}(G))\right)$$
denote the nodes in G which are no longer needed. The nodes that correspond to P_i^{del} in $\mathsf{reduce}(G)$ have no pebbles on them; the nodes of P_i^{del} also contain nodes that are no longer needed in G because they are no longer the parents of nodes corresponding to the pebbled nodes of P_i'.
4. $P_i = \left(P_{i-1} \cup P_i^{\mathsf{add}}\right) \setminus P^{\mathsf{del}}$.

Let us prove the legality of P. The first step is legal since the nodes in P_1' have no parents by the legality of P'. Now suppose P is legal for steps $1, \ldots, i-1$; we will now analyze the transition from P_{i-1} to P_i. Since removing pebbles is always legal, we just need to make sure $P_{i-1} \to P_{i-1} \cup P_i^{\mathsf{add}}$ is legal. Furthermore, since P_{i-1} contains all nodes whose parents are in P_i^{add}, we just need to show that

every node $P_i^{\mathsf{add}} \setminus \mathsf{orig}(P'_i)$ will be pebbled in P before it is used (as a parent) to place a pebble on a node.

The challenge is that a node $v_k^j \in P'_i \setminus P'_{i-1}$ could be added in round i, but in P we do not have a pebble on j nor would it be legal add this pebble in round i. For example, suppose j has parents $p_1, p_2, p_3 \in [N]$, and the edges $(v_\delta^{p_i}, v_i^j)$ are in G'. It could be the case that $v_\delta^{p_1}$ was pebbled so that v_1^j could be pebbled, but $v_\delta^{p_3}$ isn't pebbled until many round later. During this time between when v_1^j and $v_\delta^{p_3}$ are pebbled in P', we cannot have a pebble on j in P because it is possible that we may not have had any pebbles on the parent $p_3 \in [N]$ of j. However, when P' eventually pebbles v_3^j, there must have been pebbles on all parents $v_\delta^{p_i}$ of j at some point during the pebbling. That is why in P, we always keep around pebbles that correspond to the parents of those nodes j if we have not finished pebbling the interval $v_1^j, \ldots, v_\delta^j$.

More generally, $P_{i-1} \to P_i$ is legal since for j to be used to pebble a node in P_i, it must be the case that v_δ^j is pebbled in P'_{i-1}. For v_δ^j to be pebbled in P'_{i-1}, all of its ancestors must have been pebbled while there was still a pebble on some node in $\{v_1^j, \ldots, v_\delta^j\}$.

Lemma 16. *Fix a node $j \in V$. If $v_\delta^j \in P'_i$, then $v \in P_i$ and is a legal placement from the pebbling configuration P_{i-1}.*

Proof. Let $p = \left|\mathsf{Parents}(j, G)\right|$. For $k \in [p]$, let $t_k \leq i-1$ denote the latest time step prior to round i in which there was a pebble on $\mathsf{parent}^\star\left(v_k^j, \mathsf{reduce}(G)\right)$; these must exist due to the legality of P'. Furthermore, since v_δ^j is eventually pebbled, it must be the case that for all $\min_k t_k \leq i' \leq \max_k t_k$, a node in $\mathsf{reduce}(G)$ corresponding to v was pebbled in P'; that is, $P_{i'} \cap \{v_1^j, \ldots, v_\delta^j\} \neq \emptyset$. By the construction of P, this means that pebbles remained on each $\mathsf{orig}\left(\mathsf{parent}^\star\left(v_k^j, \mathsf{reduce}(G)\right)\right)$ between time t_k and $i-1$. Since $\mathsf{Parents}(v, G) = \mathsf{orig}\left(\mathsf{Parents}^\star\left(v_k^j, \mathsf{reduce}(G)\right)\right)$, it follows that placing a pebble on node j from the pebbling configuration P_i is a legal pebbling move. Finally, since $v_\delta^j \in P_i$, it follows that $j \in P_i^{\mathsf{add}}$ and $j \notin P_i^{\mathsf{del}}$. □

Now we know that each step of $P = (P_1, \ldots, P_T)$ is legal. Finally, P is indeed a pebbling since if s is a sink of G, v_δ^s is a sink in $\mathsf{reduce}(G)$. Since every sink of $\mathsf{reduce}(G)$ is in P'_T, every sink of G is in P_T.

Finally, we must analyze the cumulative complexity. By construction, the time complexity of P and P' is the same. However, $\left|P'_i\right|$ could be as much as $\delta|P_i|$ since we also keep around the parents of nodes in P'. Thus, $\mathsf{cc}(P) \leq \delta\mathsf{cc}(P')$. □

Now that we know that indegree reduction preserves CC, it is straightforward to show that it also preserves ancestral robustness. The ideas are similar to the proof of Lemma 11, but are made easier by the fact that δ is a constant.

Corollary 3 (Indegree Reduction Preserves Ancestral Robustness). *If a graph G is (a, C, f)-ancestrally robust, then* $\mathsf{reduce}(G)$ *is $(a, C/\delta, f/\delta)$-ancestrally robust*

Proof. Let $G = (V, E)$ be an (a, C, f)-ancestrally robust graph, $G' = (V', E') = \mathsf{reduce}(G)$, and A' be a set of vertices of G' with size at most a. We want to show that there are $\delta f N$ nodes in $G' - A'$ whose ancestors have cumulative complexity at least C. Let $A = \mathsf{orig}(A')$. We will use some notation and ideas from Lemma 11. First, notice that $\mathsf{reduce}(G \ominus A)$ is a supergraph of $G' - A' = \mathsf{reduce}(G) - A'$. Since $|A| \leq a$, it follows that there are fN nodes j in $G - A$ whose ancestors have cumulative complexity at least C. Next, we want to examine $\mathsf{Ancestors}(v^j_\delta, G' - A')$. Notice again that $\mathsf{reduce}(G \ominus A \ominus V \setminus \mathsf{Ancestors}(j, G - A))$ is a supergraph of $G' \ominus A' \cap \mathsf{Ancestors}(v^j_\delta)$. By Theorem 12, it follows that $\mathsf{cc}(\mathsf{Ancestors}(v^j_\delta, G' - A')) \geq \mathsf{cc}(\mathsf{Ancestors}(j, G - A))/\delta \geq C/\delta$. Thus, G' is $(a, C/\delta, f/\delta)$-ancestrally robust. □

By applying *Corollary* 3, *Lemma* 15 and *Lemma* 14 to *Theorem* 10 and *Theorem* 11. We are able to prove Theorem 9.

Proof of Theorem 9. Notice that $\mathsf{indegree}(\mathsf{DRS} \cup \mathsf{Grates}(N, \varepsilon)) = 3$. By Corollary 3, for EGS to be $(pN/m, c'N^2/m, f')$-ancestrally robust, it suffices for $\mathsf{DRS} \cup \mathsf{Grates}(N, \varepsilon)$ to be $(pN/m, cN^2m, f)$-ancestrally robust for $f = 2f'$ and $c = 2c'$. By Theorem 10 that DRS is $(pN/m, cN^2/m, f)$ with probability at least $1 - p_{\mathsf{DAR}}(N, m, p, \delta c, \delta f)$. By Theorem 10, when p, c, and f are fixed and satisfy $\delta = \frac{1-2p-5f/12-\sqrt{20c/9}}{10} \geq 0$, we have $p_{\mathsf{DAR}}(N, m, p, \delta c, \delta f) \leq 21N^2 \exp(\frac{\delta m}{2500})$.

By Theorem 11, we know that $\mathsf{Grates}(N, \varepsilon)$ (and thus $\mathsf{DRS} \cup \mathsf{Grates}(N, \varepsilon)$) is $(\gamma N, \gamma' N^{1-\varepsilon})$-fractionally depth robust. By Lemma 15 $\mathsf{EGS} = \mathsf{reduce}(\mathsf{DRS} \cup \mathsf{Grates})$ is also $\left(\gamma N, \gamma' N^{1-\varepsilon}\right)$-fractionally depth-robust. Finally, by Lemma 14, EGS is $(\gamma N/2, \gamma' N^{1-\varepsilon}, \gamma N^{1-\varepsilon}/2\delta)$-fractionally depth-robust. □

F Fractional Depth-Robustness in the PROM

In this section we review the techniques from [ACP+17] that we need for our PROM proofs. For the rest of this section, assume $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$ for some (e, d, f)-fractionally depth-robust graph G on N nodes. We will assume that every node in G has a unique set parents. That is, for all $u \neq v \in [N]$ we have $\mathsf{Parents}(u) \neq \mathsf{Parents}(v)$. We will be using the notation and definitions related to computing MHFs in the parallel random oracle model from Section A.3. While we will think of random oracles as accepting an arbitrarily large input, we will fix some u sufficiently large, and sample random oracles $H : \{0,1\}^u \to \{0,1\}^w$.

In [ACP+17], the authors show that Scrypt, defined by

$$\mathtt{Scrypt} = \mathsf{Dynamize}(\mathsf{LineGraph}(N), N_{\mathsf{chal}})^9$$

has maximal $\Omega(wNN_{\mathsf{chal}})$ CMC in the in the parallel random oracle model. An integral component in the proof of this result is a time-space trade-off lower bound for low space adversaries evaluating Scrypt. Suppose $\mathcal{A}$ is an algorithm with oracle access to $H : \{0,1\}^* \to \{0,1\}^w$ evaluating $f_{\mathcal{S},H}$ on X. It is shown that if $\mathcal{A}$ has only allocated $|\sigma_{s_i}| = O(we)$ space at start of the ith challenge, then with probability about $1/2$, $\mathcal{A}$ must spend $\Omega(N/e)$ steps to answer the challenge (submit the prelabel to the ith challenge to H). Intuitively, this means the naive strategy of storing the prelabel for every N/eth node in the first half of $\mathcal{S}$ is optimal in minimizing space-time complexity.

We are interested in reproving this result as to take advantage of fractional depth-robustness. From this perspective, we see that $\mathsf{LineGraph}(N)$, the first half of Scrypt, is $(e, N/e, 1/2)$-fractionally depth-robust, and we saw that if the algorithms state has significantly less than we bits, then the time to answer the challenge is at least N/e with probability about $1/2$. In general, we will show that if G is (e, d, f)-fractionally depth-robust, then an evaluation algorithm with significantly less than we space allocated at the start of a challenge must spend at least d steps to answer the challenge an f-fraction of the time.

This extension only requires minor changes to the original proof. In fact, in the last subsection of Section 2 of [ACP+17], they describe how one would generalize their proof to our setting, where if the algorithm's state is sufficiently small, an f-fraction of the remaining nodes have depth d. This result was only described and not formally proven in [ACP+17] because they were concerned only with CMC(rather than SSC) and Scrypt already has maximal CC. First, we prove this trade-off for the "single challenge" case, where the evaluation algorithm is given the uniformly random challenge node, for intuition. Then we focus on the more general case, where the ith challenge is generated by the randomness of the corresponding prelabel.

F.0.1 Single-Challenge Trade-off Given an arbitrary starting state σ_0, random oracle $H : \{0,1\}^u \to \{0,1\}^w$, and random challenge $c \sim [N]$, we will

[9] Scrypt is canonically defined with $N_{\mathsf{chal}} = N$. We use the N_{chal} for the ease of analysis.

consider a PROM algorithm $\mathcal{A}^H$ on inputs c and $X \in \{0,1\}^u$, which outputs the label $X_c = H(\mathsf{prelabD}(X_c))$ for node c in G. Let $\mathsf{trace}(A, H, \sigma_0, X, c) = (\sigma_1, \ldots, \sigma_T)$ denote the trace of A^H on inputs X and c. We want to show that if $|\sigma_0|$ is significantly less than $w \cdot e$, then with probability at least f, $T \geq d$.

Let us describe some useful notation; we use the same notation as in the analogous proofs in [ACP+17]. The variable π_{ic} describes the first time during the execution of $A^H(\sigma_0, c, X)$ that X_i appears as either an input or output of H. If X_i first appears as an input to H in round k (meaning $X_c \in \mathbf{q}_k$) then we let $\pi_{ic} = k$. If X_i is first seen as a response of a query containing $\mathsf{Parents}(i)$ in round k (meaning $\mathsf{Parents}(i) \in \mathbf{q}_k$ and $X_c \in H(\mathbf{q}_k)$), then we let $\pi_{ic} = k + 1/2$. Otherwise, if X_i was never queried or appeared as an output from an input not equal to $\mathsf{Parents}(i)$, then we let $\pi_{ic} = \infty$. Note that $t_c = \pi_{cc}$.

Let $\beta_i = \min_c \pi_{ic}$ denote the first time in which X_i is seen across all challenges (all choices of $c \in [N]$). If β_i is an integer, then we say i is blue (as in it came out of the blue because it was submitted as a query by A without ever querying the labels of i's predecessors). Let B be the set of blue nodes.

We'll make use of the following information-theoretic guarantee.

Lemma 17 (Predicting Outputs [ACP+17]). *Let P be an algorithm that has oracle access to a random oracle $H : \mathcal{D} \to \mathcal{R}$ and outputs pairs $(x_1, y_1), \ldots, (x_p, y_p)$ that have never been queried to H. The probability that $H(x_i) = y_i$ for all $i \in [p]$ is at most $|R|^{-p}$.*

At a high level, we want to construct a predictor P, which, given an input state σ_0 and a short hint, can predict the output of $p \leq |B|$ points. Lemma 17 says that if P can predict many points, then $|\sigma_0|$ is large.

The predictor works as follows. The input is σ_0 of size M. For the hint, we fix $p \leq |B|$ blue indices. The hint consists of, for each of the p blue nodes, 1) the challenge it is first seen in ($\log N$ bits), 2) the round within that challenge it is first seen as an input to H ($\log q$ bits), and 3) its position in the query at that round ($\log \delta$ bits). This hint is of size $p(2 \log N + \log \delta + \log q)$. The predictor runs A with initial state σ_0 on every challenge in parallel, and aims to fill out a table of every node and it's label. For every query in round k across all queries batches from every challenge, P 1) records on the table every one of the p blue nodes that appear as an input, 2) uses the table to answer every possible query, and 3) answers the rest of the queries using H.

Lemma 18. *If $p \leq |B|$, then P successfully predicts the output of H on p points.*

These points need to be distinct for P to win the game in Lemma 17. However, collisions between the X_i only happen with probability at most $\frac{N^2}{2^{w+1}}$. The next lemma shows that for almost all oracles, the set of blue nodes isn't too large.

Lemma 19. *Given adversary A, there exists a set of random oracles good such that $\Pr_H[H \notin \mathsf{good}] \leq qN^3 2^{-w}$, and for every $H \in \mathsf{good}$, every M, and every initial state σ_0 of A of size at most M bits, $|B| \leq p - 1$, where $p = \lceil \frac{(M+1)}{w - 2\log N - \log \delta - \log q} + 1 \rceil$*

Proof. Following [ACP+17], we observe that if $|B| > p-1$, then either there's a collision among the X_i or P successfully wins the prediction game in Lemma 17. By the birthday bound, $|\mathsf{collision}| \leq hn^2 2^{-w}/2$.

Let $\mathsf{predictable}$ denote the set of oracles in which P succeeds on for all values of M and $p \in [2:N]$.

Let M_p be the largest input length that yields that particular p. There are 2^{M_p+1} initial states of size at most M_p. Since by Lemma 17 each state-hint pair, P succeeds with probability at most 2^{-pw}, it follows that there are at most $h2^{M_p+1+p(2\log N+\log\delta+\log q-w)}$ oracles in which P succeeds on with *some* hint. By our definition of p, the quantity is at most $h2^{2\log N+\log\delta+\log q-w} = h\delta q N^2 2^{-w}$. This bounds, for a single p, the number of state (of size at most M_p) and hint (of size $p(2\log N + \log\delta + \log q)$), the number of oracles in $\mathsf{predictable}$. Union bounding over all p, we get that $|\mathsf{predictable}| \leq h\delta(N-1)N^2\delta q 2^{-w}$. □

Now we show that blue nodes act similarly to pebbles.

Lemma 20. *For all i, $t_i \geq \mathsf{depth}(i, G \cap B \cap \mathsf{Ancestors}(i)) + 1$.*

Proof. If i is blue, then $t_i \geq 0$ by definition. Now suppose otherwise.

For the base case, suppose $\mathsf{depth}(i, G \cap B \cap \mathsf{Ancestors}(i)) = 0$. Then the parents of i are all blue. For each $j \in \mathsf{Parents}(i)$, X_j is used to as an input to the query in which X_i is seen for the first time (as a response). Thus, $\beta_j < \beta_i - 1/2$ for all j.

There is some $j \in \mathsf{Parents}(i)$ with depth one less than that of i. This means $\beta_j < \beta_i$ since X_j must be used as an input to the query that yields X_i. Since both i and j are not blue by our assumption, $\beta_j \leq \beta_i - 1$. So, $t_i = \pi_{ii} \geq \lceil \beta_i \rceil \geq \mathsf{depth}(i, G \cap B \cap \mathsf{Ancestors}(i)) + 1$ □

Putting it all together, we have that if $p < e$, then the depth of the challenge node c in the graph $G \cap B \cap \mathsf{Ancestors}(c, G)$ is at least d with probability at least f.

Lemma 21 (Single-Challenge Trade-off). *Let $G = ([N], E)$ be an (e, d, f)-fractionally depth-robust graph such that every node has distinct parents. There exists a set of random oracles good such that $\Pr_H[h \notin \mathsf{good}] \leq qN^3 2^{-w}$, and for every $h \in \mathsf{good}$, the following holds: for every memory size M and every input state σ_0 of length at most M, either $p \geq e$ or with probability at least f, $t_j \geq d$.*

F.0.2 Multi-Challenge Trade-off In the last section, we had an algorithm A^H with some existing state. We picked the challenge $c \sim [N]$ and asked A^H to return the label for node c. If the state of A^H had size at most e, then with probability approximately f, A^H took at least d steps to respond. In this section we will prove a similar bound, except that the challenges will be issued in the way that is standard for data-dependent memory-hard functions. In particular, A^H will be computing the function $f^H_{\mathcal{G}}$, where $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$. Here, $\mathcal{G} = \mathsf{Dynamize}(G, N_{\mathsf{chal}})$, where $G = ([N], E)$ is some (e, d, f)-fractionally robust graph. For this section, we let $\mathcal{D} = \{\ell_1, \ldots, \ell_{N_{\mathsf{chal}}}\}$ be the set of N_{chal} dynamic

nodes (the last N_{chal} nodes of $\mathcal{G}$). Furthermore, we let $R(x) = x \mod N$. This is a shorthand for describing the dynamic predecessor to a node. The answer to the ith challenge (the dynamic prelabel to ℓ_i)is equal to $H(\ell_{i-1} \| 0 \| \mathsf{lab}(\ell_{i-1})) \mod N$. The algorithm A^H succeeds in computing $f_{\mathcal{G}}^H$ if it outputs the label for node $\ell_{N_{\mathsf{chal}}}$.

Let us first more explicitly review our notation for pre-labeling/labeling, which essentially follows [ABH17].

- For a node i, if $\mathsf{Parents}(i, \mathsf{G}) = \emptyset$ then $\mathsf{prelabS}(i) = i \| 0 \| X$. Otherwise, $\mathsf{prelabS}(i) = i \| 0 \| X_{\mathsf{Parents}(i,\mathsf{G})}$ and $X_i \coloneqq \mathsf{lab}(i) = H(\mathsf{prelabS}(i))$. Here, X_S for a set of nodes S is the concatenation of the labels X_i for $i \in S$ in lexicographic order.
- The pre-label for a node v is $T_v = v \| 1 \| X_{v-1} \| X_{R(X_v)}$, and $S_v = h(T_v)$, so S_v is the true (dynamic) label for node v.
- For a particular oracle H we will refer to X_i^H, S_i^H, and T_i^H when they are computed with respect to the oracle H. This will be excluded when the context is clear.
- We let s_k denote the first step of the kth challenge (the step after the $\mathsf{prelabS}(X_{\ell_k})$ is sent to the oracle to compute X_{ℓ_k}). t_k is defined as before: t_k is the number of steps between s_k and the step that T_{ℓ_k} is sent to the oracle.

For our result, we simulate A on a series of oracles, one for each challenge. Initially we run A on a random oracle H_0. Once A sends $\mathsf{prelabS}(\ell_1)$ to the oracle, we pick a random challenge c and construct the new oracle H_1, which is equivalent to H_0, except $H_1(\mathsf{prelabS}(\ell_1))$ is defined such that $R(X_{\ell_1}^{H_1}) = c$, and we repeat this for each $i \in [N_{\mathsf{chal}}]$. As long as these challenges are answered in the correct order, the execution of $A^{H_0}(X)$ and $A^{H_1}(X)$ is identical until the step in which $\mathsf{prelabS}(\ell_1)$ is sent to H. We repeat this for all $i \in [N_{\mathsf{chal}}]$. Once a query containing $\mathsf{prelabS}(\ell_1)$ is submitted to H, our predictor P can work similarly as before. Repeating the application of this predictor for each challenge in $[N_{\mathsf{chal}}]$, we achieve a trade-off for every challenge. Now we will go through the proof in more detail. To do this, we first need some more rigorous definitions of what we described above, leveraged directly from [ACP+17]. First, we describe the "bad" events. Intuitively, as long as certain bad events do not occur, our lower bound will hold.

- $\mathsf{ChangeMod}(S, j)$ returns the value S' such that $\lfloor S/N \rfloor = \lfloor S'/N \rfloor$ and $R(S') = j + 1$. If N is not a power of two, then it's possible for this value to be larger than 2^w. We say that $\mathsf{ChangeMod}$ *fails* on these values.
- The oracle H_0 is chosen uniformly at random. For $j \geq 1$ and challenges c_j, the oracle H_j is equal to H_{j-1}, except $X_{\ell_j}^{H_j} = \mathsf{ChangeMod}\left(X_{\ell_j}^{H_{j-1}}, c_j\right)$. In other words, The oracle H_j has the challenge c_j embedded in the output of the static label for node ℓ_j.
- $\mathsf{RoundingProb}_k$ is the set of random oracles H for which the value of at least one of $S_0^H, \ldots, S_k^H$ is greater than $\lfloor 2^w/N \rfloor \cdot N - 1$. These are the values in which $\mathsf{ChangeMod}$ fails. This can happen when N is not a power of two.

- $\mathsf{collision}_k^*$ is the set of all H where there is at least one collision among X, $X_1^{H_k}, \ldots, X_N^{H_k}$, and T_v for $v \in \mathcal{D}$ for $j \in [k]$. We let

$$\mathsf{collision}_k = \mathsf{RoundingProb}_{\min\{k, N_{\mathsf{chal}}-1\}} \cup \mathsf{collision}_k^*$$

- For every k, if $H_k \notin \mathsf{collision}_k$, we let $H_{k,j}$ be an oracle equal to H_k at every point except $H_{k,j}(T_{\ell_{k-1}}^{H_{k,j}}) = \mathsf{ChangeMod}(S_{\ell_{k-1}}^{H_{k-1},j}, j)$.
- The set $\mathsf{predictable}$ again consists of all random oracles for which there exists an input state σ_{s_k} of size m_k (for which $1 \leq p \leq N$), and a hint of length $\log q + p_k \log(N N_{\mathsf{total}} \delta q)$, such that the (updated construction of) predictor $\mathcal{P}$ correctly outputs p_k distinct values among X_i^H for all nodes i without querying them.
- $\mathsf{WrongOrder}_k$ is the set of oracles H such that there exists j_1 and j_2 such that $j_1 < j_2 \leq k$ and, in the run of A^H, query $T_{j_2}^H$ occurs, while query $T_{j_1}^H$ does not occur before query $T_{j_2}^H$.

The predictor P functions similarly as before. The predictor P takes as input σ_{s_k}, and the same hint as before with a few differences. First, $\mathcal{G}$ has an additional N_{chal} nodes (which may come out of the blue). For this reason, let $N_{\mathsf{total}} = N + N_{\mathsf{chal}}$. The new hint consists of p strings each containing:

1. $\log N_{\mathsf{total}}$ bits to describe which node is blue,
2. $\log N$ bits to describe which challenge the node is first blue in,
3. $\log q$ bits to describe which query the node is blue in, and
4. $\log \delta$ bits to describe the position of the blue node within the query.

Now, there is a final $\log q$ bits to determine which query contains the query whose output is the challenge (the predictor changes this output to simulate A on all challenges). The differences are:[10]

1. We give the predictor an additional $\log q$ hint to know which of the queries in $\mathbf{q}_{s_k}$ is X_{ℓ_k}.
2. p_k is now

$$p_k = \lceil \frac{|\sigma_{s_k}| + 1 + \log q}{w - \log(N N_{\mathsf{total}} \delta q)} \rceil$$

due to the hint changes.

Then, $\mathcal{P}$ simulates A on input σ_{s_k} with oracles $H_{k,1}, \ldots, H_{k,N}$, recording blue nodes as usual.

We will spend the rest of the section proving Theorem 5, which says that with high probability, the space-time trade-off holds for A on $\sim f N_{\mathsf{chal}}$ challenges. This is a weaker version (but generalized from line graphs to fractionally depth-robust graphs) of Theorem 5 in [ACP+17].

While proving Theorem 5, we match the claims/definitions to the ones in [ACP+17] when applicable. The first step is establish a claim analogous to the predictor argument in Lemma 21.

[10] In [ACP+17], they needed an additional bit of hint. This is because in their setting they computed T_j as $T_j = X_{\ell_j} \oplus X_{R(X_{\ell_j})}$. Since we adopt the label convention of [AS15] we don't need this extra hint.

Lemma 22 (Lemma 1). *Fix any challenge $k \in [N_{\mathsf{chal}}]$ and assume $H_{k-1} \notin \mathsf{collision}_{k-1} \cup \mathsf{predictable}$. Let $s_k > 0$ be the step after $X_k^{H_{k-1}}$ is sent to the oracle. Recall that challenge k is hard if either $p_k = \lceil \frac{\left|\sigma_{s_k}\right| + \log q + 1}{w - \log N - N_{\mathsf{total}} - \log \delta - \log q} \rceil \geq e$ or $t_k \geq d$. Then*

$$\Pr[k \text{ is hard}] \geq f.$$

Proof. If $p_k \geq e$ then k is already hard, so assume $p_k < e$. By employing the predictor P defined above on A with σ_{s_k} as described above, we get the bound corresponding to Lemma 21. □

The next lemma says that if there are no label collisions and the challenges are answered in the correct order, then A runs identically between the oracles $H_0, \ldots, H_{N_{\mathsf{chal}}}$. The proof is identical to that of [ACP+17].

Lemma 23 (Claim 9). *If for every $0 \leq j \leq n$, $H_j \notin \mathsf{collision}_j \cup \mathsf{WrongOrder}_j$, then for every k and $i \leq k$, $T_i^{H_{N_{\mathsf{chal}}}} = T_i^{H_k}$, and the execution of $A^{H_{N_{\mathsf{chal}}}}$ is identical to the execution of A^{H_k} until the query T_k is first made, which happens later than when query $T_{k-1}^{H_{N_{\mathsf{chal}}}} = T_{k-1}^{H_k}$ is first made.*

We now define the events that ensure Theorem 5 holds, and show that this is the case.

Definition 16 (Definition 7). *Let E_1 be the event that there are at least $H \geq N_{\mathsf{chal}}(f - \varepsilon)$ hard challenges. Let E_2 be the event that $H_k \notin \mathsf{collision}_k \cup \mathsf{WrongOrder}_k$ for every k and $A^{H_{N_{\mathsf{chal}}}}$ queries $T_{N_{\mathsf{chal}}}^{H_{N_{\mathsf{chal}}}}$. Let E_q be the event that $A^{H_{N_{\mathsf{chal}}}}$ makes no more than q total queries*

Lemma 24 (Claim 11). *If $E_1 \cap E_2 \cap E_q$, then there are at least $(f - \varepsilon)N_{\mathsf{chal}}$ indices ℓ_j such that:*

1. *The intervals $[s_k : s_k + t_k]$ for $k \in [N_{\mathsf{chal}}]$ are disjoint.*
2. *There are $(f - \varepsilon)N_{\mathsf{chal}}$ challenges i_j such that either $p_j \geq e$ or $t_j \geq d$.*

Proof. If E_2 holds, then A answers every challenge in order (so the intervals are disjoint), and each of the t_i and s_i are finite. By E_1, there are $(f - \varepsilon)N_{\mathsf{chal}}$ hard challenges, and for every hard challenge i, either $p_i \geq d$ or $t_i \geq d$. □

The next Lemma follows from [ACP+17] because we are using the same hint, except with one fewer bit.

Lemma 25 (Claim 12). *For every $k \in [N_{\mathsf{chal}}]$,*

$$\left|\sigma_{s_k}\right| \geq p_k \cdot (w - \log N - \log N - \log q - \log \delta - 1)/3$$

Next, we want to bound the probability that we pick a challenge c_k that causes a collision or causes H_{k-1} to be in $\mathsf{predictable}$.

Definition 17 (Bad Challenge). *The challenge c_k is **bad** if $H_{k-1} \in \mathsf{collision}_{k-1} \cup \mathsf{predictable}$. We let E_3 be the event that the number of challenges that are hard or bad is at least $(f-\varepsilon)N$. Let E_4 be the event that for every $0 \leq k < N_{\mathsf{chal}}$, $H_k \notin \mathsf{predictable}$.*

$E_2 \cap E_4$ means that no challenge is bad, so if $E_2 \cap E_3 \cap E_4$ hold then E_1 holds as well. The following Lemmas can are lifted almost exactly from [ACP+17], as their proofs are 1) invariant to the *proportion* of oracles satisfying the given property 2)

The next Lemma will allow us to focus our analysis on $H_{N_{\mathsf{chal}}}$ rather than H_k for every $k \in [N_{\mathsf{chal}}]$. Namely if some H_k has a collision or is in the wrong order, then so is $H_{N_{\mathsf{chal}}}$. The proof follows exactly as in [ACP+17].

Lemma 26 (Claim 14). *Given adversary A, if $H_k \in \mathsf{collision}_k \cup \mathsf{WrongOrder}_k$ for some $0 \leq k < N_{\mathsf{chal}}$, then $H_{k+1} \in \mathsf{collision}_{k+1} \cup \mathsf{WrongOrder}_{k+1}$ and so $H_{N_{\mathsf{chal}}} \in \mathsf{collision}_{N_{\mathsf{chal}}} \cup \mathsf{WrongOrder}_{N_{\mathsf{chal}}}$*

Next we bound the number of oracles which have collisions. This proof is different than that of [ACP+17] because of the fact that used a different labeling rule [ABP18]. The result is just that there are fewer collisions. We let h denote the number of oracles $H : \{0,1\}^u \to \{0,1\}^w$ for some $u \geq \delta w + \log N + 1$.

Lemma 27 (Claim 15). $\left|\mathsf{collision}^*_{N_{\mathsf{chal}}}\right| \leq h \cdot \binom{2N}{2} 2^{-w}$

Proof. By our labeling convention of G, $\mathsf{prelabS}(i) \neq \mathsf{prelabS}(j)$ for all $i \neq j \in [N]$. So, $\mathsf{prelabD}(i) \neq \mathsf{prelabD}(j)$ for all $i \neq j \in \mathcal{D}$. Thus, there are at most $h\binom{N+|\mathcal{D}|}{2}2^{-w} \geq h\binom{2N}{2}$ oracles in $\mathsf{collision}^*_{N_{\mathsf{chal}}}$. □

The next lemma bounds the number of oracles where $\mathsf{ChangeMod}$ doesn't work. It follow exactly as in [ACP+17]. The idea is that for every k, there are at most $N-1$ possible outputs which fail, and there are N_{chal} challenges.

Lemma 28 (Claim 16). $\left|\mathsf{RoundingProb}_{N_{\mathsf{chal}}-1} \setminus \mathsf{collision}^*_{N_{\mathsf{chal}}}\right| \leq hN_{\mathsf{chal}}(N-1)2^{-w}$

This lemma bounds the number of oracles in which A makes at most q queries but $A^{H_{N_{\mathsf{chal}}}}$ does not query $T^H_{N_{\mathsf{chal}}}$, assuming there are no label collisions. $\Pr[E_q] - \chi_q$ is the probability that A fails (so A didn't query $T_{N_{\mathsf{chal}}}$), so we just need to focus on the number of oracles in which $A^{H_{N_{\mathsf{chal}}}}$ succeeds yet doesn't query $T_{N_{\mathsf{chal}}}$. The trick from [ACP+17] is to notice that if α are the answers to A^H's queries, then the output of A depends only on α. That means that for any α, only oracles that have $H(T_{N_{\mathsf{chal}}})$ equal to this fixed output is just $h2^{-w}$.

Lemma 29 (Claim 17). *Given adversary A, the number of oracles $H \notin \mathsf{collision}^*_{N_{\mathsf{chal}}}$ such that E_q holds and $A^{H_{N_{\mathsf{chal}}}}$ does not query $T^H_{N_{\mathsf{chal}}}$ is no more than $h(\Pr[E_q] - \chi_q + 2^{-w})$, where χ_q is the probability (for a uniform H) that A^H is successful and makes no more than q queries.*

Lemma 30 bounds the number of oracles in which A makes at most q queries, a challenge is answered in the wrong order, but there are no collisions among the labels. The proof follows [ACP+17] exactly, except there are N_{chal} challenges instead of N.

Lemma 30 (Claim 18). *Given adversary A, the number of oracles H such that E_q holds and $H \in \mathsf{WrongOrder}_{N_{\mathsf{chal}}} \setminus \mathsf{collision}^*_{N_{\mathsf{chal}}}$ is no more than $hqNN_{\mathsf{chal}}2^{-w}$*

Lemma 31 (Claim 19). $\Pr[\bar{E}_3] \leq e^{-2\varepsilon^2 N_{\mathsf{chal}}}$

Proof. Let y_k be the indicator random variable of challenge k being hard. If $H_k \in \mathsf{collision}_{k-1} \cup \mathsf{predictable}$ then y_k is bad. Otherwise, by Lemma 22, if $p < e$, then k is hard with probability at least f, and if $p \geq e$ then k is always hard, regardless of the results of the previous rounds. Therefore, for any $b_1, \ldots, b_{N_{\mathsf{chal}}} \in \{0,1\}$,

$$\Pr_{c_k}[y_k = 1 \mid \forall j \in [k-1],\ y_j = b_j] \geq f.$$

By applying Lemma 39, there are at least $(f-\varepsilon)N_{\mathsf{chal}}$ hard challenges, except with probability at most $e^{-2\varepsilon^2 N_{\mathsf{chal}}}$ □

To finish the proof of Theorem 5, we will first upper bound the size of $\bar{E}_4$, the set of oracles such that for every $0 \leq k < N_{\mathsf{chal}}$ $h_k \notin \mathsf{predictable}$.

Lemma 32.

$$\Pr[\bar{E}_4] \leq \delta {N_{\mathsf{chal}}}^3 Nq2^{-w}$$

Proof. Like the single challenge argument, we get that

$$h \cdot 2^{(M_p+1+\log q)+p(\log N+\log N_{\mathsf{chal}}+\log\delta+\log q-w)} \leq 2^{\log N+\log N_{\mathsf{chal}}+\log q+\log\delta-w}.$$

By the definition of p an M_p. There are N_{chal} possible values of p, so

$$|\mathsf{predictable}| \leq hN_{\mathsf{chal}}2^{\log N+\log N^2+\log\delta+\log q-w} = h\delta {N_{\mathsf{chal}}}^2 Nq2^{-w}.$$

Union bounding over all N_{chal} oracles H_k, we have that

$$\Pr[\bar{E}_4] \leq \delta {N_{\mathsf{chal}}}^3 Nq2^{-w}.$$

□

Now we will complete the proof of Theorem 5. We have

$$\begin{aligned}
\Pr[E_2 \cap E_q \cap E_3 \cap E_4] &\geq \Pr[E_q] - \Pr[\bar{E}_2 \cap E_q] - \Pr[\bar{E}_3] - \Pr[\bar{E}_4] \\
&\geq \chi_q - \left(\binom{2N}{2} + N_{\mathsf{chal}}(N-1) + \delta {N_{\mathsf{chal}}}^3 Nq\right) 2^{-w} - e^{-2\varepsilon^2 N_{\mathsf{chal}}} \\
&\geq \chi_q - \left(2N^2 + N_{\mathsf{chal}}N + \delta {N_{\mathsf{chal}}}^3 Nq\right) 2^{-w} - e^{-2\varepsilon^2 N_{\mathsf{chal}}} \\
&\geq \chi_q - \frac{2N^2 + \delta q({N_{\mathsf{chal}}}^3 N + 1)}{2^w} - e^{-2\varepsilon^2 N_{\mathsf{chal}}}
\end{aligned}$$

To get Theorem 5, see that

$$\Pr[(E_2 \cap E_q \cap E_3 \cap E_4) \cup E_q] \geq \chi - \frac{2N^2 + \delta q({N_{\mathsf{chal}}}^3 N + 1)}{2^w} - e^{-2\varepsilon^2 N_{\mathsf{chal}}}$$

since $\Pr[E_q] + \chi_q \geq \chi$.. Thus, we let

$$p_{\mathsf{Dyn}}(N, N_{\mathsf{chal}}, w, \varepsilon) \coloneqq \frac{2N^2 + \delta q({N_{\mathsf{chal}}}^3 N + 1)}{2^w} - e^{-2\varepsilon^2 N_{\mathsf{chal}}}$$

G After-the-Fact Pebblings of After-the-Fact Graphs

In this section, we will show that the iMHF pebbling reduction from [AS15] also works for after-the-fact dynamic graphs. In particular, we want to show that if $\mathcal{A}$ computes $f_{\mathcal{G}}^H$ on X, then we can extract a CC-equivalent legal pebbling of the graph $\mathcal{G}_H$, the (static) graph that is fixed when H is fixed. It's also important that we ensure that the timestep in which each X_i and T_j are queried for the first time is preserved in P. That way, inferences made about the the size of memory state at some timestep j will have consequences in our after-the-fact pebbling in this section. These proofs work via the following information theoretic result.

Throughout this section, fix any dynamic pebbling graph $\mathcal{G}$ on N nodes and any input X. We will be choosing $H : \{0,1\}^* \to \{0,1\}^w$ uniformly at random. When H is fixed, so is a static graph $\mathcal{G}_{H,X}$ in the support of $\mathcal{G}$. In particular, $\mathcal{G}_{H,X}$ is the graph obtained by fixing each dynamic edge according to their labels (which is defined by X and H). Recall that $\mathsf{prelabS}(v)$ is the prelabel of the static nodes of v, while $\mathsf{prelabD}(v)$ is the (true) prelabel of v, including nodes from dynamic edges. We will say that an algorithm A^H computes $f_{\mathcal{G},H}$ if A makes the query to the prelabel fo the final node in $\mathcal{G}$. For concreteness, let guess denote the event that A^H outputs $f_{\mathcal{G},H}(X)$ without querying this prelabel. By Lemma 17 in the prior section, we see that $\Pr[\mathsf{guess}] \leq 2^{-w}$. Let us now proceed to the main argument.

We define the ex post facto pebbling sequence $P = (P_0, \ldots, P_t) \coloneqq \mathsf{extract}(A, H, X, G)$ of G from the trace of A^H on input X. First, we let $P_0 = \emptyset$. For $i \geq 1$, we define the set P_i as follows with respect to the ith query batch $\mathbf{q}_i$:

1. Initialize $P_i = P_{i-1}$
2. For each $y \in \mathbf{q}_i$, if $y = \mathsf{prelabD}(v)$ then add $v \in P_i$.
3. The node $v \in [N]$ is called ***necessary*** if there exists a $\mathbf{q}_j$ with $j > i$ containing a correct oracle call with v as an input-node (a query contains the label of v) but there is no $\mathbf{q}_k$ with $i < k < j$ containing a correct oracle call for v. Remove all $v \in P_i$ which are not necessary.

It's important to keep in mind that P (along with its space usage and time t) is a function of H. So, when we argue about the probability P satisfies some property, we are really "counting" the number of oracles H in which the corresponding pebbling $\mathsf{extract}(A, H, X, G)$ satisfies the property. We show that with high probability (over H), P is legal using the following information-theoretic result.

Lemma 33 ([ABP18]). *Let $B = (b_1, \ldots, b_u)$ be a sequence of random bits. Let $\mathcal{P}$ be a randomized procedure which gets a hint $h \in \mathcal{H}$, and can adaptively query any of the bits of B by submitting an index i and receiving b_i. At the end of the execution, $\mathcal{P}$ outputs a subset $S \subseteq [u]$ of $|S| = k$ indices which were not previously queried, along with guesses for all of the bits $\{b_i \mid i \in S\}$. Then the probability (over the choice of B and randomness of P) that these exists some $h \in \mathcal{H}$ for which $P(h)$ outputs all correct guesses is at most $\frac{|\mathcal{H}|}{2^k}$.*

The idea is that an illegal pebbling move is witnessed by a query to H. So if A^H on X produces such an illegal move, then there is a $\log q$ hint that allows a predictor to output an pair $(x, H(x))$ without querying x. By Lemma 33, only a $\frac{q}{2^w}$ fractions of oracles can have this property. [11]

Lemma 34 (After-the-Fact Pebbling Legality). *Fix an input X to $H : \{0,1\}^* \to \{0,1\}^w$, an algorithm A^H, and a dynamic pebbling graph $\mathcal{G}$. The pebbling $P = [P_0 = \emptyset, P_1, \ldots, P_T] \coloneqq \mathsf{extract}(A, H, X, G)$ is a legal sequence of pebbling moves for $G_{H,X}$ with probability at least $1 - \frac{q}{2^w}$ over the choice of H.*

Proof. We will construct a predictor that can guess the output of H on an input it hasn't queried if P has an illegal move (a node v is placed without it's neighbors already being pebbled). The predictor $\mathcal{P}$ takes as a hint the index $i' \in [q]$ of the first oracle call to v causing the illegal pebble placement. Suppose that this query takes place during round i (because $\mathcal{A}$ is parallel it is possible that $i < i'$). Because we place a new pebble on node v the value of query i' is $\mathsf{prelab}(v)$. In particular, we have $\mathsf{prelab}(v) \in \mathbf{q}_i$ but because it is illegal to place a pebble on node v there must some parent node w such that $w \notin P_{i-1}$ even though $(w, v) \in E(G_{H,X})$. We first note that $\mathsf{prelab}(w) \notin \bigcup_{j<i} \mathbf{q}_j$ i.e., the query $\mathsf{prelab}(w)$ is never submitted to the random oracle. Suppose for contradiction that the query $\mathsf{prelab}(w)$ had been previously submitted and let $j < i$ be the last round where $\mathsf{prelab}(w) \in \mathbf{q}_j$ before round i. In this case we would have placed a pebble on node w during round j i.e., $w \in P_j$ and node w would be ***necessary*** during every intermediate round j' such that $j < j' < i$. Thus, we would still have a pebble on node w during round P_{i-1}. Contradiction!

Our extractor $\mathcal{P}$ simulates $\mathcal{A}$ keeping track of the total number of random oracle queries made so far. This allows the extractor to identify the round i when i'th query to the random oracle is submitted. In rounds $j < i$ the extractor $\mathcal{P}$ simply forwards the random oracle queries $\mathbf{q}_j$ to the random oracle and passes the answer back to $\mathcal{A}$ to continue the simulation. Instead of submitting $\mathbf{q}_i$ to the random oracle the extractor uses the hint i' to locate the query $\mathsf{prelab}(v)$ and then parses this query to extract the label of node w i.e., $H(\mathsf{prelab}(w))$. Now the extractor $\mathcal{P}$ can use the honest evaluation algorithm to generate labels

[11] For the exact claim, we prove that for any *fixed graph* G, P is a *legal pebbling sequence* starting from $P_0 = \emptyset$. Even if A^H is computing some other function, we can still show that we can extract a legal pebbling sequence, even if it is just $[P_0 = \emptyset]$. This way, our claims will be true even if A^H fails to compute $f_{G,H}$ on X. When A^H does succeed in computing $f_{G,H}(X)$, then we will be able make inferences about the runtime via our extracted pebbling.

in $G_{H,X}$ in topological order until it recovers $\mathsf{prelab}(w)$ — because v appears after w in topological order we can generate $\mathsf{prelab}(w)$ without submitting the query $\mathsf{prelab}(v)$. Finally, the extractor outputs the pair $(\mathsf{prelab}(w), H(\mathsf{prelab}(w)))$ without querying the random oracle H at this point.

If there is an illegal placement, then there's a hint of length $\log q$ for which the extractor $\mathcal{P}$ succeeds. By Lemma 33, there exists such a hint with probability at most $\frac{q}{2^w}$. □

From the definition of our after-the-fact pebbling, we see that if the label for some node v is seen as an output for the first time at some step i, then $v \in P_i$. This allows up to make inferences about the ith step in the pebbling and the ith query in the algorithm it is extracted from.

Lemma 35 (After-the-Fact Pebbling vs. Algorithm Steps). *If the after-the-fact pebbling sequence P is legal for some PROM algorithm A^H and graph G, then a pebble is placed node v for the first time in step i ($v \in P_i$ and $v \notin P_j$ for all $j < i$) if and only if the prelabel of v is first seen as an input in to the ith query to the oracle.*

Now we show that the size of the ith pebbling step P_i lower bounds the size of the algorithm's state at time i.

Lemma 36 (After-the-Fact Pebbling cost Equivalence). *For $i \in [t]$, let $m_i = |\sigma_i|$ be the bit-length of the state σ_i. Then for any $\lambda \geq 0$:*

$$\Pr\left[\forall i \in [t] : |P_i| \leq \frac{m_i + \lambda}{w - 2\log q}\right] \geq 1 - 2^{-\lambda}$$

Proof. We proceed as in [AS15]. Oracle queries which cause a node $v \in P_i$ to be necessary are ***critical*** queries. Suppose there are $C \leq |P_i|$ critical calls. As before, we will construct a predictor that simulates A^H on σ_i to predict the output of nodes that come out of the blue. The only difference between this predictor and the one in the previous section is that we are doing this for every step i instead of just every challenge, and there is no need to run A on multiple inputs at once. This is because we've fixed the graph once we've fixed H. The hint of the predictor are indices $J = \{j_1, \ldots, j_C\}$ in increasing order and $K = \{k_1, \ldots, k_C\}$ of critical queries made by A and the indices k_ℓ where the prelabel of the node corresponding to the critical query j_ℓ, so the hint is of size $2C \log q$. While the predictor is simulating A, we say that a query is correct if it is a critical query or if the query consists of $\mathsf{prelabD}(v)$ for a node v, and a correct call for each parent of v has been made. The predictor keeps a cumulative table of node-label pairs as they become known, and does the following for each $\ell \in [2C]$ (in order):

- Find the ℓth smallest element of $J \cup K$
- If $\ell \in K$, simulate A until query $\ell - 1$. For each of A's query, the predictor uses the table when possible, and querying H otherwise. If a query is of the form $\mathsf{prelabD}(v)$ for some v, record the pair $(v, \mathsf{prelabD}(v))$.

– If $\ell \in J$, then examine query j_ℓ. Since $j_\ell > k_\ell$, the prelabel $\mathsf{prelab}(v)$ for the correponding node v is already recorded. Record and output $(\mathsf{prelab}(v), X_v)$.

Now, the predictor can correctly output guesses for the labels of the nodes in P_i given the correct hint.

The number of possible hints for C nodes is $2^{2C \log q}$, so by Lemma 33, the probability that $|P_i| \geq \frac{m_i+\lambda}{w-2\log q}$ is at most

$$2^{m_i+2|P_i|\log q-|P_i|w} \leq 2^{m_i-\frac{m_i+\lambda}{w-2\log q}(w-2\log q)} = 2^{-\lambda}$$

□

Proof. (of Theorem 6) Theorem 6 follows directly from Lemma 34, Lemma 35 and Lemma 36. We set $\lambda = w$ and define $p_{\mathsf{EPF}}(q) = \frac{1+q+q^2}{2^w}$ as our upper bound the probability of a bad event that $|P_i| > \frac{m_i+\lambda}{w-2\log q}$ for some round i or that the ex-post facto pebbling sequence is illegal. We note that the extra $1/2^w$ comes from the event guess, the event that A^H guesses $f_{\mathcal{G},H}(X)$ without querying the prelabel of the final node.

□

H Local Expansion of DRSample's Metagraph

Proof of Lemma 9. We first recall from prior works the notion of a ***good***-node with respect to a set S. Lemma 37 shows that there are always at least $N - 2|S|$ good nodes (with respect to S); Lemma 38 says that if G is a δ-local expander (with $\delta < 1/6$), then the set of good nodes are all connected. Putting it all together, we get that a δ-local expander satisfies the properties in the theorem statement.

Definition 18 (Good Node [ABP18], [EGS75]). *Let $G = ([N], E)$ be a metagraph of some directed graph G and $S \subseteq [N]$. A node $v \in [N]$ is c-good for some constant $c > 0$ if*

– *for all $r \in [v]$, $\left|I_r(v) \cap S\right| \leq cr$ and*
– *for all $r \in [m - v + 1]$, $\left|I^*_r(v) \cap S\right| \leq cr$.*

Lemma 37 ([ABP18]). *Let $G = ([N], E)$ be a DAG and fix $S \subseteq V$ and $0 < c < 1$. There are at least $N - \frac{1+c}{1-c}|S|$ c-good nodes with respect to S in $G - S$.*

We want $0 < \frac{1+c}{1-c} \leq 1/2$, so pick $c = 1/3$.

Lemma 38 ([ABP18]). *Fix a DAG $G = ([N], E)$ and constant $c > 0$. If G is a δ-local expander with $\delta < \min\left\{c/2, 1/4\right\}$. Then for any set $S \subseteq [N]$, the graph $G - S$ contains a directed path through all nodes in G which are c-good with respect to S.*

□

Proof of Lemma 13. Fix $k \in [N]$, $v \in \left[N/m\right]$, $A \subseteq I_v(k)$, and $B \subseteq I_v^*(k)$. Let $G = ([N], E) \sim \mathsf{DRSample}(N)$. Then, referring to the graph distribution in Definition 14,

$$\begin{aligned}\Pr_G\left[A \times B \cap E_m = \emptyset\right] &\leq \left(1 - \frac{|B|}{5k \log N}\right)^{m|A|/5} \\ &\leq \exp\left(-\frac{m|A| \cdot |B|}{25k \log N}\right)\end{aligned}$$

Now we want to union bound the above probability over all possible subsets. We only need to consider subsets of size δk, as they are the smallest subsets we need to consider for δ-local expansion. By Stirling's approximation

$$\binom{k}{\delta k} \leq \frac{\exp\left(\delta k \log \frac{1}{\delta} + (1-\delta)k \log \frac{1}{1-\delta}\right)}{2\pi e^{-1}\sqrt{\delta(1-\delta)k}}.$$

Simplifying, we see that $\delta \log \frac{1}{\delta} + (1-\delta) \log \frac{1}{1-\delta} \leq 1$. Thus, $\binom{k}{\delta k} \leq \frac{e^{k-1}}{\sqrt{\delta(1-\delta)k}}$. So, the probability that this expansion property doesn't hold for some subsets of $I_v^*(k)$ and $I_v(k)$ of size at least δk is at most

$$\begin{aligned}\binom{k}{\delta k}^2 \exp\left(-\frac{m|A| \cdot |B|}{25k \log N}\right) &\leq \frac{\exp\left(2k - \frac{m|A| \cdot |B|}{25k \log N}\right)}{\delta(1-\delta)k} \\ &= \exp\left(2k - \frac{m|A| \cdot |B|}{25k \log N} - \log \delta(1-\delta)k\right) \\ &\leq \exp\left(2k - \frac{m\delta^2 k}{25 \log N} - \log k + 1\right)\end{aligned}$$

Now, union bounding over $k \in [\lfloor N/m \rfloor]$, v is a δ-local expander, except with probability at most

$$\frac{N}{m} \cdot \exp\left(-\frac{m\delta^2}{25 \log N} + 3\right) \leq \exp\left(\log N - \frac{m\delta^2}{25 \log N} + 3\right)$$

Union bounding over all N/m nodes, we have that G_m is a δ-local expander, except with probability at most

$$p_{\mathsf{LE}}(N, m, \delta) \coloneqq \exp\left(2 \log N + 3 - \frac{m\delta^2}{25 \log N}\right)$$

□

Lemma 39 ([ACP+17] Claim 7). *If $V_1, \ldots, V_Q$ are binary random variables such that for any $0 \leq i \leq Q$ and any values of $v_1, \ldots, v_i \in \{0,1\}$,*

$$\Pr[V_{i+1} \mid V_1 = v_1, \ldots, V_i = v_i] \geq \rho,$$

then for any $\varepsilon > 0$, with probability at least $1 - e^{-2\varepsilon^2 Q}$,

$$\sum_{i \in [Q]} V_i \geq Q(\rho - \varepsilon).$$

I Sustained Space/Cumulative Complexity Trade-offs for Argon2id

Argon2id [BDK16] is a widely deployed data-dependent memory-hard function. Here, we briefly show how we can apply our techniques to prove an SSC/CC lower bound for Argon2id.

The static portion of Argon2id is called Argon2i; Argon2i itself is used as a data-independent memory-hard function.

Definition 19 (Argon2i Graph [BH22,?]). *The underlying DAG $G = ([N], E) \sim \mathsf{Argon2i}_N$ for Argon2i is sampled from a graph distribution $\mathsf{Argon2i}_N$ with the following properties:*

- *G has N nodes and is a supergraph of $\mathsf{LineGraph}(N)$,*
- *Every node $v > 2$ has an additional parent $\mathsf{parent}^\star(v) \in [v-2]$ that is sampled from a distribution such that*

$$\Pr[\mathsf{parent}^\star(v) = u] = \Pr_{x \sim [M]}\left[v\left(1 - \frac{x^2}{M^2}\right) \in (u-1, u]\right]$$

for some $M \gg N$.

A useful fact is that Argon2i's edge distribution is very close to uniform.

Lemma 40 (Claim 5 of [BZ17]). *Let $G \sim \mathsf{Argon2i}$. For all $v \in [N]$ and all $u \in [v-2]$ we have*

$$\Pr[\mathsf{parent}^\star(v) = u] \geq \frac{1}{4N}\ .$$

We will analyze $\mathsf{Argon2iDyn}_N = \mathsf{Dynamize}(\mathsf{Argon2i}_N, N)$.

Remark 2. An astute reader may note that the definition of Argon2id is not completely equivalent to $\mathsf{Argon2iDyn}_N$. The static portions of the graph is identical in both cases, but the dynamic edge distribution is slightly different. Specifically, for each node $v > N$ in $\mathsf{Dynamize}(\mathsf{Argon2i}_N, N)$ includes a dynamic edge $(\mathsf{parent}^\star(v), v)$ where $\mathsf{parent}^\star(v) \in [N]$ is selected uniformly at random from $[N]$. By contrast, in Argon2id the dynamic edge $(\mathsf{parent}^\star(v), v)$ will pick

$\mathsf{parent}^\star(v)$ from a non-uniform distribution over $[v-2]$. In particular, for each node $u \in [v-2]$ we have

$$\Pr[\mathsf{parent}^\star(v) = u] = \Pr_{x \sim [M]}\left[v\left(1 - \frac{x^2}{M^2}\right) \in (u-1, u]\right] ,$$

this is the same edge distribution for Argon2i except that the edge $(\mathsf{parent}^\star(v), v)$ is ***dynamic*** when $v > N$. Thus, while the distribution is non-uniform by Lemma 40 we still have $\Pr[\mathsf{parent}^\star(v) = u] \geq \frac{1}{4(2N)} \geq \frac{1}{8N}$. Thus, up to a small change in constant factors, all of our analysis for $\mathsf{Argon2iDyn}_N$ also applies to Argon2id. This includes our dynamic pebbling analysis and our PROM analysis.

We obtain the following sustained space/cumulative complexity trade-offs for Argon2id.

Theorem 13 (Argon2id Pebbling Bound). *Fix any dynamic pebbling strategy $\mathcal{S}$ and constants $2/3 < \beta < 1$ and $0 < \varepsilon < 1/2$. Suppose $G' \sim \mathsf{Argon2i}$ satisfies the fractional depth-robustness of Theorem 15 and the ancestral robustness of Lemma 42, which happens with probability at least $1 - o(1/N)$. Then except, except with negligible probability over the choice of $G \sim \mathsf{Dynamize}(G', N)$, $\mathcal{S}(G)$ either sustains $e = N^\beta$ pebbles for $\Omega(N)$ steps, or has cumulative complexity at least $\Omega\left(N^{4.5-3\beta-\varepsilon}\right)$.*

Let us look at a concrete bound via Theorem 13. Suppose we set $\beta = 3/4$ and $\varepsilon = 0.05$. Then we get (up to polylog factors) that any strategy sustains $N^{0.75}$ pebbles for N steps, or incurs CC $N^{2.2}$.

Theorem 14 (Argon2id PROM Bound). *Fix a graph $G' = ([N], E)$ with the fractional-depth/ancestral robustness properties of Argon2i as defined in Theorem 15 and Lemma 42, respectively. Let A be a PROM algorithm that succeeds with probability χ on $f := f_{\mathsf{Dynamize}(G',N)}$, where $H : \{0,1\}^* \to \{0,1\}^w$ is a random oracle. Fix any constants $0 < \varepsilon < 1/2$ and $2/3 < \beta < 5/6 - \varepsilon$. With probability at least $\chi - \exp\left(-\Omega(N)\right)$ over the choice of H, A either sustains $\Omega\left(wN^\beta\right)$ for $\Omega(N)$ steps incurs cumulative memory complexity at least $\Omega\left(N^{3.25-3\beta/2-\varepsilon/2}\right)$*

Again, using the example of $\beta = 3/4$ and $\varepsilon = 0.05$, we see that any PROM algorithm computing the MHF corresponding to $\mathsf{Argon2iDyn}_N$ either (up to polylog factors) sustains $wN^{0.75}$ space for N steps, or has cumulative complexity $wN^{2.1}$.

Let us now prove Theorem 13 and Theorem 14. The main task is to show Argon2i is ancestrally robust for suitably chosen parameters, as prior work has already bounded Argon2i's fractional depth-robustness.

Theorem 15 (Fractional Depth-Robustness of Argon2i [BZ17]). *Let $G \sim \mathsf{Argon2i}_N$. For some $\ell_N = \mathsf{polylog}^{-1}(N)$ and constants $0 < \varepsilon, f, c < 1$, except with probability $1 - o(1/N)$, G is*

$$\left(e, \ell_N \frac{N^3}{e^3}, f\right) \textit{-fractionally depth robust}$$

for all $e \geq cN^{2/3}$.

Since the edge distribution of Argon2i is close to uniform, it is straightforward to show that, except with negligible probability, the $\omega(\sqrt{N})$-metagraph G_m of Argon2i is the complete graph. The probability upper bound $\exp\left(-\frac{m^2}{200N} + 2\log N\right)$ from Lemma 41 exceeds 1 if $m \leq \sqrt{N}$. However, if $m \geq \sqrt{200N(\lambda + 2\log N)}$ then Lemma 41 implies that the meta graph G_m of $G = \mathsf{Argon2id}_N \cap [N]$ is complete except with probability $\exp(-\lambda)$. In particular, for $m = \Omega(\sqrt{N \log N})$ the meta graph G_m will be complete with probability $1 - o(1/N)$.

Lemma 41 (Metagraph of Argon2i). *Fix* $m < N$ *and sample* $G \sim \mathsf{Argon2i}_N$. *Except with probability at most* $\exp\left(-\frac{m^2}{200N} + 2\log N\right)$ *the meta graph* G_m *of* G *is the complete graph on* N/m *nodes.*

Proof. We need to show that for each $v \in [N/m]$ there is an edge from some node in M_u^{last} to M_v^{first} in G for all $u \leq v-2$. The probability metanodes u and v are not connected is at most $\left(1 - \frac{m}{40N}\right)^{m/5} \leq \exp\left(-\frac{m^2}{200N}\right)$. Thus the probability that G_m is complete is at least $1 - \binom{N}{2}\exp\left(-\frac{m^2}{200N}\right) \geq 1 - \exp\left(-\frac{m^2}{200N} + 2\log N\right)$. □

Note that if $G \sim \mathsf{Argon2iDyn}_N$ then the subgraph $G \cap [N]$ induced by the first N nodes is distributed identically to Argon2i. Thus, Lemma 41 also applies if we sample $G \sim \mathsf{Argon2iDyn}_N$ and then take the meta-graph G_m of $G \cap [N]$. Now we can easily bound the ancestral robustness of Argon2i.

If we pick $m = \Omega(\sqrt{N \log N})$ then the the meta-graph G_m will be the complete graph on N/m nodes with high probability $1 - o(1/N)$.

Lemma 42 (The Ancestral Robustness of Argon2i). *Fix any constant* $0 < \varepsilon < 1/2$. *Sample* $G' \sim \mathsf{Argon2iDyn}_N$ *and let* $G = G \cap [N]$. *Then, there exist constants* $0 < c, c', f < 1$ *such that, except with probability* $O\left(\exp\left(-\frac{N^{2\varepsilon}}{201}\right)\right)$, G *is* $\left(cN^{1/2-\varepsilon}, c'N^{3/2-\varepsilon}, f\right)$*-ancestrally robust.*

Proof. By Lemma 41 the meta-graph G_m of G will be complete except with probability $\exp\left(-\frac{m^2}{200N} + 2\log N\right)$. If we set $m \geq N^{1/2+\epsilon}$ then $-\frac{m^2}{200N} + 2\log N = -\frac{N^{2\epsilon}}{200} + 2\log N$. For suitably large N, we have $-\frac{N^{2\epsilon}}{200} + 2\log N \leq -\frac{N^{2\varepsilon}}{201}$. Thus, G_m will be complete except with probability $\exp\left(-\frac{N^{2\varepsilon}}{201}\right)$.

Observe that if G_m is the complete graph on $N' = N/m$ nodes then for every set $A \subseteq V(G_m)$ of $|A| \leq N'/4$ nodes we can find a set $B \subseteq V(G_m) \setminus A$ of size $|B| = N'/4$ such that for every node $v \in B$ the graph $H = \mathsf{Ancestors}(v, G_m - A)$ is $(N'/4, N'/4)$-depth robust. In particular, we can take B to be the topologically last $N'/4$ nodes in $G_m - A$ and then for every $v \in B$ the graph $H = \mathsf{Ancestors}(v, G_m - A)$ is a complete graph with at least $N' - |A| - |B| \geq N'/2$ nodes. Thus, H is certainly $(N'/4, N'/4)$-depth robust. Thus, we can simply apply Lemma 12 to conclude that G is $(N/(4m), 3N^2/(5 \cdot 4^2 m), 1/5)$-ancestrally robust. Lemma 42 now follows by plugging in $m = \Theta\left(N^{1/2+\epsilon}\right)$. □

We may now apply Theorem 4 and Theorem 7 to prove the main results of this section.

Proof of Theorem 13. Sample $G' \sim \mathsf{Argon2iDyn}_N$ and let $G = G \cap [N]$. By Lemma 42 the graph G is $\left(cN^{1/2-\varepsilon}, c'N^{3/2-\varepsilon}, f\right)$-ancestrally robust except with probability $O\left(\exp\left(-\frac{N^{2\varepsilon}}{201}\right)\right)$ and by Theorem 15 the graph G is $\left(e, \ell_N \frac{N^3}{e^3}, f\right)$-fractionally depth robust except with probability $o(1/N)$ for some poly-logarithmic term ℓ_N.

Assuming that G is ancestrally robust and fractionally depth robust we can apply Theorem 4 to conclude that for any dynamic pebbling strategy, except with negligible probability $\exp\left(-\Omega(N)\right)$, we will either have $\left|\{i : |P_i| \geq e\}\right| = \Omega(N)$ rounds where the space usage exceeds e, or incur cumulative pebbling cost penalty at least

$$\Omega\left(\min\left\{N^{1/2-\varepsilon} \cdot \ell_N \frac{N^3}{e^3} \cdot N, N^{3/2-\varepsilon}N\right\}\right) = \Omega\left(\min\left\{N^{4.5-\varepsilon-3\beta} \cdot \ell_N, N^{2.5-\varepsilon}\right\}\right).$$

Since $\beta > 2/3$, it follows that $\Omega\left(\min\left\{N^{4.5-\varepsilon-3\beta}, N^{2.5-\varepsilon}\right\}\right) = \Omega\left(N^{4.5-\varepsilon-3\beta}\right)$. Thus, the cumulative complexity penalty is at least $\Omega(\ell_N N^{4.5-\varepsilon-3\beta})$. By union-bounds we can conclude that, except with probability $o(1/N)$, G is both $\left(cN^{1/2-\varepsilon}, c'N^{3/2-\varepsilon}, f\right)$-ancestrally robust and $\left(e, \ell_N \frac{N^3}{e^3}, f\right)$-fractionally depth robust. Similarly, the failure probability for Theorem 4 is at most $\exp\left(-\Omega(N)\right)$. Finally, we can remove the polylog factors from the fractional depth-robustness of Argon2i by replacing ε with some $\varepsilon' < \varepsilon$, which differs by any small constant. □

Proof of Theorem 14. From Theorem 7, we see that the sustained space parameter is $\Omega(we)$ with $e = N^\beta$. The cumulative complexity parameter is (up to at most log factors) $N \min\left\{a'd, C/a\right\}$, where $a' = N^{1/2-\varepsilon}$, $d = N^{3-3\beta}$, and $C = N^{3/2-\varepsilon}$. With these parameters, we do not achieve a good bound. Recall, though, that if G is (a', C, f)-ancestrally robust, then it is also (a, C, f)-ancestrally robust for $a < a'$. Let $a = N^{\alpha-\varepsilon}$ for some $0 < \alpha < 1/2$. Then our cumulative memory complexity penalty (up to log factors of N and factors of w) is $\min\left\{N^{4+\alpha-\varepsilon-3\beta}, N^{2.5-\alpha}\right\}$. Then the best assignment is for $\alpha = \frac{3/2-\varepsilon-3\beta}{2}$. Since $\alpha < 1/2$, this is valid for $\beta \leq 5/6 - \varepsilon/3$. Thus, the CC penalty is $\tilde{\Omega}\left(Nad\right) = \tilde{\Omega}\left(N^{4-3\beta-\alpha-\varepsilon}\right) = \tilde{\Omega}\left(N^{3.25-3\beta/2-\varepsilon/2}\right)$ as long as $0 < \varepsilon, \alpha, \beta < 1$, $\alpha < 1/2 - \varepsilon$, and $\beta \leq 5/6 - \varepsilon/3$. As in the proof of Theorem 13, we can without loss of generality remove the log factors in this bound by adjust ε to be slightly smaller. □

J Comparing the Overhead for DEGSample with [BH22]

DEGSample achieves the best known sustained space/cumulative complexity in the dynamic pebbling model, matching the theoretical construction of [BH22],

which we will call STSample — see Corollary 1. In this section we compare the overhead of both constructions and argue that overhead for STSample is at least $1,000$ times higher.

The primary static subgraph of STSample satisfies an expensive combinatorial property called "ST-Robustness" introduced in [BCLS22]. A DAG G is c-maximally ST-robust if there are disjoint sets of nodes I (inputs) and O (outputs) of size $|I|= |O|= n$ such that for every subset $S \subseteq V(G)$ of $|S|\leq cn$ nodes that we could delete from G the graph $G - S$ contains a subgraph H containing at least $|I \cap V(G)|\geq n - |S|$ input nodes, at least $|O \cap V(G)|\geq n - |S|$ output nodes and for ***every*** pair $s \in I \cap V(H)$ and $t \in O \cap V(H)$ the graph H contains a directed path from s to t. If $c = 1$ then the graph is said to be maximally ST-robust. We note that the $\mathsf{STSample}$ construction of [BH22] relies on a maximally ST-robust graph with $c = 1$. Blocki et al. [BCLS22] provided the only known construction ST_N of a maximally ST-Robust graph with N input nodes and N output nodes, constant indegree and at most $O(N)$ nodes overall. However, the hidden constants are quite large as their construction relies on ***numerous*** copies of combinatorial objects such as superconcentrators[Pip77] and a family $\{H_N\}$ of $(e = cN, d = N^{2/3})$-depth robust graph with constant indegree constructed by [Sch83]. By contrast, DEGSample only uses a single copy of H_N and this is the only component in the construction that is even moderately expensive.

We will compare DEGSample and STSample by lower bounding the number of copies of H_N that are required. The construction of [BCLS22] relies on a gadget M_N which contains three separate copies of H_N — see [BCLS22, Construction 15]. The gadget M_N is proven to be c'-maximally ST-robustness with $c' = 9c/80$ where c is the constant from the underyling $(e = cN, d = N^{2/3})$-depth robust graph H_N. Finally, Blocki et al. [BCLS22] bootstrap from the c'-maximally ST-robust M_N to obtain a maximally robust graph using $\lceil 1/c' \rceil$ copies of M_N — see [BCLS22, Construction 18]. Thus, the total number of copies of H_N is at least $3 \times \left(\frac{80}{9c}\right)$ where the constant c comes from the construction of H_N is [Sch83]. While c is not explicitly computed in [Sch83] we can lowerbound $c = \frac{1}{8f(1/3)} \geq 0.026$ where $f(\epsilon) = \frac{2}{1-\epsilon}\log_2\left(\frac{2}{1-\epsilon}\right)$ — see the last line of the proof of [Sch83, Theorem A] as well as the first Corollary. Thus, [BCLS22, Construction 18] uses ***at least*** $1,025$ copies of H_N and this is likely a significant underestimate!