The Impact of Reversibility on Parallel Pebbling

Abstract. The (parallel) classical black pebbling game is a helpful abstraction which allows us to analyze the resources (time, space, space-time, cumulative space) necessary to evaluate a function f with a static data-dependency graph G on a (parallel) computer. In particular, the parallel black pebbling game has been used as a tool to quantify the (in)security of Data-Independent Memory-Hard Functions (iMHFs). However, the classical black pebbling game is not suitable to analyze the cost of quantum preimage attack. Thus, Blocki et al. [BHL22] introduced the parallel reversible pebbling game as a tool to analyze resource requirements for a quantum computer. While there is an extensive line of work analyzing pebbling complexity in the (parallel) black pebbling game, comparatively little is known about the parallel reversible pebbling game. Our first result is a lower bound of $\Omega\left(N^{1+\sqrt{\frac{2-o(1)}{\log N}}}\right)$ on the reversible cumulative pebbling cost for a line graph on N nodes. This yields a separation between classical and reversible pebbling costs demonstrating that the reversibility constraint can increase cumulative pebbling costs (and space-time costs) by a multiplicative factor of $N^{(\sqrt{2}+o(1))/\sqrt{\log N}}$ — the classical pebbling cost (space-time or cumulative) for a line graph is just $\mathcal{O}(N)$. On the positive side, we prove that *any* classical parallel pebbling can be transformed into a reversible pebbling strategy whilst increasing space-time (resp. cumulative memory) costs by a multiplicative factor of at most $\mathcal{O}\left(N^{\sqrt{\frac{8}{\log N}}}\right)$ (resp. $\mathcal{O}\left(N^{\mathcal{O}(1)/\sqrt[4]{\log N}}\right)$). We also analyze the impact of the reversibility constraint on the cumulative pebbling cost of depth-robust and depth-reducible DAGs exploiting reversibility to improve constant factors in a prior lower bound of Alwen et al. [ABP17]. For depth-reducible DAGs we show that the state-of-the-art recursive pebbling techniques of Alwen et al. [ABP17] can be converted into a recursive reversible pebbling attack without any asymptotic increases in pebbling costs. Finally, we extend a result of Blocki et al. [BLZ20] to show that it is Unique Games hard to approximate the reversible cumulative pebbling cost of a DAG G to within any constant factor.

Keywords: Parallel Reversible Pebbling · Data-Independent Memory-Hard Function · Quantum Preimage Attacks.

1 Introduction

The classical black pebbling game is a powerful computational abstraction that is used to analyze the relationship between the space and time complexity needed to evaluate a function f_G with a static data-dependency graph G. Intuitively, the nodes of the directed acyclic graph (DAG) G represent intermediate data-values generated during the computation of f_G and the edges in G encode static data-dependencies between these intermediate values, e.g., if $z = H(x, y)$ then the DAG G would include edges (x, y) and (y, z) to indicate that we need to have labels x and y in memory before we can compute label z. A pebbling of G is a sequence $P = (P_0, \ldots, P_t) \subseteq V(G)$ of subsets where P_i denotes the subset of nodes that have pebbles on them during round i. Intuitively, P_i represents the intermediate labels that are stored in memory at time i. In the field of cryptography, the *parallel* pebbling game has been used to analyze the security of *Data-Independent Memory-Hard Functions* (iMHFs), e.g., see [AS15, AB16, ABP17, BZ17]. Due to their side-channel resistance, iMHFs are an attractive tool to protect low-entropy secrets such as user passwords against brute-force attacks.

There is a wide-body of literature analyzing the classical pebbling complexity of graphs G under various cost metrics: space complexity [PTC76, HPV77], space-time complexity [LT82, LT79], amortized space-time complexity (or equivalently, cumulative pebbling complexity) [AS15, AB16, ABP17, BZ17, BHK+19, ABH17, BZ18, BLZ20, AGK+18]. Space complexity ($\Pi_s(P) \doteq \max_i |P_i|$) focuses on the memory resources that are *necessary* to perform a computation. By contrast, space-time complexity ($\Pi_{st}(P) \doteq t \cdot \max_i |P_i|$) focuses on the *full cost* of computation, i.e., the amount of space that is locked up multiplied by the running time of the computation. While early work on graph pebbling focused on the sequential black pebbling game [PTC76, HPV77, LT82, LT79], Alwen and Serbinenko [AS15] observed that parallelism and amortization can have a dramatic impact on the space-time complexity of a graph[1]. Because a brute-force attacker can be parallel and can amortize costs over multiple different inputs, it is important to construct iMHFs with high amortized space-time complexity in order to protect low-entropy secrets (e.g., user passwords) against brute-force attacks. Thus, a recent line of crypto research has focused on analyzing the cumulative pebbling complexity ($\Pi_{cc}(P) = \sum_i |P_i|$) of prominent iMHF candidates and constructing graphs with high cumulative pebbling cost, e.g., see [AS15, AB16, ABP17, BZ17, BHK+19, ABH17, BZ18, BLZ20, AGK+18].

Unfortunately, the (classical) parallel pebbling game is insufficient for analyzing the full cost of a *quantum* preimage attack on iMHFs. For example, suppose that we want to recover a preimage of $f_G(x)$ when the input $x \in \{0,1\}^m$ is a random m-bit string. Classically, an attacker would need to make $\Omega(2^m)$ queries to the function f_G to find a preimage, but a quantum adversary could exploit

[1] For example, the sequential space-time complexity of the bit-reversal graph on N nodes was proven to be $\Omega(N^2)$[LT79], but the parallel space-time complexity of this graph is just $\mathcal{O}\left(N\sqrt{N}\right)$. There is a another construction of a constant indegree DAG with N nodes such that (1) any parallel pebbling of G has space-time cost $\Omega\left(N\sqrt{N}\right)$, and (2) one can pebble $\sqrt{N}$ disjoint copies of G with amortized space-time cost at most $\mathcal{O}(N)$, i.e., there is a parallel pebbling $G^{\otimes\sqrt{N}}$ ($\sqrt{N}$ disjoint copies of G) with total space-time cost $\mathcal{O}\left(N\sqrt{N}\right)$.

quantum superposition and and recover the preimage after just $\mathcal{O}\left(2^{m/2}\right)$ quantum queries to f_G using Grover's algorithm — a quadratic reduction. However, in order to run Grover's algorithm, we need to construct a quantum circuit that coherently computes our iMHF f_G. Because quantum computation utilizes reversible unitary operations, an efficient black pebbling of the DAG G does not necessarily correspond to an efficient quantum circuit that evaluates f_G coherently. Thus, Blocki, Holman, and Lee [BHL22] introduced the *parallel reversible pebbling game* as a tool to analyze the (amortized) cost of a quantum circuit evaluating an iMHF coherently.

Intuitively, the parallel reversible pebbling game extends the classical parallel black pebbling game by imposing restrictions on when pebbles can be removed. By contrast, the classical black pebbling game imposes no restrictions on when pebbles can be removed to free up space. One of the primary motivations of the parallel reversible pebbling game is to provide a tool to analyze the full cost of quantum preimage attacks against an iMHF. The full cost of a single quantum preimage attack using Grover's algorithm will be proportional to the space-time cost of G in the parallel reversible pebbling game. If the attacker is running multiple preimage attacks in parallel (e.g., cracking multiple breaches passwords) then the attacker's amortized costs will scale proportional to the amortized space-time complexity of the underlying graph G in the parallel reversible pebbling game. Thus, to protect low-entropy secrets against quantum preimage attacks in the future, it is useful to characterize the parallel reversible space-time complexity of prominent iMHF candidates.

While there has been an extensive body of work analyzing the space-time and cumulative pebbling costs of DAGs in the parallel black pebbling game (e.g., see [AS15, ABP17, ABP18, BHK+19, BZ17]), comparatively little is known about the reversible pebbling game. Blocki et al. [BHL22] gave parallel reversible pebbling strategies for iMHFs such as Argon2 [BDK16] and DRSample [ABH17] which improve upon the naïve reversible pebbling strategy by modest factors of $\sqrt[3]{\log N}$ and $\frac{\log N}{\log\log N}$, respectively. They also showed that the line graph can be pebbled with space-time complexity $\mathcal{O}\left(N \cdot 2^{2\sqrt{\log N}}\right)$, whereas in the classical pebbling game the space-time complexity is simply N. However, prior to this work, there was no non-trivial lower bound on the cumulative pebbling cost of the line graph which would help us to characterize the full cost of a quantum preimage attack against popular password-based key-derivation functions like BCRYPT [PM99] or PBKDF2 [Kal00]. Similarly, our characterization of the cumulative pebbling cost for many prominent iMHF candidates (including Password Hashing Competition [PHC15] winner Argon2i [BDK16]) was far from tight.

If we dropped the reversibility constraint, it is natural to wonder whether or not would we be able to find parallel pebbling attacks with lower costs for graphs such as Argon2i [BDK16], DRSample [ABH17], or the line graph. This leads us to ask the following natural question:

> *Can we characterize the impact of the reversibility constraint on pebbling costs?*

More generally, what is the necessary overhead (in terms of space-time/amortized space-time complexity) to build a quantum circuit for a classical algorithm? If there is such an inherent penalty for reversibility, is there a systematic way to map classical algorithms to quantum circuits that never exceed this penalty? In this paper, we answer both questions in the affirmative in the parallel reversible pebbling model.

1.1 Our Results

In this paper, we are concerned with characterizing the extent to which reversibility impacts pebbling costs. While we are primarily motivated by characterizing the post-quantum security of Memory-Hard Functions, we note that the reversible pebbling game is a general tool to analyze space-time trade-offs of reversible computation. Thus, our results will likely be of interest outside the field of cryptography, e.g., quantum circuit compilation. At a high level, our main results show that

(1) any generic procedure (captured by the parallel reversible pebbling game) for converting a classical algorithm running in time t into an equivalent quantum circuit must increase amortized space-time complexity (cumulative pebbling complexity) by a factor of at least $2^{(\sqrt{2}-o(1))\sqrt{\log t}}$, and

(2) there exists a procedure for converting classical algorithms into quantum circuits that increases amortized space-time complexity by a factor of at most $2^{\mathcal{O}\left(\log^{3/4} t\right)}$.

There is a wealth of analysis of iMHF candidates in the classical parallel black pebbling game, e.g., [AB16, AB17, BZ17, ABP17, ABH17]. The second result immediately transforms all of these classical pebbling attacks into reversible pebbling without significantly increasing the amortized space-time complexity.

1.1.1 A Separation between Reversible and Irreversible Pebbling. Bennett [Ben89] presented the first *sequential* reversible pebbling of the line graph, and it has remained open whether Bennett's original pebbling is optimal [FA17]. Blocki et al. [BHL22] provided slight modifications to Bennett's pebbling to show that the parallel reversible cumulative pebbling complexity of the line graph $\mathcal{L}_N$ on N nodes is *at most* $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N) = \mathcal{O}\left(N \cdot 2^{2\sqrt{\log N}}\right)$ — the notation $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N)$ denotes the minimum cumulative pebbling complexity taken over all legal parallel ($\parallel$) and reversible ($\Longleftrightarrow$) pebblings of $\mathcal{L}_N$. Prior work of Knill [Kni95] showed that for reversible sequential pebbling we have $\Pi_{st}^{\Longleftrightarrow}(L_N) = \Omega\left(N \cdot 2^{2\sqrt{\log N}}\right)$. However, proving lower bounds is substantially harder when we allow for parallel pebbling strategies and when we consider cumulative pebbling costs instead of space-time costs. We show that any parallel reversible pebbling of the line graph has cumulative pebbling complexity $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N) = \Omega\left(N \cdot 2^{(\sqrt{2}-o(1))\sqrt{\log N}}\right)$ (see Theorem 1). This immediately implies that the space-time complexity of any parallel reversible pebbling is at least

$\mathit{\Pi}_{st}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N) = \mathit{\Omega}\left(N \cdot 2^{(\sqrt{2}-o(1))\sqrt{\log N}}\right)$ since the cumulative pebbling cost of a pebbling always upper bounds the space-time cost, i.e., $\mathit{\Pi}_{cc}(P) \geq \mathit{\Pi}_{st}(P)$ for any pebbling P.

This result immediately implies a multiplicative gap between the reversible and irreversible pebbling costs. In particular, there is a classical sequential pebbling of the line graph $\mathcal{L}_N$ which runs in time N and keeps at most $\mathcal{O}(1)$ pebbles on the graph during any round. Thus, for classical pebblings, the sequential space-time cost (and cumulative pebbling cost) is at most $\mathit{\Pi}_{cc}^{\parallel}(\mathcal{L}_N) = \mathit{\Pi}_{st}^{\parallel}(\mathcal{L}_N) = \mathcal{O}(N)$ for the line graph $\mathcal{L}_N$. It follows that

$$\frac{\mathit{\Pi}_{st}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N)}{\mathit{\Pi}_{st}^{\parallel}(\mathcal{L}_N)} = \mathit{\Omega}\left(2^{(\sqrt{2}-o(1))\sqrt{\log N}}\right), \text{ and } \frac{\mathit{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N)}{\mathit{\Pi}_{cc}^{\parallel}(\mathcal{L}_N)} = \mathit{\Omega}\left(2^{(\sqrt{2}-o(1))\sqrt{\log N}}\right).$$

Our results also show that the attack of Blocki et al. [BHL22] is optimal within the subpolynomial factor of $N^{\frac{\sqrt{2}}{\sqrt{\log N}}} = 2^{\sqrt{2\log N}}$. Our lower bounds also have implications about the full cost of quantum password cracking attacks when the BCRYPT [PM99] or PBKDF2 [Kal00] hash function was used. See Section 3.1 for details.

1.1.2 Pebbling Attacks: Making Computation Reversible.

In light of the previous result, it is natural to wonder if we can find a family of graphs G_N with a larger multiplicative gap between the reversible/classical pebbling costs than the line graph $\mathcal{L}_N$, specifically with respect to the stronger metric of cumulative pebbling complexity. In the sequential computation setting, Bennett [Ben89] showed how to transform an irreversible pebbling into a reversible pebbling while preserving *space-time complexity*. We demonstrate that this transformation can be extended to the parallel setting. More specifically, we show that an irreversible parallel pebbling $P = (P_0, \ldots, P_t)$ of G can made reversible using a reversible line graph pebbling $Q = (Q_0, \ldots, Q_{t'})$ of the line graph $\mathcal{L}_t$. In particular, we argue that the composed pebbling $R = (R_0, \ldots, R_{t'})$ with $R_i = \bigcup_{j\in Q_i} P_j$ for each $i \leq t'$ is a legal reversible pebbling of G. Trivially, we have $\max_i |R_i| \leq (\max_i |P_i|)(\max_j |Q_j|)$, i.e., the maximum space usage for our reversible pebbling is the product of the maximum space usage of P and Q. We can use the reversible line graph pebbling from [BHL22] to instantiate our pebbling $Q = (Q_0, \ldots, Q_{t'})$ and show that the irreversible space-time can never be too far from reversible space-time complexity.

Theorem 2 (Classical vs. Reversible Space-Time Complexity). *Let $G = (V = [N], E)$ be a DAG. Then*

$$\mathit{\Pi}_{st}^{\Longleftrightarrow,\parallel}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right) \cdot \mathit{\Pi}_{st}^{\parallel}(G),$$

and

$$\mathit{\Pi}_{st}^{\Longleftrightarrow}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\sqrt{\log N}\right) \cdot \mathit{\Pi}_{st}(G).$$

Unfortunately, the above strategy (generalized from Bennett [Ben89]) completely fails to preserve cumulative memory costs. Suppose for example that the pebbling $P = (P_1, \dots, P_t)$ of G has low $\mathit{\Pi}_{cc}(P) = \sum_i |P_i|$. It is possible that there is some round i where the space usage $|P_i| \gg \mathit{\Pi}_{cc}(P)/t$ greatly exceeds the average space usage per round. Observe that for our composed pebbling, we will have $|R_j| \geq |P_i|$ for every round $j \leq t'$ such that $i \in Q_j$. If we get unlucky it could be that the reversible line graph pebbling $Q = (Q_1, \dots, Q_{t'})$ of $\mathcal{L}_t$ keeps a pebble on node i in almost every round j so that $\mathit{\Pi}_{cc}(R) \gg \mathit{\Pi}_{cc}(P)$. We address this problem by introducing a *weighted version* of the reversible pebbling game where the cost of placing a pebble on a node i is equal to its weight. Intuitively, we will set the weight of node i in $\mathcal{L}_t$ to be $|P_i|$. We then design efficient reversible pebbling strategies for the weighted line graph to compose with such irreversible pebblings. If we take $Q = (Q_1, \dots, Q_{t'})$ to be our CC-efficient, reversible weighted line graph pebbling then we can compose this reversible pebbling with $P = (P_1, \dots, P_t)$ to obtain a composed pebbling $R = (R_1, \dots, R_{t'})$ such that $\mathit{\Pi}_{cc}(R) \leq \mathit{\Pi}_{cc}(P) \cdot \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right)$. We stress that this is the primary technical challenge as adding weights to the nodes makes it substantially more challenging to develop efficient reversible pebbling strategies.

Theorem 3 (Classical vs. Reversible Cumulative Pebbling Complexity). *Let $G = (V = [N], E)$ be a DAG. Then*

$$\mathit{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \mathit{\Pi}_{cc}^{\parallel}(G),$$

and

$$\mathit{\Pi}_{cc}^{\Longleftrightarrow}(G) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \mathit{\Pi}_{cc}(G).$$

This means that to find an efficient reversible pebbling (up to these subpolynomial factors), it suffices to find an efficient classical pebbling. See Section 3.2 for details.

1.1.3 Reversibility and Depth-Robust Graphs. Classically, an important property of pebbling a graph is *depth robustness*. A DAG $G = (V, E)$ is (e, d)-depth robust if for any subset $S \subseteq V$ of e nodes, the graph $G - S$ still contains a directed path of length d (if G is not (e, d)-depth robust then we say that G is (e, d)-reducible). Alwen et al. [ABP17] showed that if G is (e, d)-depth robust then any classical parallel pebbling of G has cumulative pebbling cost at least $\mathit{\Pi}_{cc}^{\parallel}(G) \geq ed$. While the same lower bound holds for the parallel *reversible* CC, it is natural to ask if one could achieve a better lower bound. We show that if G is (e, d)-depth robust then any parallel reversible pebbling of G has cumulative pebbling cost at least $\mathit{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G) \geq e(2d-1)$, and furthermore if $G - \mathsf{sinks}(G)$ (where $\mathsf{sinks}(G)$ denotes the set of sink nodes of G) is (e, d)-depth robust then $\mathit{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G) \geq 2ed$ (see Theorem 9). Intuitively, the lower bound of Alwen et al. [ABP17] followed from the observation that given a pebbling $P = (P_0, \dots, P_t)$

of G such that for any $1 \leq i \leq d$, the set $B_i = P_i \cup P_{i+d} \cup P_{i+2d} \ldots$ is a depth-reducing set, i.e., $G - B_i$ contains no path of length d. Intuitively, if $G - B_i$ had a path $v_1, \ldots, v_d$ of length d then we would *never* place a pebble on node v_d (It takes d steps to walk a pebble down the path, but every d rounds we are guaranteed to have no pebbles on the path). Our key observation is that for a reversible pebbling it would take at least $2d$ rounds to walk a pebble down to node v_d and then remove pebbles from every node in the path. Thus, we can increase our gap to $2d$, define $B_i = P_i \cup P_{i+2d} \cup P_{i+4d} \cup P_{i+6d} \ldots$, and argue that $G - \mathsf{sinks}(G) - B_i$ contains no path of length d.[2]

We also consider a parallel *relaxed* reversible pebbling where it is not required to remove pebbles from the intermediate nodes at the final round. In this setting, we cannot apply our new lower bound directly since we cannot assume that all pebbles on non-sink nodes are cleared by the end of the pebbling, e.g., it is possible during the last d pebbling rounds we pebble all of the nodes in the path $v_1, \ldots, v_d$ and leave them. To lower bound the cost of a parallel relaxed reversible pebbling, it is helpful to define a graph $G_{\mathsf{Trunc},d} \coloneqq G - [N-d+1, N]$ where we truncate last d nodes and incident edges from the graph G. We show that if $G_{\mathsf{Trunc},d}$ is (e,d)-depth robust then $\widetilde{\Pi}_{cc}^{\leftrightarrow,\parallel}(G) \geq e(2d-1)$ (see Theorem 10), where $\widetilde{\Pi}_{cc}^{\leftrightarrow,\parallel}(G)$ denotes the *parallel relaxed reversible CC* of G (see Definition 2 for a formal definition). This yields improvement by a multiplicative factor of ≈ 1.885 for the parallel relaxed reversible CC of DRSample [ABH17] with suitable parameters. See Section 5 for details.

1.1.4 Reversible Recursive Pebbling Attack.

Alwen and Blocki [AB16] gave a generic parallel pebbling attack on any (e,d)-reducible graph G with $\Pi_{cc}^{\parallel}(G) \leq \mathcal{O}\left(eN + N\sqrt{Nd}\right)$. While Blocki et al. [BHL22] gave a reversible version of the attacks from Alwen et al. [AB16], the state-of-the-art upper bounds on $\Pi_{cc}^{\parallel}(G)$ for most depth-reducible graphs actually utilize the recursive depth-reducing attack of [ABP17] — a recursive extension of [AB16] for graphs that are (e_i, d_i)-reducible for a set of points $(e_0, d_0), (e_1, d_1), \ldots$ with decreasing depth parameters $d_{i+1} < d_i$ and increasing size parameters $e_i > e_{i-1}$. We provide a reversible extension of the recursive depth-reducing attack of [ABP17]. As an immediate corollary, we obtain upper bounds on $\Pi_{cc}^{\leftrightarrow,\parallel}(G)$ which (asymptotically) match the best known classical pebbling upper bounds on $\Pi_{cc}^{\parallel}(G)$ for several iMHF candidates including Argon2iA (an older version of Argon2i) and Argon2iB (the current version). See Section 4 for details.

1.1.5 Approximation Hardness of the Parallel Reversible Cumulative Pebbling Cost.

We establish the approximation hardness of $\Pi_{cc}^{\leftrightarrow,\parallel}(G)$ for a constant-indegree DAG G within any constant factor in the worst-case analysis under the Unique Games Conjecture. Our result extends the prior approxima-

[2] We have to exclude $\mathsf{sinks}(G)$ because if the final node v_d in our path was a sink node then the pebbling may never remove a pebble from node d. In this case, it may be possible to walk a pebble to node v_d and then remove pebbles from $v_1, \ldots, v_{d-1}$ in just $2d-1$ steps.

tion hardness result by Blocki et al. [BLZ20] which demonstrates that given a constant-indegree DAG G, it is Unique Games hard to approximate $\Pi_{cc}^{\parallel}(G)$ within any constant factor. The reduction of [BLZ20] transformed a unique games instance[3] G into a new graph $\mathsf{superconc}(G)$ by overlaying G with a special combinatorial graph called a superconcentrator. If the Unique Games instance G was sufficiently depth-robust then it is possible to lower-bound the pebbling cost $\Pi_{cc}^{\parallel}(\mathsf{superconc}(G))$ — since $\Pi_{cc}^{\Leftrightarrow,\parallel}(\mathsf{superconc}(G)) \geq \Pi_{cc}^{\parallel}(\mathsf{superconc}(G))$, the lower bound immediately extends to the reversible setting. Similarly, if the Unique Games instance G was sufficiently depth-reducible then one could find a low-cost pebbling of $\mathsf{superconc}(G)$. Our primary contribution is showing how to modify the pebbling strategy of Blocki et al. [BLZ20] to obtain a reversible pebbling of $\mathsf{superconc}(G)$ without asymptotically increasing the pebbling cost. The approximation hardness of $\Pi_{cc}^{\Leftrightarrow,\parallel}(G)$ immediately follows. See Appendix B for details.

1.2 Related Work

Reversible pebbling games [Ben89, Krá01, MSR+19, Kni95] were introduced to analyze the space-time complexity of quantum algorithms in the context of the limitations imposed by reversibility and the Quantum No-Deletion Theorem. These pebbling games only model *sequential* computation, meaning only one pebble can be placed or removed each round. In contrast, quantum adversaries computing an MHF $f_{G,H}$ can make quantum queries to H in parallel, making these sequential games insufficient for analyzing the security of MHFs. For this reason, Blocki et al. [BHL22] introduced the *parallel reversible pebbling game*, which extends the reversible pebbling game by allowing any number of legal placing and removing of pebbles in each round. The authors used the parallel reversible pebbling game to analyze the post-quantum security of iMHFs to provide reversible space-time cost upper bounds of $\mathcal{O}\left(N \cdot 2^{2\sqrt{\log N}}\right)$ for line graphs and the first reversible space-time cost upper bound of $\mathcal{O}\left(\frac{N^2 \log\log N}{\log N}\right)$ for Argon2i. They also designed a reversible depth-reducing attack with cumulative pebbling complexity asymptotically equivalent to its counterpart in [ABP17].

Kornerup et al. [KSS21] introduced the (sequential) *spooky pebbling game* which models measurement-based deletion. The goal of the spooky pebbling game is to save quantum memory by measuring, storing the result of the measurement in classical memory, and then later using the result to restore the original state. A disadvantage to the spooky pebbling game in the context of a preimage attack is that it requires a linear number of measurements for each query to $f_{G,H}$, making it unsuitable for our applications [BHL22, KSS21].

[3] The problem is to distinguish between the case that G is "highly depth-robust" or "highly depth-reducible."

2 Preliminaries and Definitions

2.1 Notation

For a positive integer N, we denote $[N] \coloneqq \{1, \ldots, N\}$. Similarly, for positive integers $a \leq b$, we define $[a, b] \coloneqq \{a, \ldots, b\}$. For simplicity, we let $\log(\cdot)$ be a log with base 2, i.e., $\log x \coloneqq \log_2 x$. The notation $\xleftarrow{\$}$ denotes a uniformly random sampling, e.g., we say $x \xleftarrow{\$} [N]$ when x is sampled uniformly at random from 1 to N.

Let $G = (V, E)$ be a directed acyclic graph (DAG) with the set of nodes V and the set of edges E. Without loss of generality, we often times simply let $V = [N]$ where N is the number of nodes in G. Throughout the paper, we will follow this notation convention (that $V = [N]$) unless specified differently. For $v \in V$, we define $\mathsf{parents}(v, G)$ to be the *immediate parents* of node v in G, i.e., $\mathsf{parents}(v, G) \coloneqq \{u \in V : (u, v) \in E\}$. Similarly, for a subset $W \subseteq V$, we say $\mathsf{parents}(W, G) \coloneqq \bigcup_{w \in W} \{u : (u, w) \in E\}$ to be the *immediate parents* of the set W in G. We define $\mathsf{ancestors}(v, G)$ to be the set of all *ancestors* of v in G, i.e., $\mathsf{ancestors}(v, G) \coloneqq \bigcup_{i \geq 1} \mathsf{parents}^i(v, G)$, where $\mathsf{parents}^1(v, G) = \mathsf{parents}(v, G)$ and $\mathsf{parents}^i(v, G) = \mathsf{parents}(\mathsf{parents}^{i-1}(v, G), G)$. Similarly, $\mathsf{ancestors}(W, G) \coloneqq \bigcup_{i \geq 1} \mathsf{parents}^i(W, G)$, where $\mathsf{parents}^1(W, G) = \mathsf{parents}(W, G)$ and recursively define $\mathsf{parents}^i(W, G) = \mathsf{parents}(\mathsf{parents}^{i-1}(W, G), G)$. We say $\mathsf{sinks}(G) \coloneqq \{v \in V : \nexists (v, u) \in E\}$ to be the set of all *sink nodes* of G. For $v \in V$, $\mathsf{depth}(v, G)$ denotes the number of nodes in the longest directed path in G ending at node v, and $\mathsf{depth}(G) = \max_{v \in V} \mathsf{depth}(v, G)$ denotes the number of nodes in the longest directed path in G. The indegree of a node $v \in V$ is the number of incoming edges into v, i.e., $\mathsf{indeg}(v, G) \coloneqq |\mathsf{parents}(v, G)|$, and the maximum indegree in G is defined by $\mathsf{indeg}(G) \coloneqq \max_{v \in V} \mathsf{indeg}(v)$. For a subset $S \subseteq V$, we define $G - S$ to be the subgraph of G obtained by deleting all the nodes in S and all edges that are incident to S. For $k \in [N]$, $G_{\leq k} \coloneqq G - [k+1, N]$, $S_{\leq k} \coloneqq S \cap [k]$, and $S_{\geq k} \coloneqq S \setminus [k-1]$ for $k \geq 2$ (if $k = 1$ then $S_{\geq k} = S$). For sets S and R, we let $S \oplus R = (S \setminus R) \cup (R \setminus S)$.

We say that a DAG $G = (V, E)$ is (e, d)-*depth robust* if for any subset $S \subseteq V$ such that $|S| \leq e$ we have $\mathsf{depth}(G - S) \geq d$. Otherwise, we say that G is (e, d)-*reducible* and call the subset S a *depth-reducing set* (which is of size at most e and yields $\mathsf{depth}(G - S) < d$).

2.2 Reversible Pebbling Game

Blocki et al. [BHL22] gave a definition of the *parallel reversible pebbling game*. Their definition was somewhat complicated extending the rules for classical pebbling with several new rules to capture constraints imposed by reversible pebbling. We provide a simpler, more intuitive, definition that is equivalent to [BHL22]. Recall that a classical pebbling sequence $(P_j, \ldots, P_k)$ is legal for a directed graph G if we have $\mathsf{parents}(P_{i+1} \setminus P_i, G) \subseteq P_i$ for all $j \leq i < k$ i.e., if we place a new pebble on node $v \in P_{i+1} \setminus P_i$ during round $i + 1$ then all of v's parents must have been pebbled during round i. Arguably, one should impose an additional constraint that $\mathsf{parents}(P_{i+1} \setminus P_i) \subseteq P_{i+1}$ i.e., if we place a new pebble on node $v \in P_{i+1} \setminus P_i$ during round $i+1$ and $u \in \mathsf{parents}(v, G)$ is a parent of node v then we cannot remove the pebble on node u until after round $i+1$. While the literature on parallel black pebbling games does not include this additional restriction it is still natural, and we will call a classical black pebbling sequence $(P_j, \ldots, P_k)$ *extra legal* if it satisfies this additional constraint[4]. Now we can simply say that $(P_j, \ldots, P_k)$ is a legal reversible pebbling sequence if and only if the sequence $(P_j, \ldots, P_k)$ and its reverse $(P_k, \ldots, P_j)$ are both *extra legal*. The formal definition of reversible graph pebbling is presented below.

Definition 1 ((Parellel) Reversible Graph Pebbling). *Let $G = (V, E)$ be a DAG and let $T \subseteq V$ be a target set of nodes to be pebbled. We say that a pebbling sequence $(P_j, \ldots, P_k)$ is called* extra legal *if it satisfies the following properties:*

- *A pebble can be added only if all of its parents were pebbled at the end of the previous pebbling round, i.e., $\forall i \in [j, k) : \mathsf{parents}(P_{i+1} \setminus P_i, G) \subseteq P_i$.*
- *If a pebble was required to generate new pebbles, then we must keep the corresponding pebble around, i.e., $\forall i \in [j, k) : \mathsf{parents}(P_{i+1} \setminus P_i, G) \subseteq P_{i+1}$.*

We say that $(P_j, \ldots, P_k)$ is a reversible pebbling sequence if both $(P_j, \ldots, P_k)$ and the reversed sequence $(P_k, \ldots, P_j)$ are extra legal. A legal parallel reversible pebbling *of a graph G with a target set T is a reversible sequence $P = (P_0, \ldots, P_t)$ such that:*

(1) (Start/Finish) $P_0 = \emptyset$ and $T \subseteq P_t$, i.e., the pebbling should start with no pebbles and end with pebbles on all of the target nodes.
(2) (Reversible) Both P and its reverse $P^* := (P_t, \ldots, P_0)$ are extra legal.
(3) (Remove Excess Pebbles (Optional)) $P_t = T$.

If a reversible pebbling sequence $(P_0, \ldots, P_t)$ satisfies conditions (1) and (2), but does not satisfy condition (3), then we call our pebbling a relaxed *reversible pebbling of G with target set T. The pebbling sequence is* sequential *if it additionally satisfies (4) below, i.e., if at most one pebble is added or removed in each round.*

(4) (Sequential pebbling only) *At most one pebble is added or removed in each round, i.e., $\forall i \in [t] : |(P_i \cup P_{i-1}) \setminus (P_i \cap P_{i-1})| \leq 1$.*

We use $\mathcal{P}^{\Leftrightarrow,\parallel}_{G,T}$ (resp. $\mathcal{P}^{\Leftrightarrow}_{G,T}$) to denote the set of all legal (resp. all legal sequential) reversible pebblings of G with a target set T i.e., all pebbling sequences that satisfy conditions (1), (2) and (3) (resp. conditions (1), (2), (3) and (4)). We denote with $\widetilde{\mathcal{P}}^{\Leftrightarrow,\parallel}_{G,T}$ (resp. $\widetilde{\mathcal{P}}^{\Leftrightarrow}_{G,T}$) the set of all legal relaxed *reversible pebblings of G with target set T satsifying conditions (1) and (2) (resp. conditions*

[4] It is worth noting that adding this constraint does not significantly impact pebbling complexity in the classical parallel black pebbling game. In particular, if $(P_j, \ldots, P_k)$ is a legal pebbling sequence then we can define an extra legal sequence $(E_j, \ldots, E_k)$ as follows: $E_j = P_j$ and $E_i = P_i \cup P_{i-1}$ for all $j < i \leq k$.

(1), (2) and (4)). We will mostly be interested in the case where $T = \mathsf{sinks}(G)$ *in which case we simply write* $\mathcal{P}_G^{\Longleftrightarrow,\parallel}$ *or* $\widetilde{\mathcal{P}}_G^{\Longleftrightarrow,\parallel}$ *(or* $\mathcal{P}_G^{\Longleftrightarrow}$ / $\widetilde{\mathcal{P}}_G^{\Longleftrightarrow}$ *for the sequential counterparts).*

We will often write $P = (P_1, \ldots, P_t)$ instead of $P = (P_0, P_1, \ldots, P_t)$ for a legal parallel reversible pebbling of G since pebbling rules dictate that $P_0 = \emptyset$. In Appendix A, we provide the original definition of Blocki et al. [BHL22] and prove that our simpler definition is equivalent.

We also recall basic notions of reversible pebbling complexity.

Definition 2 (Reversible Pebbling Complexity). *Given a DAG* $G = (V, E)$, *we essentially use the same definitions for the reversible pebbling complexity as defined in the previous literature [AS15, ABP17, ABP18, BHL22]. That is, the standard notion of* time, space, space-time *and* cumulative pebbling complexity (CC) *of a* reversible *pebbling* $P = \{P_0, \ldots, P_t\} \in \mathcal{P}_G^{\Longleftrightarrow,\parallel}$ *are also defined to be:*

- *(time complexity)* $\Pi_t(P) = t$,
- *(space complexity)* $\Pi_s(P) = \max_{i\in[t]} |P_i|$,
- *(space-time complexity)* $\Pi_{st}(P) = \Pi_t(P) \cdot \Pi_s(P)$, *and*
- *(cumulative pebbling complexity)* $\Pi_{cc}(P) = \sum_{i\in[t]} |P_i|$.

For $\alpha \in \{s, t, st, cc\}$ *and a target set* $T \subseteq V$, *the* (non-relaxed/relaxed) parallel *reversible pebbling complexities of* G *are defined as*

$$\Pi_\alpha^{\Longleftrightarrow,\parallel}(G, T) = \min_{P\in\mathcal{P}_{G,T}^{\Longleftrightarrow,\parallel}} \Pi_\alpha(P), \text{ and } \widetilde{\Pi}_\alpha^{\Longleftrightarrow,\parallel}(G, T) = \min_{P\in\widetilde{\mathcal{P}}_{G,T}^{\Longleftrightarrow,\parallel}} \Pi_\alpha(P),$$

respectively. When $T = \mathsf{sinks}(G)$ *we simplify notation and write* $\Pi_\alpha^{\Longleftrightarrow,\parallel}(G)$.

We define the time, space, space-time *and* cumulative pebbling complexity *of a* sequential *reversible pebbling* $P = \{P_0, \ldots, P_t\} \in \mathcal{P}_G^{\Longleftrightarrow}$ *in a similar manner:* $\Pi_t^{\Longleftrightarrow}(P) = t$, $\Pi_s^{\Longleftrightarrow}(P) = \max_{i\in[t]} |P_i|$, $\Pi_{st}^{\Longleftrightarrow}(P) = \Pi_t^{\Longleftrightarrow}(P) \cdot \Pi_s^{\Longleftrightarrow}(P)$, *and* $\Pi_{cc}^{\Longleftrightarrow}(P) = \sum_{i\in[t]} |P_i|$. *Similarly, for* $\alpha \in \{s, t, st, cc\}$ *and a target set* $T \subseteq V$, *the* sequential *reversible pebbling complexities of* G *are defined as* $\Pi_\alpha^{\Longleftrightarrow}(G, T) = \min_{P\in\mathcal{P}_{G,T}^{\Longleftrightarrow}} \Pi_\alpha^{\Longleftrightarrow}(P)$. *When* $T = \mathsf{sinks}(G)$ *we simplify notation as well and write* $\Pi_\alpha^{\Longleftrightarrow}(G)$.

We also introduce a new complexity notion that will be useful in our efficient pebbling compositions. The toggle number of a node v in a pebbling P is the number of times it is pebbled or unpebbled. The toggle number of a pebbling is its maximum toggle number over all nodes.

Definition 3 (Toggle Number). *Let* P *be a pebbling for a DAG* $G = (V = [N], E)$ *and* $v \in V$. *We let* $\mathsf{toggle}(P, v) \coloneqq |\{i \mid v \in P_i \oplus P_{i+1}\}|$, *and* $\mathsf{toggle}(P) \coloneqq \max_{v\in[N]} \mathsf{toggle}(P, v)$.

As mentioned in the prior work [BHL22], when we compare the relaxed and non-relaxed pebbling of a DAG G, the space-time cost and the cumulative pebbling complexity of a relaxed/non-relaxed reversible pebbling is not fundamentally different. We note that compared to the relaxed reversible pebbling, the running time of a non-relaxed pebbling increases by a multiplicative factor of 2 and the space usage increases by an additive factor of $|T| \leq |P_t|$ where T is the target set. Hence, the overall space-time costs increase by a multiplicative factor of 4 *at most* [BHL22] and so is the cumulative pebbling complexity since CC is always upper bounded by the space-time cost. In the remainder of the paper, when we write "legal reversible pebbling" we assume that the pebbling is parallel and non-relaxed by default.

3 The Cost of Reversibility on Pebbling

In this section, we discuss the extent to which the additional rules imposed by reversibility impact the space-time and cumulative pebbling complexity of pebbling graphs. We first show that any reversible pebbling for the line graph $\mathcal{L}_N$ on N nodes has CC $\Omega\left(N^{1+\frac{\sqrt{2}-o(1)}{\sqrt{\log N}}}\right)$. Since cumulative pebbling complexity lower bounds space-time complexity, this also implies that the reversible space-time complexity of the line graph is $\Omega\left(N^{1+\frac{\sqrt{2}-o(1)}{\sqrt{\log N}}}\right)$. Since the classical space-time and cumulative pebbling complexity of the line graph is $\mathcal{O}\left(N\right)$, this result shows that, in general, we cannot hope to provide reversible pebblings with cost equivalent to the best classical pebblings. On the other hand, we also show that any sequential pebbling for a graph G can be converted to into a reversible pebbling for G with a space-time overhead of $\mathcal{O}\left(N^{\frac{2}{\sqrt{\log N}}}\right)$ and a CC overhead of $N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}$.

3.1 A Separation between General and Reversible Pebbling

In this section, we show that line graphs are witnesses to among the greatest asymptotic separations between general and reversible pebblings. In particular, Theorem 1 shows that, in terms of cumulative pebbling complexity, the pebbling in Theorem 5 is tight and the composition in Theorem 2 is tight up to a factor of $N^{\frac{\sqrt{2}}{\sqrt{\log N}}}$.

Theorem 1 (Line Graphs Cumulative Pebbling Complexity Lower Bound). *The cumulative pebbling complexity of the line graph $\mathcal{L}_N$ on N nodes is*

$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N) = \Omega\left(N^{1+\frac{\sqrt{2}-o(1)}{\sqrt{\log N}}}\right).$$

The idea of the lower bound for reversibly pebbling line graphs is as follows. Let $C(N) = \Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathcal{L}_N)$. Any pebbling for $\mathcal{L}_N$ first pebbles the sub-line graph $\mathcal{L}_{k(N)}$ for some increasing function $k(N) \leq N$, incurring cost $C(k(N))$. Now, to pebble the rest of $\mathcal{L}_N$ (incurring cost at least $C(N-k(N))$), the pebbling must at some point either unpebble $[k(N)]$ (with cost $C(k(N))$) or must keep a pebble on $[k(N)]$ (with cost at least $N-k(N)$, the time required to finish pebbling $\mathcal{L}_N$). This leads to Lemma 1.

Lemma 1. *Let $C(N) = \Pi_{cc}^{\leftrightarrow,\|}(\mathcal{L}_N)$. Then for any $1 < k(N) < N$ we have*

$$C(N) \geq C\left(k(N)\right) + C\left(N - k(N)\right) + \min\left\{C\left(k(N)\right), N - k(N)\right\}.$$

We will choose $k(N)$ such that $C(k(N)) \leq N - k(N)$, meaning we only need to bound $C(N) \geq 2C\left(k(N)\right) + C\left(N - k(N)\right)$. Using this relation, we show that $C(N) = \Omega\left(N^{1+\frac{\sqrt{2}-o(n)}{\sqrt{\log N}}}\right)$. We choose $k(N) = N \cdot 2^{-c\sqrt{\log N}} = N^{1-\frac{c}{\sqrt{\log N}}}$ for any $0 < c < \sqrt{2}$ and let $f(N) = N \cdot 2^{c\sqrt{\log N}} = N^{1+\frac{c}{\sqrt{\log N}}}$. By induction, we show that $C(N) \geq c' f(N)$ for some constant $c' > 0$. To prove this, we first show that $2f\left(k(N)\right) + f\left(N - k(N)\right) \geq f(N)$ for all sufficiently large N. The proof is left to Appendix C.

Lemma 2. *Define functions h, f, and g such that for any $0 < c < \sqrt{2}$, $h(N) = 2^{c\sqrt{\log N}}$, $f(N) = N \cdot h(N)$, and $g(N) = 2f\left(\frac{N}{h(N)}\right) + f\left(N - \frac{N}{h(N)}\right)$. There exists $N_0 \geq 1$ such that $f(N) \leq g(N)$ for all $N \geq N_0$.*

Putting it all together, we lower bound the reversible cumulative pebbling complexity of line graphs.

Proof of Theorem 1. Let $C(N) = \Pi_{cc}^{\leftrightarrow,\|}(\mathcal{L}_N)$. Define h, f, and g as in Lemma 2 (for any constant $0 < c < \sqrt{2}$, setting $k(N) = N/h(N)$. Then by Lemma 1, we have that $C(N) \geq C\left(k(N)\right) + C\left(N - k(N)\right) + \min\left\{C\left(k(N)\right), N - k(N)\right\}$. We will prove that $C(N) = \Omega\left(f(N)\right)$ via induction. Define f and g as in Lemma 2. Fix N_0 large enough for (1) Lemma 2 to hold, and (2) $f\left(N/h(N)\right) \leq N - N/h(N)$ for all $N \geq N_0$.

Now pick a sufficiently small constant $c' > 0$ so that $C(N_0) \geq cf(N_0)$. And suppose for all $N_0 \leq N' < N$, that $C(N') \geq cf(N')$. We have

$$\begin{aligned}
C(N) &\geq C\left(k(N)\right) + C\left(N - k(N)\right) && \triangleleft \text{ by Lemma 1} \\
&\qquad + \min\left\{C\left(k(N)\right), N - k(N)\right\} \\
&= 2 \cdot C\left(k(N)\right) + C\left(N - k(N)\right) \\
&\geq 2c' f(k(N)) + c' f(N - k(N)) && \triangleleft \text{ inductive hypothesis} \\
&= c' g(N) \\
&\geq c' f(N). && \triangleleft \text{ by Lemma 2}
\end{aligned}$$

Since this holds for every $0 < c < \sqrt{2}$, it follows that $C(N) = \Omega\left(N^{1+\frac{\sqrt{2}-o(1)}{\sqrt{\log N}}}\right)$. □

Discussion. The data-dependency graph for both the BCRYPT [PM99] and PBKDF2 [Kal00] key-derivation functions is a line graph. Thus, understanding the reversible pebbling complexity of the line graph helps us to characterize the (amortized) cost of cracking passwords hashed with BCRYPT or PBKDF2. Our results provide the first lower bound on the amortized space-time cost of BCRYPT [PM99] and PBKDF2 [Kal00] or any other password hash function that uses hash iteration. While neither BCRYPT or PBKDF2 is memory-hard, the PBKDF2 [Kal00] is still approved by NIST [GNP+17] for password hashing and both password hash functions have been widely deployed — billions of leaked password hashes utilized BCRYPT or PBKDF2.

3.2 Efficient Transformations from Classical to Reversible Pebblings

In this section, we discuss the extent to which it is possible to "convert" parallel irreversible pebblings into parallel reversible pebblings while minimizing the overhead in terms of space-time and cumulative pebbling complexity. The main idea is to consider an irreversible pebbling $P = (P_1, \ldots, P_t)$ of some graph G. Since P is irreversible, it is possible that in some transition $P_i \to P_{i+1}$, some node j was deleted without having its parents pebbled or placed while deleting one of its parents. So, we can simulate $P_i \to P_{i+1}$ by keeping around any pebbles that make this step irreversible. Now suppose our pebbling state contains $P_i \cup P_{i+1} \cup P_{i+2}$. Then we can free up space by removing all pebbles in $P_{i+1} \setminus (P_i \cup P_{i+2})$. This is reversible because $\mathsf{parents}(P_i \setminus P_{i+1}, G)$ and $\mathsf{parents}(P_{i+1} \setminus P_i, G)$ are contained in P_i by the (irreversible) legality of the pebbling P. More generally, we can instead focus on reversibly pebbling the line graph $\mathcal{L}_t$, where each node $i \in [t]$ of $\mathcal{L}_t$ represents the pebbling configuration P_i. By the reversibility of the pebbling of $\mathcal{L}_t$, the resulting pebbling steps of the graph G will be reversible. This is the intuition behind *pebbling composition.*

Definition 4 (Pebbling Composition). *Let $P = (P_1, \ldots, P_t)$ be a pebbling for a graph G and and $L = (L_1, \ldots, L_{t'})$ be a pebbling of the line graph $\mathcal{L}_t$. The composition of L with P is the pebbling $Q = L \circ P$, defined by $Q_i \coloneqq \bigcup_{j \in L_i} P_j$ for $i \in [t']$.*

Using pebbling composition, we show that classical and reversible space-time and cumulative pebbling complexity of graphs are within subpolynomial factors in N of each other.

Theorem 2 (Classical vs. Reversible Space-Time Complexity). *Let $G = (V = [N], E)$ be a DAG. Then*

$$\Pi_{st}^{\longleftrightarrow,\parallel}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right) \cdot \Pi_{st}^{\parallel}(G),$$

and

$$\Pi_{st}^{\longleftrightarrow}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}} \sqrt{\log N}\right) \cdot \Pi_{st}(G).$$

As a brief application of this result, we obtain a new upper bound on the parallel reversible space-time complexity of the bit-reversal graph, underlying MHFs such as Catena [FLW13] — a finalist in the Password Hashing Competition [PHC15]. Alwen and Serbinenko [AS15] show that the parallel space-time complexity of the bit-reversal graph is $\mathcal{O}\left(N^{1.5}\right)$. Applying Theorem 2, we see that the parallel reversible space-time complexity of the bit-reversal graph is $\mathcal{O}\left(N^{1.5+\frac{2\sqrt{2}}{\sqrt{\log N}}}\right)$. See Section 3.2.1 for the proof of Theorem 2.

Theorem 3 (Classical vs. Reversible Cumulative Pebbling Complexity). *Let $G = (V = [N], E)$ be a DAG. Then*

$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \Pi_{cc}^{\parallel}(G),$$

and

$$\Pi_{cc}^{\Longleftrightarrow}(G) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \Pi_{cc}(G).$$

Before these results, there were large gaps between the known upper and lower bounds of the reversible cumulative pebbling complexity of graphs underlying prominent MHFs such as Argon2i [BDK16], Balloon Hash [BCS16], and Catena[FLW13], whereas their classical cumulative pebbling complexity is well understood. For example, for Argon2i (winner of the password hashing competition [PHC15]) the best classical pebbling attack on Argon2i has cumulative pebbling cost $\Pi_{cc}^{\parallel}(G) = \mathcal{O}\left(n^{1.768}\right)$ [BZ17] while the best reversible pebbling attack had cumulative cost $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) = \mathcal{O}\left(n^{1.8}\right)$ [BHL22]. Applying Theorem 3, we immediately obtain reversible pebblings which match the best classical pebblings up to this subpolynomial factor. In future sections, we will show how we can match the classical upper bounds for these particular functions within a constant factor. See Appendix C for the proof of Theorem 3.

3.2.1 Reversible Space-Time Complexity.

Let $P = (P_1, \ldots, P_t)$ be a (irreversible) pebbling or a graph $G = (V = [N], E)$ and $L = (L_1, \ldots, L_{t'})$ be a reversible pebbling for the line graph $\mathcal{L}_t$. We first want to show that the pebbling composition $Q = L \circ P = (Q_1, \ldots, Q_{t'})$, where $Q_i \coloneqq \bigcup_{j \in L_i} P_j$ for $i \in [t']$, is a *legal reversible pebbling*. Notice that since P starts as an empty pebbling, so does Q. Likewise, $L_{t'} = \{t\}$, which implies $Q_{t'} = \bigcup_{j \in L_{t'}} P_j = P_t$, meaning that the end conditions are also satisfied.

It remains to show that Q satisfies Property (2) of Definition 1, i.e., both Q and its reverse Q^* are *extra legal*. To show that Q is extra legal, we need to show that $\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq Q_i$ and $\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq Q_{i+1}$ for all $i \in [t'-1]$. Since $Q_i = \bigcup_{j \in L_i} P_j$, we observe that[5]

$$\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq \bigcup_{j \in \mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t)} P_j, \tag{1}$$

and from the extra legality of L, we have that $\mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t) \subseteq L_i$ and $\mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t) \subseteq L_{i+1}$. Hence, we have $\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq \bigcup_{j \in L_i} P_j = Q_i$ and $\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq \bigcup_{j \in L_{i+1}} P_j = Q_{i+1}$, which implies the extra

[5] See Appendix C for the formal proof of Eq. (1).

legality of Q. We can also prove that Q^* is extra legal using a similar argument (by switching $Q_{i+1} \leftrightarrow Q_i$ and $L_{i+1} \leftrightarrow L_i$). See Appendix C for further details.

Now we analyze the space-time complexity of Q. At any step i, Q_i contains at most $\Pi_s(L)$ configurations of P. Thus, $\Pi_s(Q) \leq \Pi_s(L) \cdot \Pi_s(P)$. Likewise, $\Pi_t(Q) = \Pi_t(L)$, leading to Theorem 4. See Appendix C for the formal proof of Theorem 4.

Theorem 4 (Reversible Composition Pebbling). *Let $P = (P_1, \ldots, P_t)$ be a (possibly irreversible) pebbling for a DAG G, and $L = (L_1, \ldots, L_{t'})$ be a reversible pebbling for $\mathcal{L}_t$. Then the composition $L \circ P$ is a legal reversible pebbling of G satisfying $\Pi_{st}(Q) \leq \Pi_s(P) \cdot \Pi_{st}(L)$.*

At a high level, Theorem 4 says that we can combine any pebbling for an arbitrary DAG G with a reversible pebbling of a line graph to obtain a reversible pebbling of G with comparable space-time complexity. We will use the reversible pebbling from [BHL22].

Theorem 5 (Reversible Line Graph Pebbling [BHL22]). *There exist a family of sequential pebblings L_N and a family of parallel reversible pebblings $L_N^{\parallel}$ for line graphs $\mathcal{L}_N$ such that*

(1) $\Pi_t\left(L_N\right) = \mathcal{O}\left(N^{1+\frac{1}{\sqrt{\log N}}}\right)$, $\Pi_s\left(L_N\right) = \mathcal{O}\left(N^{\frac{1}{\sqrt{\log N}}}\sqrt{\log N}\right)$, $\Pi_{st}\left(L_N\right), \Pi_{cc}\left(L_N\right) = \mathcal{O}\left(N^{1+\frac{2}{\sqrt{\log N}}}\sqrt{\log N}\right)$, *and* $\mathsf{toggle}(L_N) = \mathcal{O}\left(N^{\frac{1}{\sqrt{\log N}}}\right)$, *and*

(2) $\Pi_t\left(L_N^{\parallel}\right) = \mathcal{O}\left(N\right)$, $\Pi_s\left(L_N^{\parallel}\right) = \mathcal{O}\left(N^{\frac{2}{\sqrt{\log N}}}\right)$, $\Pi_{st}\left(L_N^{\parallel}\right), \Pi_{cc}\left(L_N^{\parallel}\right) = \mathcal{O}\left(N^{1+\frac{2}{\sqrt{\log N}}}\right)$, *and* $\mathsf{toggle}(L_N^{\parallel}) = \mathcal{O}\left(N^{\frac{1}{\sqrt{\log N}}}\right)$.

For completeness we provide proof of Theorem 5 in Appendix C. The results are implicit (but technically unproven) in the work of [BHL22].

Partial Proof of Theorem 2. If $P = (P_1, \ldots, P_t)$ is a pebbling of G and $L = (L_1, \ldots, L_{t'})$ of $\mathcal{L}_t$ and $Q = L \circ P$ is composed pebbling derived as in Theorem 4 then by Theorem 4 we have $\Pi_{st}(Q) = \Pi_s(P) \cdot \Pi_{st}(L) = \Pi_{st}(P) \cdot \Pi_{st}(L)/t$.

If $P = (P_1, \ldots, P_t)$ is the parallel pebbling of G with minimum space-time cost (i.e., $\Pi_{st}(P) = \Pi_{st}^{\parallel}(G)$) then $\Pi_{st}(Q) = \Pi_{st}^{\parallel}(G) \cdot \Pi_{st}(L)/t$. Taking $L = (L_1, \ldots, L_{t'})$ to be the parallel pebbling of $\mathcal{L}_t$ from Theorem 5 we have $\Pi_{st}(L)/t = \mathcal{O}\left(t^{\frac{2}{\sqrt{\log t}}}\right)$. Using the fact that $t \leq N^2$ (otherwise we would have $\Pi_{st}(P) > N^2$ and P would not be optimal) we have $\Pi_{st}(L)/t = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right)$. Hence, $\Pi_{st}^{\Longleftrightarrow,\parallel}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right) \cdot \Pi_{st}^{\parallel}(G)$. The proof for the sequential space-time cost is similar; see Appendix C for the full proof. □

Next, we see that composing a sequential reversible pebbling of a line graph with a "special" sequential pebbling of a graph G (which is a sequential pebbling with an extra constraint that only removes at most one pebble per round and never places and removes pebbles at the same time, i.e., for a pebbling $P = (P_1, \ldots, P_t)$, we have $|(P_i \cup P_{i+1}) \setminus (P_i \cap P_{i+1})| \leq 1$ for all $i \in [t-1]$) results in a reversible sequential pebbling for G. The proof is included in Appendix C.

Corollary 1. *If $P = (P_1, \ldots, P_t)$ is a special sequential pebbling of a DAG G and L is a reversible sequential pebbling of $\mathcal{L}_t$, then $L \circ P$ is a reversible sequential pebbling of G.*

Theorem 14 in Appendix C shows that we can transform any sequential pebbling of G to a special sequential pebbling without significantly increasing costs, e.g., space-time/cumulative pebbling costs increase by a small constant multiplicative factor.

3.2.2 Reversible Cumulative Pebbling Complexity. In this section, we will be giving a transformation that maps irreversible pebblings $P = (P_1, \ldots, P_t)$ of a graph $G = (V = [N], E)$ to reversible pebblings $Q = (Q_1, \ldots, Q_{t'})$ at the cost of just a subpolynomial factor in cumulative pebbling complexity. As with space-time complexity, the mapping will involve reversibly pebbling the line graph $\mathcal{L}_t$ associated with the given irreversible pebbling P of G. However, the method is much different. To see why the pebbling from Theorem 4 fails to preserve cumulative pebbling complexity, consider the reversible pebbling L of $\mathcal{L}_t$. The node i'_{k-1} in I'_k is kept for $\Omega(t)$ steps. It could be the case that the pebbling configuration $P_{i'_{k-1}}$ could be large (as large as $\Omega(N)$) as well. If this large space usage happens for a small amount of time in P, then $\Pi_{st}(P) \gg \Pi_{cc}(P)$ yet $\Pi_{cc}(Q)$ is of similar magnitude to $\Pi_{st}(Q) \gg \Pi_{cc}(P)$.

For this transformation, we will still be providing a reversible pebbling for $\mathcal{L}_t$, but we will have to avoid keeping pebbles on nodes i associated with large configurations P_i. For this reason, it will be useful to instead consider pebblings on weighted graphs. This way, we can describe pebbling strategies for $\mathcal{L}_t$, where the "weight" of node i is $\mathsf{wt}_i = |P_i|$.

Definition 5 (Weighted Graph Pebbling). *Let $G = (V, E)$ be a graph with weights wt_v for $v \in V$. For a pebbling $P = (P_1, \ldots, P_t)$ of G, the* weighted cumulative pebbling complexity (WCC) *of P is*

$$\Pi_{wcc}(P) = \sum_{i \in [t]} \sum_{v \in P_i} \mathsf{wt}_v,$$

and the weighted cumulative pebbling complexity *of G is*

$$\Pi_{wcc}(G) = \min_{P \in \mathcal{P}(G)} \Pi_{wcc}(P).$$

Consider a weighted line graph on N nodes. Our high-level goal is to minimize the number of pebbling rounds where we have pebbles on nodes with high weight. So, we construct a series of weight buckets $S_0, \ldots, S_\ell$, where S_0 are the lightest nodes and S_ℓ are the heaviest. In an ideal world, we would like to "ignore" heavier buckets and only pebble the nodes in S_0 pretending that these nodes form a line graph of length $|S_0|$. However, this strategy would yield an illegal pebbling of the entire graph as we are skipping over heavier nodes. We fix the issue recursively. In particular, consider nodes $u, v \in S_0$ and suppose that u is the predecessor of v in S_0 (i.e., any intermediate node w with $u < w < v$ has higher weight and is not in S_0). Now suppose that our pebble of S_0 illegally places (or removes) a pebble from node $v \in S_0$ skipping over all of the intermediate nodes between u and v. We can patch the pebbling by recursively pebbling the weighted subgraph induced by nodes $[u+1, v-1]$ and injecting these pebbling steps in between our pebbling of S_0 i.e., we recursively place a pebble on $v-1$, then place a pebble on v, then reverse the recursive pebbling to clear pebbles from the interval $[u+1, v-1]$. The number of times that we have to recursively pebble/unpebble this interval $[u+1, v-1]$ is upper bounded by the *toggle number* of our original pebbling of the line graph on $|S_0|$ nodes, which is the maximum number of times that a node is pebbled/unpebbled. In particular, this recursive call is made at most twice the toggle number of the pebbling of the line graph on $|S_0|$ nodes.

Now we describe in more detail the CC-efficient reversible, weighted line graph pebbling $\mathsf{WRevLinePeb}^{\|}$. In particular, we consider a line graph $\mathcal{L}_N$ with weights wt_i on node i satisfying $\sum_i \mathsf{wt}_i \leq N^2$. Note that without loss of generality, we can always take $\mathsf{wt}_1 = \mathsf{wt}_N = 1$, so assume this to be the case. recall that to keep the cumulative cost low, we will aim to keep pebbles on "heavy" nodes for as little time as possible, placing pebbles on the heaviest nodes only when necessary. We first partition nodes according to their weight such that $\mathcal{S} = (S_0, \ldots, S_\ell)$. Later, we will take care in assigning nodes to buckets to ensure that (1) there are not too many nodes in heavier buckets, and (2) ℓ is small, meaning there are not too many buckets overall.

Fix some family of reversible line graph pebblings $L(i)$ for $\mathcal{L}_i$ for $1 \leq i \leq N$ i.e., $L(i)$ outputs a reversible pebbling of $\mathcal{L}_i$. A set $S \subseteq [N]$ induces a line graph $\mathcal{L}_S$, where the i^{th} node of $\mathcal{L}_S$ is the i^{th} smallest value in S. We similarly let $L(S)$ denote the pebbling of $\mathcal{L}_S$ corresponding to $L(|S|)$. As L is a family of pebblings, recall that $L(i)_j$ is the j^{th} pebbling configuration of the pebbling of the line graph $\mathcal{L}_i$. For

- set of buckets $\mathcal{S} = (S_1, \ldots, S_\ell)$ based on the set of weights $\mathsf{wt} = (\mathsf{wt}_1, \ldots, \mathsf{wt}_N)$, and
- an interval $I = [a, b] \subseteq [N]$[6] and integer $i \in [0, \ell]$ such that $I \subseteq \mathcal{S}_{\geq i} := \bigcup_{j \geq i} \mathcal{S}_j$,

the weighted line graph pebbling $\mathsf{WRevLinePeb}^{\|}(I, \mathcal{S}, i, L)$ of $\mathcal{L}_I$ with weights defined by wt is defined in Algorithm 1. See Figure 1 for an illustrative example and see Appendix C.1 for further details including a concrete example.

In Figure 1, we are given a line graph $\mathcal{L}_{14}$ with 14 nodes, based on the weight $\mathsf{wt} = (\mathsf{wt}_1, \ldots, \mathsf{wt}_{14})$ that is shown above each node, we can construct subgraphs $\mathcal{L}_{S_0}, \ldots, \mathcal{L}_{S_2}$. In each subgraph, a solid edge means it is legal to pebble the next node, and a dashed edge means it is illegal to proceed with pebbling and we would need to make a recursive call. For example, in $\mathcal{L}_{S_0}$, it is illegal to place a pebble on

[6] Recall that if $a > b$ then $[a, b] = \emptyset$ and $[a, a] = \{a\}$

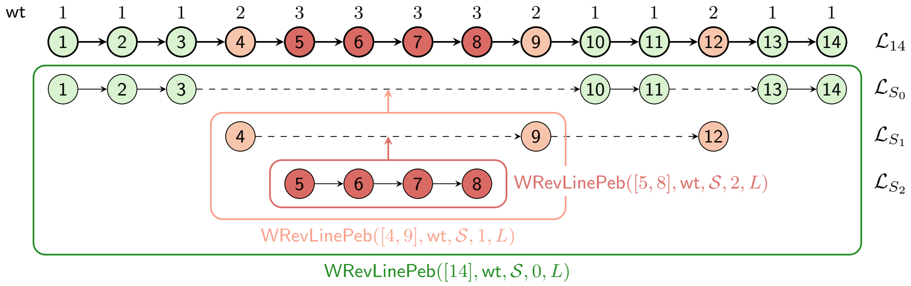

wt
1 1 1 2 3 3 3 3 2 1 1 2 1 1
1 2 3 4 5 6 7 8 9 10 11 12 13 14
$\mathcal{L}_{14}$
$\mathcal{L}_{S_0}$
$\mathcal{L}_{S_1}$
$\mathcal{L}_{S_2}$
$\mathsf{WRevLinePeb}([5,8], \mathsf{wt}, \mathcal{S}, 2, L)$
$\mathsf{WRevLinePeb}([4,9], \mathsf{wt}, \mathcal{S}, 1, L)$
$\mathsf{WRevLinePeb}([14], \mathsf{wt}, \mathcal{S}, 0, L)$

Fig. 1: An illustrative example of $\mathsf{WRevLinePeb}([14], \mathcal{S}, 0, L)$.

node 10 from node 3, so we need to run $\mathsf{WRevLinePeb}([4,9], \mathcal{S}, 1, L)$ recursively to place a pebble on node 9 and then proceed to node 10. One important observation here is that even though the number of recursive calls grows exponentially with the level of recursion, the size of the nodes in each level is decreasing even faster. This makes our weighted reversible pebbling CC-efficient.

To analyze the weighted cumulative pebbling complexity of our pebbling, we need to know the maximum number of times we place or remove pebbles on any particular node. Recall that the toggle number for a node v in a pebbling P is $\mathsf{toggle}(v, P) = |\{i \mid v \in P_i \oplus P_{i+1}\}|$, and $\mathsf{toggle}(P) = \max_v \mathsf{toggle}(v, P)$. The toggle number of our non-weighted reversible line graph pebbling will help us upper bound the number of times we will end up pebbling nodes in S_i.

Consider the pebbling $\mathsf{WRevLinePeb}^{\|}([N], \mathcal{S} = (S_0, \ldots, S_\ell), 0, L)$ of the weighted line graph on N nodes and weights wt. The analysis consists of two components for each $i \in [\ell]$: (1) $T(i)$, the number of steps that at least one pebble is contained in S_i, and (2) $M(i)$, the greatest number of pebbles contained in S_i at any step. This way,

$$\Pi_{wcc}\left(\mathsf{WRevLinePeb}^{\|}([N], \mathcal{S}, 0, L)\right) \leq \sum_{0 \leq i \leq \ell} T(i) M(i) \max_{j \in S_i} \mathsf{wt}_j .$$

First we bound $T(i)$. For now, assume $\Pi_t(L(N)) \leq cN$ for some constant c. If we have sub-intervals $I_1, \ldots, I_k \subseteq [N]$, then the time it takes to pebble each interval individually is at most $c \sum_i |I_i|$. Now consider the number of steps in which there's a pebble in S_ℓ. Every time we pebble/unpebble a node in the S_0 pebbling, we call a pebbling in S_1. This happens at most $\tau := 2 \cdot \mathsf{toggle}(L(N))$ times (to pebble then unpebble). Therefore, throughout the pebbling of S_0, we (re)pebble nodes in S_ℓ at most $2^\ell \tau^\ell$ times, and the total number of steps with a pebble in S_ℓ is at most $T(\ell) \leq c 2^\ell \tau^\ell |S_\ell|$. Now consider $S_{\ell-1}$. We similarly see that we repebble $S_{\ell-1}$ at most $2^{\ell-1}\tau^{\ell-1}$ times, but now we may also have pebbles in $S_{\ell-1}$ while we're waiting for pebblings of subsets of S_ℓ to complete. Thus, $T(\ell-1) \leq c 2^{\ell-1}\tau^{\ell-1}|S_{\ell-1}| + c 2^\ell \tau^\ell |S_\ell|$. More generally, we have

$$T(i) \leq c \sum_{i \leq j \leq \ell} 2^j \tau^j |S_j| \leq c(\ell+1) 2^\ell \tau^\ell |S_i|,$$

Algorithm 1: $\mathsf{WRevLinePeb}^{\|}(I = [a,b], \mathcal{S} = (S_0, \ldots, S_\ell), i, L)$

```
   Preconditions : have pebble on node $a-1$ (or $a=1$) and $I \subseteq S_{\geq i}$
   Postconditions: have pebble on node $b$
 1 if $i = \ell + 1$ or $I = \emptyset$ then
 2     return
 3 for $j = 1, \ldots, |L(I \cap S_i)| - 1$ do
 4     foreach $v \in L(S_i \cap I)_j \oplus L(S_i \cap I)_{j+1}$;   // in parallel for each $v$
         to be pebbled or unpebbled (Note: If different parallel calls
         to $\mathsf{WRevLinePeb}^{\|}$ take different number of steps then delay
         execution for shorter recursive calls so that they all finish
         on the same round)
 5      do
 6         Let $u = \max\{a-1\} \cup (I \cap S_i \cap [v-1])$;        // $v$'s predecessor
 7         Let $I' = [u+1, v-1]$;
 8         Pebble $I'$ using $\mathsf{WRevLinePeb}^{\|}(I', \mathcal{S}, i+1, L)$;
 9         if $v \in L(S_i \cap I)_{j+1}$ then
10             Pebble $v$;                                  // as $v-1$ is pebbled
11         else
12             Unpebble $v$;
13         Unpebble $I'$ by reversing $\mathsf{WRevLinePeb}^{\|}(I', \mathcal{S}, i+1, L)$;
14 Let $b_i = \max\{a-1\} \cup (I \cap S_i)$ ; // lines 4-14 leave pebble on node $b_i$
15 Run $\mathsf{WRevLinePeb}^{\|}([b_i + 1, b], \mathcal{S}, i+1, L)$;  // finish pebbling if $b_i < b$
```

under the assumption that the sizes of the buckets S_i are decreasing with respect to i. While this bound may seem crude, we will assign the buckets $\mathcal{S}$ such that 2^ℓ and τ^ℓ are small, subpolynomial terms, meaning $T(i)$ is not too much larger than $|S_i|$ in general.

Now we bound $M(i)$. By the construction of L, $M(0) \leq \mathit{\Pi}_s(L(S_0))$. Notice that the pebbling L cannot pebble/repebble more than $\mathit{\Pi}_s(L(S_0))$ nodes in a single step. Then for S_1, there are at most $\mathit{\Pi}_s(L(S_0))$ calls in a single step to intervals containing nodes in S_1. For each of these calls, there are at most $\mathit{\Pi}_s(L(S_1))$ pebbles on the graph in S_1. So, $M(1) \leq \mathit{\Pi}_s(L(S_1)) \cdot \mathit{\Pi}_s(L(S_0))$. More generally, we see that

$$M(i) \leq \prod_{0 \leq j \leq i} \mathit{\Pi}_s(L(S_j)) \leq \mathit{\Pi}_s(N)^{i+1}.$$

Here, we rely on the fact that both ℓ and $\mathit{\Pi}_s(i)$ are relatively small. We will see shortly that $\mathit{\Pi}_s(i)^\ell$ is still subpolynomial.

Putting it all together, we get

$$\mathit{\Pi}_{wcc}\left(\mathsf{WRevLinePeb}^{\|}([N], \mathcal{S}, 0, L)\right) \leq \sum_{0 \leq i \leq \ell} T(i) M(i) \max_{j \in S_i} \mathsf{wt}_j$$

$$\leq c(\ell+1)2^{\ell}\tau^{\ell}\sum_i |S_j| \cdot \mathit{\Pi}_s(L(N))^i \max_{j\in S_i} \mathsf{wt}_i.$$

Now fix $L = \mathsf{RevLinePeb}^{\|}$. All that is left is to define the buckets. Let $\mathsf{wt}_{\mathsf{avg}}$ be the average weight in wt and $S_i = \left\{j \mid \tau^{\alpha i}\mathsf{wt}_{\mathsf{avg}} \leq \mathsf{wt}_j \leq \tau^{\alpha(i+1)}\mathsf{wt}_{\mathsf{avg}}\right\}$, where $\tau = \mathsf{toggle}(L(S_0)) = \Theta\left(N^{\frac{1}{\sqrt{\log N}}}\right)$ and $\alpha = \sqrt[4]{\log N}$. This implies that the number of weight buckets is $\ell \leq \frac{\log N}{\alpha \log \tau} = \mathcal{O}\left(\sqrt[4]{\log N}\right)$. Now, we know that $|S_i| \leq \frac{\sum_j \mathsf{wt}_j}{\tau^{\alpha i}\mathsf{wt}_{\mathsf{avg}}} = \frac{N}{\tau^{\alpha i}}$. Thus, the sizes of the sets S_i shrink fairly quickly with respect to i. Consequently, the summand

$$T(0)M(0)\max_{j\in S_0}\mathsf{wt}_j = N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \cdot N^{1+\frac{1}{\sqrt{\log N}}} = N^{1+\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}$$

dominates the entire WCC sum above. This results in Theorem 6, which takes a crucial part in proving Theorem 3 (see Appendix C for the full proof of Theorem 3).

Theorem 6 (Reversible Cumulative Pebbling Complexity of Weighted Line Graphs). *Given a weighted line graph $\mathcal{L}_N$ with weights $\mathsf{wt}_i \leq N$ for nodes $i \in N$. Then there exists a parallel reversible pebbling P and sequential pebbling S for $\mathcal{L}_N$ with*

$$\mathit{\Pi}_t(P) = \mathcal{O}(N), \qquad \mathit{\Pi}_{wcc}(P) = N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \cdot \sum_i \mathsf{wt}_i, \text{ and}$$

$$\mathit{\Pi}_t(S) = \mathcal{O}\left(N^{1+\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right), \qquad \mathit{\Pi}_{wcc}(S) = N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \cdot \sum_i \mathsf{wt}_i.$$

Proof. Consider the pebbling $P = \mathsf{WRevLinePeb}^{\|}([N], \mathcal{S}, 0, L = \mathsf{RevLinePeb}^{\|})$ as defined in the discussion above. First, we have that $|S_i| \leq \frac{\sum_j \mathsf{wt}_j}{\tau^{\alpha i}\mathsf{wt}_{\mathsf{avg}}} \leq \frac{N}{\tau^{\alpha i}}$. Next $\mathit{\Pi}_s(N)^i \leq c' N^{\frac{2\sqrt{2}i}{\sqrt{\log N}}}$ for some constant $c' > 0$. Finally, $\max_{j\in S_i}\mathsf{wt}_j \leq \tau^{\alpha(i+1)}\mathsf{wt}_{\mathsf{avg}}$. So, the weighted cumulative pebbling complexity is at most

$$\begin{aligned}
\mathit{\Pi}_{wcc}(P) &\leq \sum_i T(i)M(i)\max_{j\in S_i}\mathsf{wt}_j \\
&\leq c(\ell+1)2^{\ell}\tau^{\ell}\sum_{0\leq i\leq \ell}|S_i| \cdot \mathit{\Pi}_s(L(N))^i \max_{j\in S_i}\mathsf{wt}_i \\
&\leq cc'(\ell+1)2^{\ell}\tau^{\ell+\alpha}N\mathsf{wt}_{\mathsf{avg}}\sum_{0\leq i\leq \ell} N^{\frac{2\sqrt{2}i}{\sqrt{\log N}}} \\
&\leq cc'(\ell+1)^2 2^{\ell}\tau^{\ell+\alpha}N^{\frac{2\sqrt{2}\ell}{\sqrt{\log N}}} \cdot N\mathsf{wt}_{\mathsf{avg}} \\
&\leq cc'(\ell+1)^2 2^{\ell}\tau^{\ell+\alpha}N^{\frac{2\sqrt{2}\ell}{\sqrt{\log N}}} \cdot \sum_{j\in[N]}\mathsf{wt}_j.
\end{aligned}$$

Now we need to analyze the coefficient on $\sum_j \mathsf{wt}_j$. Since $\ell = \mathcal{O}\left(\sqrt[4]{\log N}\right)$ and $\alpha = \mathcal{O}\left(\sqrt[4]{\log N}\right)$, it follows that $2^\ell = N^{\frac{\mathcal{O}(1)}{\log^{3/4} N}}$ and $\tau^{\ell+\alpha} = N^{\frac{\mathcal{O}\left(\sqrt[4]{\log N}\right)}{\sqrt{\log N}}} = N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}$. Putting it all together, we get

$$\Pi_{wcc}(P) = N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \sum_{j \in [N]} \mathsf{wt}_j .$$

Since we assume nodes $1, N \in S_0$, the time complexity of P at most

$$\begin{aligned}
\Pi_t(P) \le T(0) &\le c \sum_{0 \le j \le \ell} 2^j \tau^j |S_j| \\
&\le c|S_0| + c \sum_{1 \le j \le \ell} 2^j \frac{N}{\tau^{j(\alpha-1)}} \\
&\le cN + 2c\ell \frac{N}{\tau^{\alpha-1}} && \triangleleft \text{ sum is decreasing in } j \\
&\le 3cN && \triangleleft \ \ell = o(\tau^{\alpha-1}).
\end{aligned}$$

To finish up, we need a sequential weighted pebbling Q. Note that we can sequentially simulate P by executing the pebble placing/removing one at a time per step. There are at most $2\Pi_s(P)$ pebbles placed or removed in a pebbling step of P. Then $\Pi_t(Q) \le \Pi_t(P) \cdot 2\Pi_s(P)$. Likewise, $\Pi_s(Q) \le 2\Pi_s(P)$. The number of steps spent with a pebble in S_i increases by at most a factor of $\Pi_s(P)$. We have that $\Pi_s(P) \le \ell \tau^\ell N^{\frac{2}{\sqrt{\log N}}} = N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}$, so

$$\begin{aligned}
\Pi_{wcc}(Q) &= N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \Pi_{wcc}(P) \\
&= N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \cdot N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \cdot \sum_{j \in [N]} \mathsf{wt}_j \\
&= N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}} \cdot \sum_{j \in [N]} \mathsf{wt}_j .
\end{aligned}$$

□

4 Reversible Recursive Pebbling Attack

In Section 3, we showed that the reversible cumulative pebbling complexity of a graph is always within a factor of $N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}$ of the classical cumulative pebbling complexity. In this section, we show for classes of graphs that satisfy certain depth-reducibility properties, there are reversible pebblings that match the best-known classical pebblings in cumulative pebbling complexity. Blocki et al. [BHL22] introduced a reversible pebbling attack for (e, d)-reducible graphs $G = (V = [N], E)$, based on the classical depth-reducing attack of [ABP17].

Theorem 7 (Reversible Depth-Reducing Pebbling Strategy). *For any (e, d)-depth reducible graph $G = (V = [N], E)$, target set T, and parameter*

$g \in [d, N]$, there exists a reversible parallel pebbling $P = (P_1, \dots, P_{2N}) =$ $\mathsf{RGenPeb}(G)$ with $P_{2N} = T$ such that

$$\Pi_{cc}(P) \le 2N\left(\frac{2Nd}{g} + e + (\delta+1)g + |T|\right) + N + \frac{2Nd}{g}.$$

We construct a more general reversible pebbling attack based on the recursive attack of [ABP17]. As a result, we obtain asymptotically stronger reversible CC upper bounds for several iMHFs.

Review of Algorithm in Theorem 7: Let $G = (V = [N], E)$ be an (e, d)-depth robust graph with depth reducing set $S \subseteq [N]$ of size at most e_1. The pebbling $\mathsf{RGenPeb}(G)$ is composed of a sequence of alternating phase: *light phases* and *balloon phases*. Each light phase lasts $2g$ rounds. The goal of the c^{th} light phase is to pebble the nodes $I_c = [(c-1)g + 1, cg]$ one at a time with low space usage. To achieve this, we will enforce the light phase precondition on the pebbling configuration P_j, the step before the start of the c^{th} light phase. In particular, it must be the case that

$$\mathsf{LightReq}_0^c = S_{\le (c-1)g+1} \cup \mathsf{parents}(I_c) \setminus I_c.$$

If this condition is satisfied, then we can simply place a pebble on node $(c-1)g+k$ in P_{j+k} for all $k \in [g]$. The end condition for the c^{th} light phase is then $P_{j+g} = S_{\le cg} \cup I_c \cup \mathsf{parents}(I_c)$. We then reverse the light phase, while keeping pebbles only on $S_{\le cg}$, so $\mathsf{LightReq}_{g+j}^c = \mathsf{LightReq}_{g-j}^c \cup S_{\le cg}$. However, this leaves us unprepared for the $(c+1)^{th}$ light phase. To fix this, we can simply start a balloon phase with the goal of pebbling $\mathsf{LightReq}_0^{c+1}$. The pebbling attack of [BHL22] accomplishes this by simply applying a greedy pebbling strategy. In particular, if $\mathsf{BalloonReq}_{2g-2d-1}^c$ is the step before the balloon phase begins, we must have pebbles on $S_{\le cg}$. Then $\mathsf{BalloonReq}_{2g+j}^c$ pebbles any node that can be legally pebbled from $\mathsf{BalloonReq}_{2g+j-1}^c$. In d rounds, nodes $[cg]$ will be pebbled. Then we can reverse both the light phase and the balloon phase, keeping pebbles only on $S_{\le cg} \cup \mathsf{parents}(I_{c+1}) \setminus I_{c+1}$.

Now we can describe the reversible recursive attack $\mathsf{RRGenPeb}$. The main difference is that we replace the greedy balloon phases with more efficient algorithms when G is (e_i, d_i)-depth reducible along multiple points i. The proof is similar to that of [ABP17], but special consideration is needed to account for reversibility.

4.1 Reversible Recursive Pebbling Strategy

Let $G = (V = [N], E)$ be an (e_1, d_1)-depth reducible graph of depth $d_1 \le d_0$, satisfying $2d_1N \le e_1d_0$. Our goal is to pebble some target set $T \subseteq V$. The light phases will be pebbling intervals of length $g = \left\lceil \frac{e_1 d_0}{N} \right\rceil \ge 2d_1$. These light phases are slightly different than in $\mathsf{RGenPeb}$. Since we know that the depth of G is $d_0 \le N$, we can instead pebble all nodes of the same depth each step, meaning the pebbling time will be at most $2d_0$. More formally, we define sets $D_1, \dots D_{2d_1}$ such that $\mathsf{parents}(D_1) = \emptyset$, $\mathsf{parents}(D_{i+1}) \subseteq \bigcup_{j \le i} D_j$, and each $|D_i| \le \frac{N}{d_0}$. Analogously to before, we let $I_c = \bigcup_{1 \le j \le cg} D_j$. Likewise, for any set R, we let $R_{\preceq i} := R \cap \bigcup_{j \le i} D_j$. So, for $0 \le i \le g$, the i^{th} step of the c^{th} light phase will maintain

$$\mathsf{LightReq}_i^c = S_{\preceq (c-1)g+i} \cup T_{\preceq (c-1)g+i} \cup \bigcup_{(c-1)g \le j \le \min\{(c-1)g+i, N\}} D_j.$$

As before, we will let $\mathsf{LightReq}_{cg+i}^c = \mathsf{LightReq}_{cg-i}^c \cup S_{\preceq cg} \cup T_{\preceq cg}$ for $0 \le i \le \min\{g, N - cg\}$. Now, for some G' with depth at most d, let $B(G', T', t)$ be a pebbling of G' with the target set T' that terminates in at most $t \ge 2d$ steps. Then we can let

$$\mathsf{BalloonReq}^c = B\left(G_{\preceq cg} - S_{\preceq cg}, \mathsf{parents}\left(I_{c+1}\right) \setminus I_{c+1}, 2d_1\right).$$

There are technicalities we must account for with these new balloon phases:

- **(Before Start)** We let $\mathsf{BalloonReq}_j^c = \emptyset$ for $1 \le j \le cg - 2d_1$.
- **(Early Termination)** If $\mathsf{BalloonReq}$ terminates in less than $t \le 2d_1$ rounds, then we will let $\mathsf{BalloonReq}_{cg-2d_1+t+j}^c = \mathsf{BalloonReq}_t^c$ for $1 \le j \le 2d_1 - t$.

The pebbling, excluding clean-up, is

$$P' := \mathsf{LightReq}^1 \cup \mathsf{BalloonReq}^1 + \dots + \mathsf{LightReq}^{\lceil 2d_0/g \rceil - 1} \cup \mathsf{BalloonReq}^{\lceil 2d_0/g \rceil - 1} + \mathsf{LightReq}^{\lceil 2d_0/g \rceil}$$

The final pebbling $P = \mathsf{RRGenPeb}(G, \{(e_1, d_1, S_1)\}, B)$ is obtained by then reversing P' while keeping nodes on the target set T. More formally, for all $1 \le j \le |P'|$, we have $P_{|P'|+j} = P'_{|P'|-j} \cup T$. Showing that P is a legal reversible pebbling is straightforward, and the proof is left to the Appendix D.

Lemma 3. *For any (e_1, d_1)-depth reducible DAG $G = (V = [N], E)$ of depth d_0, target set $T' \subseteq [N]$, and family of pebblings $B(G', T', t')$ for all DAGs $G' = (V', E')$, target sets $T' \subseteq V'$, and $t' \ge 2 \cdot \mathsf{depth}(G')$, the pebbling*

$$P = \mathsf{RRGenPeb}(G, d_0, \{(e_1, d_1, S_1)\}, B)$$

is a legal parallel reversible pebbling of G, where S_1 is a depth-reducing set of size e_1.

Now we bound the CC of P. The argument is a straightforward accounting of (1) the CC contributions of the light phases and (2) the CC of the B called $\frac{2e_1}{N}$ times. The proof is left to Appendix D.

Lemma 4. *For any (e_1, d_1)-depth reducible DAG $G = (V = [N], E)$ of depth d_0, target set $T' \subseteq [N]$, and family of pebblings $B(G', T', t')$ for all DAGs $G' = (V', E')$, target sets $T' \subseteq V'$, and $t' \ge 2 \cdot \mathsf{depth}(G')$,*

$$\begin{aligned} &\Pi_{cc}\left(\mathsf{RRGenPeb}(G, d_0, \{(e_1, d_1, S_1)\}, B)\right) \\ &\quad \le 4d_0(\delta + 2)e_1 + 4d_0|T| + \frac{2e_1}{N} \cdot \max_{|T'| \le \delta e_1} \Pi_{cc}\left(B\left(G - S, T', 2d_1\right)\right), \end{aligned}$$

where S_1 is a depth-reducing set of size e_1.

By replacing B with a CC-optimal pebblings, we obtain Theorem 8.

Theorem 8. *Let $G = (V = [N], E)$ be an (e_1, d_1)-depth robust graph with depth d_0, then*

$$\Pi_{cc}^{\longleftrightarrow,\parallel}(G, T, 4d_0) \leq 4d_0(\delta + 2)e_1 + 4d_0|T| + \frac{2e_1}{N} \cdot \max_{|T'| \leq \delta e_1} \Pi_{cc}\left(G - S, T', 2d_1\right).$$

Now we will apply our theorem on graphs that are (e_i, d_i)-depth reducible for more than two values of i. It will be useful to employ a more general notion of depth-reducibility.

Definition 6 (f-reducibility, [ABP17]). *Let $G = (V, E)$ be a DAG with N nodes and let $f : \mathbb{N} \to \mathbb{N}$ be a function. We say that G is* f-reducible *if for every positive integer $0 < d \leq N$, G is $(f(d), d)$-depth reducible.*

Next, we show that if g is f-reducible and decreasing slowly enough in d, then we can apply Theorem 8 recursively to obtain better Reversible CC upper bounds. The proof of this lemma follows almost exactly as the analogous theorem in [ABP17], so the proof is left in Appendix C.

Lemma 5. *Let G be an f-reducible DAG of depth on N nodes then if $f(d) = \widetilde{\mathcal{O}}\left(\frac{N}{d^b}\right)$ for some constant $0 < b \leq 2/3$ and let $a = \frac{1-2b+\sqrt{1+4b^2}}{2}$. Then for any constant $\varepsilon > 0$, $\Pi_{cc}^{\longleftrightarrow,\parallel}(G) \leq \mathcal{O}\left(\delta N^{1+a+\epsilon}\right)$.*

It turns out that many graphs of interest are f-reducible as required in the above lemma. In particular, we examine:

(1) Argon2i won the 2015 Password Hashing Competition. We use Argon2iB to refer to the current version and we use Argon2iA is Argon2's original edge distribution (uniform) and Argon2iB to refer to the current (non-uniform) edge distribution [BDK16].
(2) Balloon Hash is a prominent memory-hard function introduced by [BCS16]. We examine the single buffer (SB) graph SB_N and the double buffer and linear graphs Lin_τ^σ on $N = \sigma \cdot \tau$ nodes as defined in [ABP17].
(3) Catena was a finalist in the 2015 Password Hashing Competition [FLW13]. We examine Catena graphs DFG_λ^N and BFG_λ^N as defined in [ABP17].

Lemma 6 ([ABP17], [BZ17]). *Let $f_b(d) = \widetilde{\mathcal{O}}\left(\frac{N}{d^b}\right)$, then*

(1) With high probability, $\mathsf{Argon2i\text{-}A}_N$ is $f_{0.5}$-reducible.
(2) With high probability, $\mathsf{Argon2i\text{-}B}_N$ is $f_{1/3}$-reducible.
(3) With high probability, SB_N is $f_{0.5}$-reducible.
(4) The Balloon Hashing (Linear and Double Buffer (DB)) graph Lin_τ^σ is f_1-reducible for $\tau = \mathcal{O}\left(\mathrm{polylog}(N)\right)$.
(5) The Catena Double Butter are both f_1-reducible for $\lambda = \mathcal{O}\left(\mathrm{polylog}(N)\right)$.

Now we can put these results together to upper bound the reversible CC of graph underlying MHFs.

Corollary 2. *We have the following:*

(1) $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{Argon2i\text{-}A}_N) = \mathcal{O}\left(N^{1.708}\right)$*,*
(2) $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{Argon2i\text{-}B}_N) = \mathcal{O}\left(N^{1.768}\right)$*,*
(3) $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{SB}_N) = \mathcal{O}\left(N^{1.708}\right)$*,*
(4) $\Pi_{cc}^{\Longleftrightarrow,\parallel}\left(\mathsf{Lin}_\tau^\sigma\right) = \widetilde{\mathcal{O}}\left(N^{\frac{13}{8}}\right) = \widetilde{\mathcal{O}}\left(N^{1.625}\right)$*, where the number of vertices is* $N = \sigma\tau$*, and*
(5) $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{DFG}_\lambda^N), \Pi_{cc}^{\Longleftrightarrow,\parallel}\left(\mathsf{BFG}_\lambda^N\right) = \widetilde{\mathcal{O}}\left(N^{\frac{13}{8}}\right) = \widetilde{\mathcal{O}}\left(N^{1.625}\right)$*.*

5 Depth Robustness and Reversible CC

In this section, we improve the lower bound of *reversible* CC for a depth-robust DAGs. Alwen et al. [ABP17] proved the lower bound of *classical* CC of a DAG G given its depth-robustness. In particular, they showed that if G is (e,d)-depth robust then $\Pi_{cc}^{\parallel}(G) \geq ed$. This immediately implies the same lower bound for *reversible* CC as well since for any DAG G we have $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) \geq \Pi_{cc}^{\parallel}(G)$. However, it was not known if there is a *tighter* lower bound for reversible CC in terms of depth-robustness. We provide a constant-factor (factor of ≈ 2) improvement on the lower bound of reversible CC when a DAG is depth-robust. Our main results are stated in Theorem 9 and Theorem 10.

We first consider a *non-relaxed* reversible pebbling, where we would require the condition that in the final round we have pebbles only on the sink nodes and pebbles from all of the intermediate nodes have been removed. Since removing pebbles is not free in a reversible pebbling and needs reversible pebbling steps, we can get a better lower bound for a reversible CC than a classical CC.

Theorem 9. *If G is (e,d)-depth-robust DAG then $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) \geq e(2d-1)$. Furthermore, if $G - \mathsf{sinks}(G)$ is (e,d)-depth-robust then $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) \geq 2ed$.*

Proof. Let $P = (P_1, \ldots, P_t)$ be a parallel reversible pebbling for G such that $\Pi_{cc}(P) = \Pi_{cc}^{\Longleftrightarrow,\parallel}(G)$. We first consider the case that G is (e,d)-depth-robust.

We will show that there exists a set $B \leq \frac{\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)}{2d-1}$ such that there is no path of length d in $G - B$, meaning G *is not* $\left(\frac{\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)}{2d-1}, d\right)$-depth robust. If G is (e,d)-depth robust for some e, then it must be the case that $e \leq \frac{\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)}{2d-1}$, implying $e(2d-1) \leq \Pi_{cc}^{\Longleftrightarrow,\parallel}(G)$.

Let $B_i = P_i \cup P_{i+2d-1} \cup P_{i+2(2d-1)} \cup \ldots$ for $i \in [2d-1]$ (defining $P_j = \emptyset$ for $j > t$). Since $\sum_i |B_i| \leq \sum_j |P_j| = \Pi_{cc}^{\Longleftrightarrow,\parallel}(G)$, there exists some $B := B_i$ in which $|B_i| \leq \frac{\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)}{2d-1}$.

Now we will show there is no path of length d in $G - B$. Let $v_1, \ldots, v_d$ be a path in G and let $p(v_d)$ be the first step in which node v_d is pebbled. Let $k < p(v_d)$ denote the last round before $p(v_d)$ when we had no pebble on the entire path

$\{v_1, \ldots, v_d\}$. Let $p(v_i)$ denote the first step after round k where we place a pebble on node v_i (because v_1 is the first node in our path we have $p(v_1) = k+1$). Then $p(v_1) < p(v_2) < \cdots < p(v_d)$ by Item 2 of Definition 7. Now let $u(v_i)$ denote the first round after round $p(v_d)$ where we remove a pebble from node v_i. Observe we always have *at least one* pebble on our path $v_1, \ldots, v_d$ in between rounds $p(v_1)$ and $u(v_1)$ inclusive. If v_d is a sink node that it is possible that $u(v_d) = \infty$. However, we are guaranteed that $u(v_1) > u(v_2) > \cdots > u(v_{d-1})$ by Item 4 of Definition 7 and we also know that $u(v_{d-1}) > p(v_d)$ since we needed to have a pebble on node v_{d-1} in round $p(v_d) - 1$ and we are not allowed to simultaneously remove the pebble from node v_{d-1} while we are placing a pebble on node v_d.

This means that $|\{i : p(v_1) \leq i \leq u(v_1)\}| \geq 2d - 1$. It follows that there is some j such that $p(v_1) \leq i + j(2d-1) \leq u(v_1)$. Since, $|P_j \cap \{v_1, \ldots, v_d\}| \geq 1$ it follows that B_i contains at least one node on our path. Since every path of length d intersects with B, G is not $\left(\frac{\Pi_{cc}^{\leftrightarrows,\parallel}(G)}{2d-1}, d\right)$-depth robust.

The argument is similar when we assume $G - \mathsf{sinks}(G)$ is (e, d)-depth robust. We now define $B_i = P_i \cup P_{i+2d} \cup P_{i+2(2d)} \cup P_{i+3(2d)} \ldots$ for $i \in [2d]$ (defining $P_j = \emptyset$ for $j > t$). Similar to our above argument there exists some $B = B_i$ such that $|B_i| \leq \frac{\Pi_{cc}^{\leftrightarrows,\parallel}(G)}{2d}$. Now if $v_1, \ldots, v_d$ is a path of length d in $G - \mathsf{sinks}(G)$ then v_d cannot be a sink node (by definition). We therefore have $p(v_d) < u(v_d) < u(v_{d-1})$ and it follows that $|\{i : p(v_1) \leq i \leq u(v_1)\}| \geq 2d$ since $p(v_1) < p(v_2) < \ldots p(v_d) < u(v_d) < \ldots < u(v_1)$. Therefore, B_i contains at least one node on our path since there exists some j such that $p(v_1) \leq i + 2jd \leq u(v_1)$. Since every path of length d in $G - \mathsf{sinks}(G)$ intersects with B it follows that $G - \mathsf{sinks}(G)$ is not $\left(\frac{\Pi_{cc}^{\leftrightarrows,\parallel}(G)}{2d}, d\right)$-depth robust. Since $G - \mathsf{sinks}(G)$ is (e, d)-depth robust it follows that $\Pi_{cc}^{\leftrightarrows,\parallel}(G) \geq 2ed$. □

On the other hand, Theorem 9 is *not* directly applicable to the *relaxed* reversible pebbling since it is not necessary to unpebble intermediate nodes. Considering that unpebbling is the reverse of pebbling, it is tempting to suggest that the reversible CC of relaxed pebbling might be approximately half that of non-relaxed pebbling. However, we can indeed derive a similar lower bound to the non-relaxed setting for depth-robust graphs. Oversimplifying a bit, a main bottleneck why the proof of Theorem 9 does not apply to the relaxed reversible pebbling is that there might be a possibility of having a path of length longer than d in $G - B$ if $N \equiv s \mod 2d$ with $s > d$ where N is the number of nodes in G. We can resolve this issue by truncating last d nodes from the graph. Given a DAG $G = (V = [N], E)$, we define $G_{\mathsf{Trunc},d} \coloneqq G - [N - d + 1, N]$ to be a DAG which truncates last d nodes and incident edges from G. Then we have the following theorem. The proof of Theorem 10 and analysis of the relaxed reversible CC of DRSample [ABH17] can be found in Appendix E.

Theorem 10. *Let $G = (V = [N], E)$ be a DAG such that $(i, i+1) \in E$ for all $i < N$ and the graph $G_{\mathsf{Trunc},d}$ is (e, d)-depth robust. Then $\widetilde{\Pi}_{cc}^{\leftrightarrows,\parallel}(G) \geq e(2d-1)$.*

References

AB16. Joël Alwen and Jeremiah Blocki. Efficiently computing data-independent memory-hard functions. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 241–271. Springer, Berlin, Heidelberg, August 2016. 2, 4, 7, 33, 34, 35

AB17. Joël Alwen and Jeremiah Blocki. Towards practical attacks on argon2i and balloon hashing. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, pages 142–157. IEEE, 2017. 4

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*, pages 1001–1017. ACM Press, October / November 2017. 2, 3, 4, 7, 27, 51

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, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 3–32. Springer, Cham, April / May 2017. 1, 2, 3, 4, 6, 7, 8, 11, 22, 23, 25, 26, 33, 42, 51, 52

ABP18. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Sustained space complexity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 99–130. Springer, Cham, April / May 2018. 3, 11

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*, pages 51–65. ACM Press, April 2018. 2

AS15. Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 595–603. ACM Press, June 2015. 2, 3, 11, 14

BCS16. Dan Boneh, Henry Corrigan-Gibbs, and Stuart E. Schechter. Balloon hashing: A memory-hard function providing provable protection against sequential attacks. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 220–248. Springer, Berlin, Heidelberg, December 2016. 15, 25

BDK16. Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. Argon2: new generation of memory-hard functions for password hashing and other applications. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 292–302. IEEE, 2016. 3, 15, 25

Ben89. Charles H. Bennett. Time/space trade-offs for reversible computation. *SIAM J. Comput.*, 18(4):766–776, aug 1989. 4, 5, 6, 8, 38, 47

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, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 573–607. Springer, Cham, August 2019. 2, 3, 33, 35, 38

BHL22. Jeremiah Blocki, Blake Holman, and Seunghoon Lee. The parallel reversible pebbling game: Analyzing the post-quantum security of iMHFs. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 52–79. Springer, Cham, November 2022. 1, 3, 4, 5, 7, 8, 9, 11, 12, 15, 16, 22, 23, 31, 32, 34, 35, 36, 38, 39, 46, 47

BLZ20. Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. Approximating cumulative pebbling cost is unique games hard. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 13:1–13:27. LIPIcs, January 2020. 1, 2, 8, 33, 34, 35, 37

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, Part I*, volume 10677 of *LNCS*, pages 445–465. Springer, Cham, November 2017. 2, 3, 4, 15, 25

BZ18. Jeremiah Blocki and Samson Zhou. On the computational complexity of minimal cumulative cost graph pebbling. In Sarah Meiklejohn and Kazue Sako, editors, *FC 2018*, volume 10957 of *LNCS*, pages 329–346. Springer, Berlin, Heidelberg, February / March 2018. 2

FA17. Michael P Frank and M Josephine Ammer. Relativized separation of reversible and irreversible space-time complexity classes. *arXiv preprint arXiv:1708.08480*, 2017. 4

FLW13. Christian Forler, Stefan Lucks, and Jakob Wenzel. Catena: A memory-consuming password-scrambling framework. *Cryptology ePrint Archive*, 2013. 14, 15, 25

GNP$^+$17. Paul Grassi, Elaine Newton, Ray Perlner, Andrew Regenscheid, William Burr, Justin Richer, Naomi Lefkovitz, Jamie Danker, Yee-Yin Choong, Kristen Greene, and Mary Theofanos. Digital identity guidelines: Authentication and lifecycle management, 2017-06-22 2017. 14

HPV77. John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *J. ACM*, 24(2):332–337, April 1977. 2

Kal00. Burt Kaliski. PKCS #5: Password-Based Cryptography Specification Version 2.0. RFC 2898, RSA Laboratories, September 2000. 3, 5, 13, 14

Kni95. Emanuel Knill. An analysis of bennett's pebble game. *arXiv preprint math/9508218*, 1995. 4, 8

Krá01. Richard Král'ovič. Time and space complexity of reversible pebbling. In Leszek Pacholski and Peter Ružička, editors, *SOFSEM 2001: Theory and Practice of Informatics*, pages 292–303, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. 8

KSS21. Niels Kornerup, Jonathan Sadun, and David Soloveichik. The spooky pebble game, 2021. 8

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*, STOC '79, page 262–277, New York, NY, USA, 1979. Association for Computing Machinery. 2

LT82. Thomas Lengauer and Robert E. Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *J. ACM*, 29(4):1087–1130, October 1982. 2

LV96. Ming Li and Paul Vitányi. Reversibility and adiabatic computation: Trading time and space for energy. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1947):769–789, Apr 1996. 38

MSR$^+$19. Giulia Meuli, Mathias Soeken, Martin Roetteler, Nikolaj Bjorner, and Giovanni De Micheli. Reversible pebbling game for quantum memory management. In *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 288–291, 2019. 8

PHC15. Password hashing competition, 2013–2015. 3, 14, 15

Pip77. Nicholas Pippenger. Superconcentrators. *SIAM Journal on Computing*, 6(2):298–304, 1977. 34, 35, 37

PM99. Niels Provos and David Mazières. A future-adaptive password scheme. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '99, page 32, USA, 1999. USENIX Association. 3, 5, 13

PTC76. Wolfgang J. Paul, Robert Endre Tarjan, and James R. Celoni. Space bounds for a game on graphs. In *Proceedings of the Eighth Annual ACM Symposium on Theory of Computing*, STOC '76, page 149–160, New York, NY, USA, 1976. Association for Computing Machinery. 2

Sve12. Ola Svensson. Hardness of vertex deletion and project scheduling. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX, and 16th International Workshop, RANDOM. Proceedings*, pages 301–312, 2012. 33

A Equivalence of Definitions of Reversible Pebbling

Recall the original reversible pebbling definition by Blocki et al. [BHL22] and our new definition.

Definition 7 (Reversible Graph Pebbling, [BHL22]). *Let $G = (V, E)$ be a DAG and let $T \subseteq V$ be a target set of nodes to be pebbled. A* pebbling configuration *(of G) at round i is a subset $P_i \subseteq V$. Let $P = (P_0, \ldots, P_t)$ be a sequence of pebbling configurations. Below are the following properties which define various aspects of reversible pebblings.*

(1) The pebbling should start with no pebbles ($P_0 = \emptyset$) and end with pebbles on all of the target nodes i.e., $T \subseteq P_t$.
(2) A pebble can be added only if all of its parents were pebbled at the end of the previous pebbling round, i.e., $\forall i \in [t] : x \in (P_i \setminus P_{i-1}) \Rightarrow \mathsf{parents}(x, G) \subseteq P_{i-1}$.
(3) (Quantum No-Deletion Property) *A pebble can be deleted only if all of its parents were pebbled at the end of the previous pebbling round, i.e., $\forall i \in [t] : x \in (P_{i-1} \setminus P_i) \Rightarrow \mathsf{parents}(x, G) \subseteq P_{i-1}$.*
(4) (Quantum Reversibility) *If a pebble was required to generate new pebbles (or remove pebbles), then we must keep the corresponding pebble around, i.e., $\forall i \in [t] : x \in \mathsf{parents}(P_i \setminus P_{i-1}, G) \cup \mathsf{parents}(P_{i-1} \setminus P_i, G) \Rightarrow x \in P_i$.*
(5) (Remove Excess Pebbles) *We also consider an optional constraint that $P_t = T$. If a pebbling does not satisfy this optional constraint we call it a relaxed pebbling.*
(6) (Sequential pebbling only) At most one pebble is added or removed in each round, i.e., $\forall i \in [t] : |(P_i \cup P_{i-1}) \setminus (P_i \cap P_{i-1})| \leq 1$.

Now we give pebbling definitions with respect to the above properties.

- *A* legal parallel reversible pebbling *of T is a sequence $P = (P_0, \ldots, P_t)$ of pebbling configurations of G where $P_0 = \varnothing$ and which satisfies conditions (1), (2), (3), (4) and (5) above. If our pebbling additionally satisfies condition (6) then we say that it is a sequential pebbling. Similarly, if our pebbling does not satisfy condition (5) then we call our pebbling strategy a* relaxed pebbling.
- *A* legal reversible pebbling sequence *is a sequence of pebbling configurations $(P_0, \ldots, P_t)$ which satisfies properties (2), (3), and (4) without requiring $P_0 = \{\}$.*

We denote $\mathcal{P}^{\leftrightarrow,\parallel}_{G,T}$ the set of all legal parallel reversible pebblings of G with a target set T, respectively. We denote with $\widetilde{\mathcal{P}}^{\leftrightarrow,\parallel}_{G,T}$ the set of all legal relaxed *parallel reversible pebblings of G with target set T. We will mostly be interested in the case where $T = \mathsf{sinks}(G)$ in which case we simply write $\mathcal{P}^{\leftrightarrow,\parallel}_{G}$ or $\widetilde{\mathcal{P}}^{\leftrightarrow,\parallel}_{G}$.*

We argue that Definition 7 and Definition 1 are indeed equivalent.

Lemma 7. *Definition 7 and Definition 1 are equivalent.*

Proof. Let $P = (P_0, \ldots, P_t)$ is a pebbling sequence of G and $P^* = (P_t, \ldots, P_0)$ is its reverse. Since condition (1) of Definition 7 and condition (1) of Definition 1 are identical, it is sufficient to prove that P satisfies conditions (2), (3), and (4) of Definition 7 if and only if condition (2) of Definition 1 holds.

($\Rightarrow$) Suppose that P satisfies conditions (2), (3), and (4) of Definition 7. We observe the following:

- Condition (2) of Definition 7 implies $\mathsf{parents}(P_i \setminus P_{i-1}, G) \subseteq P_{i-1}$ for all $i \in [t]$. Furthermore, condition (4) of Definition 7 implies $\mathsf{parents}(P_i \setminus P_{i-1}, G) \subseteq P_i$ for all $i \in [t]$. Taken together, we see that P is extra legal.
- Condition (3) of Definition 7 implies $\mathsf{parents}(P_{i-1} \setminus P_i, G) \subseteq P_{i-1}$ for all $i \in [t]$. Furthermore, condition (4) of Definition 7 also implies $\mathsf{parents}(P_{i-1} \setminus P_i, G) \subseteq P_i$ for all $i \in [t]$. Taken together, we see that P^* is extra legal.

Hence, we can conclude that condition (2) of Definition 1 holds.

($\Leftarrow$) Suppose that condition (2) of Definition 1 holds, i.e., P and P^* are both extra legal. Since P is extra legal, for all $i \in [t]$, we have

$$\mathsf{parents}(P_i \setminus P_{i-1}, G) \subseteq P_{i-1}, \tag{2}$$

and

$$\mathsf{parents}(P_i \setminus P_{i-1}, G) \subseteq P_i. \tag{3}$$

Similarly, since P^* is extra legal, for all $i \in [t]$, we have

$$\mathsf{parents}(P_{i-1} \setminus P_i, G) \subseteq P_i, \tag{4}$$

and

$$\mathsf{parents}(P_{i-1} \setminus P_i, G) \subseteq P_{i-1}. \tag{5}$$

Now, we can easily see that Equation (2) implies condition (2) of Definition 7 and Equation (5) implies condition (3) of Definition 7. Furthermore, combining Equation (3) and (4), we have that $\mathsf{parents}(P_i \setminus P_{i-1}, G) \cup \mathsf{parents}(P_{i-1} \setminus P_i, G) \subseteq P_i$ for all $i \in [t]$, which implies condition (4) of Definition 7. Hence, we can conclude that if condition (2) of Definition 1 holds, then conditions (2), (3), and (4) of Definition 7 hold. This completes the proof. □

B Approximation Hardness of Reversible CC

We begin by reviewing the approximation hardness of classical CC and subsequently delve into the challenges of extending this result to reversible CC in a black-box manner. We then provide a technique to overcome the challenge by extending the reversible pebbling strategy from previous work [BHL22].

B.1 Review: Approximation Hardness of Classical CC

Blocki et al. [BLZ20] showed that given a DAG G with constant indegree, it is Unique Games hard to approximate $\Pi_{cc}^{\parallel}(G)$ within any constant factor. Basically, the intuition is that the depth-robustness of G is both necessary [AB16] and sufficient [ABP17] condition for computing $\Pi_{cc}^{\parallel}(G)$ as the upper and lower bound of $\Pi_{cc}^{\parallel}(G)$ are given as follows: for any (e,d)-reducible DAG G with N nodes and indegree $\mathsf{indeg}(G)$, $\Pi_{cc}^{\parallel}(G) \leq \min_{g \geq d}(eN + gN \cdot \mathsf{indeg}(G) + N^2 d/g)$ [AB16], and for any (e,d)-depth robust DAG G, $\Pi_{cc}^{\parallel}(G) \geq ed$ [ABP17]. Then they showed that assuming that the Unique Games Conjecture is true, it is hard to distinguish between the cases where (1) G is (e_1, d_1)-reducible with $e_1 = N^{1/(1+2\varepsilon)}/k$ and $d_1 = kN^{2\varepsilon/(1+2\varepsilon)}$ (i.e., depth-reducible with relatively small e_1 and d_1), and (2) G is (d_2, e_2)-depth robust with $e_2 = (1-\varepsilon)N^{1/(1+2\varepsilon)}$ and $d_2 = 0.9N^{(1+\varepsilon)/(1+2\varepsilon)}$, for any constant $\varepsilon > 0$ (i.e., depth-robust with even large e_2 and d_2 when ε is small). The approximation hardness of $\Pi_{cc}^{\parallel}(G)$ can be proved by showing that there is a gap between the upper and lower bound of the classical pebbling complexity between the cases above.

To prove the argument, they presented the following technical ingredients:

(1) The first technical ingredient is *Svensson's result* [Sve12]. Svensson showed that it is Unique Games hard to distinguish between the cases where a (layered) DAG G with N nodes is (e_1, d_1)-reducible with $e_1 = N/k$ and $d_1 = k$ and G is (e_2, d_2)-depth robust with $e_2 = N(1 - 1/k)$ and $d_2 = \Omega(N^{1-\varepsilon})$. But scrutinizing further, Svensson's graph has high indegree, i.e., $\mathsf{indeg}(G) = \mathcal{O}(N)$, whereas we want to have constant indegree. Furthermore, we cannot directly apply Svensson's result to get the approximation hardness of $\Pi_{cc}^{\parallel}(G)$ as there is no gap between the upper and lower bound of $\Pi_{cc}^{\parallel}(G)$ when G is a Svensson's graph.

(2) Therefore, we need to reduce the indegree of the graph, but we also want to not lose the connectivity of Svensson's graph between each layer too much as we still want to have the Unique Games hardness result to distinguish between depth-reducible and depth-robust cases. This is where a γ-*extreme depth-robust graph* comes into play. A DAG G is said to be γ-extreme depth-robust if it is (e,d)-depth robust for any $e, d > 0$ such that $e + d \leq (1-\gamma)N$. By overlaying Svensson's graph on a γ-extreme depth-robust graph, i.e., only keeping edges from layer i to layer j in Svensson's graph if there is an edge from node i to j in the γ-extreme depth-robust graph, we can reduce the indegree from $\mathcal{O}(N)$ to $\mathcal{O}(N^{\varepsilon} \log^2 N)$. Furthermore, by applying indegree reduction gadget from Blocki et al. [ABP17], they proved that it is Unique Games hard to distinguish between the cases where a constant-indegree DAG G is (e_1, d_1)-reducible with $e_1 = N^{1/(1+2\varepsilon)}/k$ and $d_1 = kN^{2\varepsilon/(1+2\varepsilon)}$ and (e_2, d_2)-depth robust with $e_2 = (1-\varepsilon)N^{1/(1+2\varepsilon)}$ and $d_2 = 0.9N^{(1+\varepsilon)/(1+2\varepsilon)}$. However, there is still no gap between the classical pebbling complexity of the two cases.

(3) To remedy the no-gap situation above, they used the *superconcentrator overlay* that was introduced by Blocki et al. [BHK+19], which is a graph denoted

by $\mathsf{superconc}(G)$ that can be constructed by overlaying a DAG G with N nodes with a superconcentrator [Pip77] with N input/output nodes. It gives a stronger lower bound $\Pi_{cc}^{\|}(\mathsf{superconc}(G)) \geq \max\{eN, dN\}/8$ for CC and an improved pebbling strategy gives an improved upper bound, through which we can finally yield a gap between the upper and lower bound of the classical pebbling complexity of the superconcentrator overlay graph.

To summarize, Blocki et al. [BLZ20] made the worst-case analysis for the approximation hardness of the classical pebbling complexity by constructing a graph — the superconcentrator overlay of an indegree-reduced version (with γ-extreme depth-robust overlay) of Svensson's graph — that has a gap between the upper and lower bound of the classical pebbling complexity. The main result of the work can be presented as the following theorem.

Theorem 11 ([BLZ20]). *Given a DAG G with constant indegree, it is Unique Games hard to c-approximate $\Pi_{cc}^{\|}(G)$ for any constant $c > 1$.*

B.2 Computing Reversible CC is Also Unique Games Hard

A natural follow-up question is whether we can have the same approximation hardness result for *reversible* cumulative pebbling complexity. It is not a trivial black-box application of the prior work [BLZ20] since some of the pebbling strategies that were used in the prior analysis are inherently irreversible. For example, the improved strategy in Blocki et al. [BLZ20] when analyzing the upper bound of CC of the superconcentrator overlay graph, it runs multiple light and balloon phases [AB16]. At the end of each balloon phase, we discard all the unnecessary pebbles at once before running the next light phase, which is an irreversible pebbling transition.

Blocki et al. [BHL22] gave a reversible pebbling strategy which takes a light phase-balloon phase pebbling attack by Alwen and Blocki [AB16] and made it reversible. In particular, they showed the upper bound of $\Pi_{cc}^{\Longleftrightarrow,\|}(G)$ when G is (e, d)-reducible.

Theorem 12 ([BHL22, Theorem 4]). *For any (e, d)-reducible DAG G with N nodes,*

$$\Pi_{cc}^{\Longleftrightarrow,\|}(G) \leq \min_{g \geq d} \left\{ 2N \left(\frac{2Nd}{g} + e + 3g \right) + N + \frac{2Nd}{g} \right\}.$$

One might be tempted to adopt this strategy in a black-box manner and apply this upper bound with $\mathsf{superconc}(G)$ to create a gap between the upper and lower bound of $\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G))$. However, we cannot directly apply Theorem 12 to yield a gap between the upper and lower bound of $\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G))$. First, we observe that $\mathsf{superconc}(G)$ is $(e + N/d, 2d + 4\log N)$-reducible whenever G is (e, d)-reducible. This implies that $\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G)) = \mathcal{O}\left(N^{\frac{2+3\varepsilon}{1+2\varepsilon}}\right)$ when we apply Theorem 12 with an (e', d')-reducible DAG $\mathsf{superconc}(G)$ where

$e' = e + N/d, d' = 2d + 4\log N$ and $e = \frac{1}{k}N^{\frac{1}{1+2\varepsilon}}, d = kN^{\frac{2\varepsilon}{1+2\varepsilon}}$. If we apply the lower bound $\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G)) \geq \min\left\{\frac{e'N}{8}, \frac{d'N}{8}\right\}$ [BHK+19, Theorem 9] as the same lower bound carries over to the reversible CC, we have that $\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G)) \geq \Omega(N^{\frac{2+2\varepsilon}{1+2\varepsilon}})$, which implies that there is no gap between the upper and lower bound of $\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G))$.

Therefore, we should open the black box and update the improved pebbling strategy from the prior work [BLZ20]. This can be done by substituting the classical pebbling strategy [AB16] to pebble all the input nodes with the reversible one [BLZ20]. We remark that this replacement would additionally require updating the light and balloon phases accordingly.

Lemma 8 ([Pip77]). *There exists a superconcentrator G with at most $7N$ vertices, containing N input vertices and N output vertices, such that* $\mathsf{indeg}(G) \leq 9$ *and* $\mathsf{depth}(G) \leq 4\log N$.

Lemma 9. *Let G be an (e,d)-reducible DAG with N nodes with* $\mathsf{indeg}(G) = 2$. *Then*

$$\Pi_{cc}^{\Longleftrightarrow,\|}(\mathsf{superconc}(G)) \leq \min_{g \geq d}\left\{3eN + 13gN + \frac{(25d+1)N^2}{g} + \frac{2Nd}{g} + 28N\log N \right.$$
$$\left. + \frac{84N^2\log N}{g} + 2N\right\}.$$

Proof. We give a *reversible* pebbling strategy for the superconcentrator overlay graph $G' = \mathsf{superconc}(G)$:

Reversible Pebbling Strategy for $G' = \mathsf{superconc}(G)$:

1. Pebble all the input nodes $\mathsf{input}(G') = G$ using the reversible pebbling strategy from Blocki et al. [BHL22].
2. Efficiently pebble $\mathsf{interior}(G')$ using the property of superconcentrator, i.e., $\mathsf{superconc}(G)$ with N input/output nodes has depth at most $4\log N$. At the end of Step 2, remove pebbles by running a reversible monotonic pebbling sequence to the precondition for each light phase.
3. Pebble all nodes in $\mathsf{output}(G')$ by alternating between light and balloon phases.
   - **Light Phase:** Walk pebble across the interval $I_i = [o_{(i-1)g+1}, o_{ig}]$ in $\mathcal{O}(g)$ steps.
     - Precondition: pebbles on $\mathsf{parents}(o_{(i-1)g+1}) \cup (\mathsf{parents}(I_i) \setminus I_i) \cup S_{\leq o_{(i-1)g}}$
     - Postcondition: pebbles on $\{o_{ig}\} \cup S$
   - **Balloon Phase:** Recover all the missing pebbles in $\mathsf{input}(G') \cup \mathsf{interior}(G')$ for the upcoming light phase.
     - Precondition: pebbles on $\{o_{ig+1}\} \cup S$
     - Midcondition: pebbles on $\{o_{ig+1}\} \cup \mathsf{input}(G') \cup \mathsf{interior}(G')$
     - Postcondition: pebbles on $\mathsf{parents}(o_{ig+1}) \cup (\mathsf{parents}(I_{i+1}) \setminus I_{i+1}) \cup S$

Analysis. We will examine the cumulative pebbling complexity of $G' = \mathsf{superconc}(G)$ for each step above.

1. We need to pebble all the input nodes $\mathsf{input}(G') = G$ using the reversible pebbling strategy from Blocki et al. [BHL22], which will be upper bounded by
$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) \leq \min_{g \geq d} \left\{ 2N\left(\frac{2Nd}{g} + e + 3g\right) + N + \frac{2Nd}{g} \right\},$$
followed by [BHL22, Theorem 4]. We remark that the difference here is that while [BHL22, Theorem 4] denotes the reversible pebbling cost to pebble the last node of G only, we need to pebble all nodes in G. However, we observe that we can recover pebbles on all nodes by running one extra balloon phase concurrently and such cost is already contained in $4N^2d/g + N + 2Nd/g$. Hence, we have the same upper bound with $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)$.
2. When we start with having pebbles on all nodes in $\mathsf{input}(G') = G$, since $\mathsf{superconc}(G)$ has depth at most $4 \log N$, we can pebble all nodes in $\mathsf{interior}(G')$ with CC at most $7N \cdot 4 \log N = 28N \log N$. Next, we would need to remove pebbles by running a reversible monotonic pebbling sequence to the precondition for each light phase. However, we can observe that the CC of this procedure is exactly the same as the CC of pebbling rounds starting from the precondition of each light phase to $\mathsf{input}(G') \cup \mathsf{interior}(G')$. This is contained in running one extra balloon phase (from midcondition to postcondition), which is going to be at most $(d + 4 \log N)7N \cdot N/g$ by the analysis of Step 3 below.
3. In this step, we would like to walk a pebble across the output nodes from o_1 to o_N. To save cost during this step, we should alternate light phases and balloon phases repeatedly N/g times in total as we split the output nodes into intervals $I_i = \left[o_{(i-1)g+1}, o_{ig}\right]$ of size g each. Let S be a (e, d)-depth-reducing set for G. In each light phase, to walk a pebble across the interval I_i, we would need to keep pebbles on S and $\mathsf{parents}(I_i) \setminus I_i$. Since each node in I_i has at most 7 parents and we keep one pebble in I_i (the current node) for each step, the maximum number of pebbles to keep would be $|S| + 7g + 1 + N/g = e + 7g + 1 + N/g$ for each step. So far, the maximum pebbling cost to reach the last node in I_i is $(e + 7g + 1)g + N$. After placing a pebble on the last node o_{ig} in I_i, we would need to discard unnecessary pebbles and prepare for the next light phase as well by running a balloon phase. Since S is a (e, d)-depth-reducing set, we have that $\mathsf{depth}(G' \setminus (S \cup \mathsf{output}(G'))) \leq d + 4 \log N$ (see Figure 2). Hence, for each balloon phase, we have reversible pebbling cost at most $(d + 4 \log N)7N$. Since we need to run balloon phase twice in each block, the total reversible pebbling cost for Step 3 will be at most $\left[(e + 7g + 1)g + N + 2(d + 4 \log N)7N\right] \frac{N}{g}$.

Taken together, we have
$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(G') \leq \min_{g \geq d} \left\{ 2N\left(\frac{2Nd}{g} + e + 3g\right) + N + \frac{2Nd}{g} + 28N \log N \right.$$

$$+ \left[(e + 7g + 1)g + N + 3(d + 4\log N)7N\right]\frac{N}{g}\Bigg\}$$

$$\leq \min_{g \geq d}\left\{3eN + 13gN + \frac{(25d+1)N^2}{g} + \frac{2Nd}{g} + 28N\log N + \frac{84N^2\log N}{g} + 2N\right\},$$

as desired. □

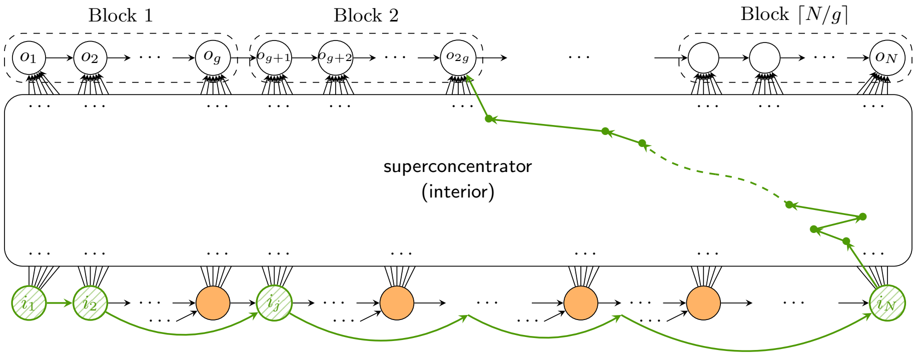


Fig. 2: A reversible pebbling strategy for a superconcentrator overlay $G' = \mathsf{superconc}(G)$. By definition, we have $\mathsf{input}(G') = G$, and all the output nodes $o_1, \ldots, o_N$ are overlayed by a line graph. We note that each input node has outdegree 6 that is connected to the interior of the superconcentrator, and each output node has indegree at most 7 (six from the interior and one from the prior output node) due to the superconcentrator construction by Pippenger [Pip77]. Here, orange nodes in the input nodes denote the depth-reducing set S of $G = \mathsf{input}(G')$. Then since we have that the depth of the superconcentrator is at most $4\log N$ and the graph G is (e,d)-depth reducible, we observe that $\mathsf{depth}(G' \setminus (S \cup \mathsf{output}(G'))) \leq d + 4\log N$, which is illustrated by a green path above.

Theorem 13. *Given a DAG G with constant indegree, it is Unique Games hard to approximate $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)$ within any constant factor.*

Proof. Let $k \geq 2$ be an integer that we shall later fix and $\varepsilon > 0$ be a constant that we will later fix as well. Given a DAG G with N nodes, we know that it is Unique Games hard to distinguish between two cases where (1) G is (e_1, d_1)-reducible for $e_1 = \frac{1}{k}N^{\frac{1}{1+2\varepsilon}}$ and $d_1 = kN^{\frac{2\varepsilon}{1+2\varepsilon}}$, and (2) G is (e_2, d_2)-depth robust for $e_2 = (1-\varepsilon)N^{\frac{1}{1+2\varepsilon}}$ and $d_2 = 0.9N^{\frac{1+\varepsilon}{1+2\varepsilon}}$ [BLZ20]. If G is (e_1, d_1)-reducible, then by Lemma 9, for $e_1 = \frac{1}{k}N^{\frac{1}{1+2\varepsilon}}, d_1 = kN^{\frac{2\varepsilon}{1+2\varepsilon}}$, and $g = e_1$, we have

$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{superconc}(G)) \leq \min_{g \geq d}\left\{3e_1N + 13gN + \frac{(25d_1+1)N^2}{g} + \frac{2Nd_1}{g} + 28N\log N\right.$$

$$
\begin{aligned}
&\qquad + \frac{84N^2 \log N}{g} + 2N \Big\} \\
&\le 16e_1N + \underbrace{\frac{(25d_1+1)N^2}{e_1} + \frac{2Nd_1}{e_1} + 28N\log N + \frac{84N^2\log N}{e_1} + 2N}_{\ll e_1 N} \\
&\le 17e_1N = \frac{17}{k} N^{\frac{2+2\varepsilon}{1+2\varepsilon}}.
\end{aligned}
$$

On the other hand, if G is (e_2, d_2)-depth robust, then we have

$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{superconc}(G)) \ge \min\left\{\frac{e_2N}{8}, \frac{d_2N}{8}\right\},$$

by [BHK+19, Theorem 9]. We remark that since $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) \ge \Pi_{cc}^{\parallel}(G)$ for any DAG G, the same lower bound for the superconcentrator overlay carries over to the reversible setting. In particular, since $e_2 \ll d_2$, we have

$$\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{superconc}(G)) \ge \frac{e_2N}{8} = \frac{1-\varepsilon}{8} N^{\frac{2+2\varepsilon}{1+2\varepsilon}}.$$

Let $c > 1$ be any constant. Setting $\varepsilon = 0.1$ and $k = \lceil \frac{1360}{9} c^2 \rceil$, we get that if G is (e_1, d_1)-reducible, then $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{superconc}(G)) \le \frac{9}{80c^2} N^{\frac{2+2\varepsilon}{1+2\varepsilon}}$ but if G is (e_2, d_2)-depth robust, then $\Pi_{cc}^{\Longleftrightarrow,\parallel}(\mathsf{superconc}(G)) \ge \frac{9}{80} N^{\frac{2+2\varepsilon}{1+2\varepsilon}}$. Hence, it is Unique Games hard to approximate $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G)$ with a factor of c. □

C Pebbling Composition

Bennett [Ben89] gave the following reversible pebbling strategy, whose analysis was improved by Li and Vitányi [LV96]. For a line graph on nodes $\left[2^k - 1\right]$, Bennett define the intervals I_j and nodes i_j such that $I_0 = \langle\rangle$ and

$$I_k = \langle I_{k-1}, i_{k-1}, \dots, I_0, i_0 \rangle .$$

Intuitively, the nodes in the recursive list I_k partitions $\left[2^k - 1\right]$, and the nodes appear from least to greatest. The interval I_1 contains 1 node and the interval I_k contains twice the nodes of I_{k-1}, with one additional node i_{k-1}. That is, $N(k) = 2N(k-1) + 1$. Blocki, Holman, and Lee [BHL22] improved this pebbling by lowering the time cost at the expense of space. For a tunable parameter c, they pebble the line graph with $N(c,k) = \Theta\left((c+1)^k\right)$, by letting $I_j^c = \left\langle I_j^{(1)}, i_j^{(1)} \dots, I_j^{(c)}, i_j^{(c)} \right\rangle$, where each $I_j^{(\ell)}$ is a copy of I_j. Finally,

$$I'_k = \langle I'_{k-1}, i'_{k-1}, \dots, I'_0, i'_0 \rangle ,$$

where each i'_j is a single node, and the elements of $[N(c,k)]$ occur in increasing order.

The pebbling P_c^k on the line graph $\mathcal{L}_{N(c,k)}$ is defined as follows:

(1) For $j = k-1, \ldots, 1$:
   (a) Pebble $I_j^{(1)}$ via P_1^j.
   (b) Place a pebble on $i_{k-1}^{(1)}$.
   (c) For $\ell = 2, \ldots, c$:
      i. Unpebble $I_{k-1}^{\ell-1}$ by reversing P_1^j.
      ii. Pebble I^ℓ via P_1^j.
      iii. Place a pebble on i_j^ℓ.

The end state of this pebbling has a pebble on node $N(c, k)$, and we can run it in reverse to remove all pebbles. Choosing $k = \sqrt{\log N}$ and $c = 2^k$ leads to Theorem 5.

Theorem 5 (Reversible Line Graph Pebbling [BHL22]). *There exist a family of sequential pebblings L_N and a family of parallel reversible pebblings $L_N^{\|}$ for line graphs $\mathcal{L}_N$ such that*

(1) $\Pi_t\left(L_N\right) = \mathcal{O}\left(N^{1+\frac{1}{\sqrt{\log N}}}\right)$, $\Pi_s\left(L_N\right) = \mathcal{O}\left(N^{\frac{1}{\sqrt{\log N}}}\sqrt{\log N}\right)$, $\Pi_{st}\left(L_N\right), \Pi_{cc}\left(L_N\right) = \mathcal{O}\left(N^{1+\frac{2}{\sqrt{\log N}}}\sqrt{\log N}\right)$, *and* $\mathsf{toggle}(L_N) = \mathcal{O}\left(N^{\frac{1}{\sqrt{\log N}}}\right)$, *and*

(2) $\Pi_t\left(L_N^{\|}\right) = \mathcal{O}\left(N\right)$, $\Pi_s\left(L_N^{\|}\right) = \mathcal{O}\left(N^{\frac{2}{\sqrt{\log N}}}\right)$, $\Pi_{st}\left(L_N^{\|}\right), \Pi_{cc}\left(L_N^{\|}\right) = \mathcal{O}\left(N^{1+\frac{2}{\sqrt{\log N}}}\right)$, *and* $\mathsf{toggle}(L_N^{\|}) = \mathcal{O}\left(N^{\frac{1}{\sqrt{\log N}}}\right)$.

Proof. Let $k = \sqrt{\log N}$ and $c = 2^k$. We'll prove the claims unproven in [BHL22]:

- **(Time Complexity)** $\Pi_t\left(L_N^{\|}\right) = O\left(N\right)$: [BHL22] show that $\Pi_t\left(L_N^{\|}\right) = O\left((c+2)^k\right) = O\left((c+2)^k\right) = O(N)$.
- **(Toggle Number)** Notice that if we pebble or unpebble any node at most t times in I'_j, then we pebble or unpebble any node in I'_{j+1} at most $2t$ times. The nodes in I'_0 are pebbled once and unpebbled once, so $\mathsf{toggle}(L_N) \leq 2^{k+1}$.

□

Theorem 2 (Classical vs. Reversible Space-Time Complexity). *Let $G = (V = [N], E)$ be a DAG. Then*

$$\Pi_{st}^{\Longleftrightarrow,\|}\left(G\right) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right) \cdot \Pi_{st}^{\|}\left(G\right),$$

and

$$\Pi_{st}^{\Longleftrightarrow}\left(G\right) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\sqrt{\log N}\right) \cdot \Pi_{st}\left(G\right).$$

Proof of Theorem 2. If $P = (P_1, \ldots, P_t)$ is a pebbling of G and $L = (L_1, \ldots, L_{t'})$ of $\mathcal{L}_t$ and $Q = L \circ P$ is composed pebbling derived as in Theorem 4 then by Theorem 4 we have $\Pi_{st}(Q) = \Pi_s(P) \cdot \Pi_{st}(L) = \Pi_{st}(P) \cdot \Pi_{st}(L)/t$.

If $P = (P_1, \ldots, P_t)$ is the parallel pebbling of G with minimum space-time cost (i.e., $\Pi_{st}(P) = \Pi_{st}^{\|}(G)$) then $\Pi_{st}(Q) = \Pi_{st}^{\|}(G) \cdot \Pi_{st}(L)/t$. Taking

$L = (L_1, \ldots, L_{t'})$ to be the parallel pebbling of $\mathcal{L}_t$ from Theorem 5 we have $\mathit{\Pi}_{st}(L)/t = \mathcal{O}\left(t^{\frac{2}{\sqrt{\log t}}}\right)$. Using the fact that $t \leq N^2$ (otherwise we would have $\mathit{\Pi}_{st}(P) > N^2$ and P would not be optimal) we have $\mathit{\Pi}_{st}(L)/t = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right)$. Hence, $\mathit{\Pi}_{st}^{\Longleftrightarrow,\parallel}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\right) \cdot \mathit{\Pi}_{st}^{\parallel}(G)$.

Similarly, let $P = (P_1, \ldots, P_t)$ be the special sequential pebbling of G with minimum space-time cost — recall that a special sequential pebbling only removes at most one pebble per round (and never removes pebbles during a round when pebbles are added). By Theorem 14 we have $\mathit{\Pi}_{st}(P) \leq 6\mathit{\Pi}_{st}(G)$ since we can transform any sequential pebbling of G into a special sequential one whilst increasing space-time costs by a multiplicative factor of 6 *at most*. Optimality of P implies that $t \leq N^2$ since there the naive pebbling strategy for G is special sequential and has space-time cost *at most* N^2. Because P is special sequential we can apply Corollary 1 to argue that the composed pebbling $Q = L \circ P$ derived as in Theorem 4 is sequential as long the reversible pebbling L of $\mathcal{L}_t$ is sequential. We use the sequential pebbling $L = (L_1, \ldots, L_{t'})$ of $\mathcal{L}_t$ as defined in Theorem 5. Now the composed pebbling $Q = L \circ P$ is sequential and we have $\mathit{\Pi}_{st}(Q) = \mathit{\Pi}_{st}(G) \cdot \mathit{\Pi}_{st}(L)/t$ with $\mathit{\Pi}_{st}(L)/t = \mathcal{O}\left(t^{\frac{2}{\sqrt{\log t}}}\sqrt{\log t}\right)$. Using the fact that $t \leq N^2$ we have $\mathit{\Pi}_{st}(L)/t = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\sqrt{\log N}\right)$ and $\mathit{\Pi}_{st}(Q) = \mathit{\Pi}_{st}(P) \times \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\sqrt{\log N}\right)$. Hence, $\mathit{\Pi}_{st}^{\Longleftrightarrow}(G) = \mathcal{O}\left(N^{\frac{2\sqrt{2}}{\sqrt{\log N}}}\sqrt{\log N}\right) \cdot \mathit{\Pi}_{st}(G)$. □

Theorem 4 (Reversible Composition Pebbling). *Let $P = (P_1, \ldots, P_t)$ be a (possibly irreversible) pebbling for a DAG G, and $L = (L_1, \ldots, L_{t'})$ be a reversible pebbling for $\mathcal{L}_t$. Then the composition $L \circ P$ is a legal reversible pebbling of G satisfying $\mathit{\Pi}_{st}(Q) \leq \mathit{\Pi}_s(P) \cdot \mathit{\Pi}_{st}(L)$.*

Proof. Consider the transition between two configurations Q_i and Q_{i+1}:

- **(Property 1, Empty Start)** This follows from the fact that L and P start with out any pebbles on the graph.
- **(Property 2, Reversible)** We want to show that both Q and its reverse Q^* are extra legal. First, to show that Q is extra legal, we need to show that $\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq Q_i$ and $\mathsf{parents}(Q_{i+1} \setminus Q_i, G) \subseteq Q_{i+1}$. We first observe that

$$\mathsf{parents}(Q_{i+1} \setminus Q_i, G) = \mathsf{parents}\left(\bigcup_{j \in L_{i+1}} P_j \setminus \bigcup_{k \in L_i} P_k, G\right) \tag{6}$$

$$\subseteq \mathsf{parents}\left(\bigcup_{j \in L_{i+1} \setminus L_i} P_j \setminus P_{j-1}, G\right) \tag{7}$$

$$\subseteq \bigcup_{j \in L_{i+1} \setminus L_i} \mathsf{parents}\left(P_j \setminus P_{j-1}, G\right) \tag{8}$$

$$\subseteq \bigcup_{j \in L_{i+1} \setminus L_i} P_{j-1} \tag{9}$$

$$= \bigcup_{k \in \mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t)} P_k. \tag{10}$$

Here, Eq. (10) follows since if $j \in L_{i+1} \setminus L_i$ then we observe that $j - 1 \in \mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t)$ since $\mathcal{L}_t$ is a line graph. Now, from Eq. (10), we obtain

$$\begin{aligned}
\mathsf{parents}(Q_{i+1} \setminus Q_i, G) &\subseteq \bigcup_{k \in \mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t)} P_k \\
&\subseteq \bigcup_{k \in L_i} P_k \qquad (11) \\
&= Q_i, \qquad (12)
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{parents}(Q_{i+1} \setminus Q_i, G) &\subseteq \bigcup_{k \in \mathsf{parents}(L_{i+1} \setminus L_i, \mathcal{L}_t)} P_k \\
&\subseteq \bigcup_{k \in L_{i+1}} P_k \qquad (13) \\
&= Q_{i+1}, \qquad (14)
\end{aligned}$$

which shows that Q is extra legal. Here, Eq. (11) and Eq. (13) follows by the fact that L is extra legal. Similarly, to show that Q^* is extra legal, we need to show that $\mathsf{parents}\,(Q_i \setminus Q_{i+1}, G) \subseteq Q_i$ and $\mathsf{parents}\,(Q_i \setminus Q_{i+1}, G) \subseteq Q_{i+1}$. We observe that

$$\begin{aligned}
\mathsf{parents}\,(Q_i \setminus Q_{i+1}, G) &= \mathsf{parents}\left(\bigcup_{j \in L_i} P_j \setminus \bigcup_{k \in L_{i+1}} P_k, G \right) \qquad (15) \\
&= \mathsf{parents}\left(\bigcup_{j \in L_i \setminus L_{i+1}} P_j \setminus \bigcup_{k \in L_{i+1}} P_k, G \right) \qquad (16) \\
&\subseteq \mathsf{parents}\left(\bigcup_{j \in L_i \setminus L_{i+1}} P_j \setminus P_{j-1}, G \right) \qquad (17) \\
&= \bigcup_{j \in L_i \setminus L_{i+1}} \mathsf{parents}\,(P_j \setminus P_{j-1}, G) \qquad (18) \\
&\subseteq \bigcup_{j \in L_i \setminus L_{i+1}} P_{j-1} \qquad (19) \\
&= \bigcup_{k \in \mathsf{parents}(L_i \setminus L_{i+1}, \mathcal{L}_t)} P_k. \qquad (20)
\end{aligned}$$

In Eq. (16) we see that for any $i \in L_i$, if i is also in L_{i+1}, then $P_j \setminus \bigcup_{k \in L_{i+1}} P_k = \emptyset$. Eq. (17) follows from the fact that L is reversible, so if j was deleted on step $i+1$, then the parents of j (which is just $j-1$) must be kept around on step $i+1$. Eq. (19) follows since P is a legal pebbling. Since $\mathcal{L}_t$ is a line graph, $\mathsf{parents}(\{j\}, \mathcal{L}_t) = \{j-1\}$. Now, from Eq. (20), we obtain

$$\begin{aligned}
\mathsf{parents}\left(Q_i \setminus Q_{i+1}, G\right) &\subseteq \bigcup_{k \in \mathsf{parents}(L_i \setminus L_{i+1}, \mathcal{L}_t)} P_k \\
&\subseteq \bigcup_{k \in L_i} P_k && (21) \\
&= Q_i, && (22)
\end{aligned}$$

and

$$\begin{aligned}
\mathsf{parents}\left(Q_i \setminus Q_{i+1}, G\right) &\subseteq \bigcup_{k \in \mathsf{parents}(L_i \setminus L_{i+1}, \mathcal{L}_t)} P_k \\
&\subseteq \bigcup_{k \in L_{i+1}} P_k && (23) \\
&= Q_{i+1}, && (24)
\end{aligned}$$

which shows that Q^* is extra legal. Here, Eq. (21) and Eq. (23) follows by the fact that L^* is extra legal.

- **(Property 3, Remove Excess Pebbles)** Since $L_{t'} = \{t\}$, $Q_{t'} = P_t = \mathsf{sinks}(G)$.

Now we examine the space-time cost of Q. We have

$$\Pi_s(Q) \leq \Pi_s(P) \cdot \Pi_s(L)$$

since Q has pebbles on at most $\Pi_s(L)$ pebbling configurations of P, each of which have space at most $\Pi_s(P)$. Since $\Pi_t(Q) = \Pi_t(L)$, we have

$$\Pi_{st}(Q) = \Pi_s(P) \cdot \Pi_s(L) \cdot \Pi_t(L). \qquad \square$$

Lemma 5. *Let G be an f-reducible DAG of depth on N nodes then if $f(d) = \widetilde{\mathcal{O}}\left(\frac{N}{d^b}\right)$ for some constant $0 < b \leq 2/3$ and let $a = \frac{1-2b+\sqrt{1+4b^2}}{2}$. Then for any constant $\varepsilon > 0$, $\Pi_{cc}^{\Longleftrightarrow,\parallel}(G) \leq \mathcal{O}\left(\delta N^{1+a+\epsilon}\right)$.*

Proof. Let $d_0 = \mathsf{depth}(G)$. Alwen et al. [ABP17] show that such a graph is (e_i, d_i) reducible for $e_i = N^{a_i + \varepsilon/3}$ with depth-reducing sets S_i of size e_i and $d_i \leq N^{\frac{1-a_i}{b}}$ for each $i > 0$. They also observe that $d_{i+1} N \leq e_{i+1} d_i / 2$ for all $i > 1$, and for any ε, there exists a constant k such that $d_k \leq N^{\varepsilon/3}$. Let

$$C_i = \max_{|T'| \leq \delta e_1} \Pi_{cc}^{\Longleftrightarrow,\parallel}\left(G - S_i, T', 2d_1\right).$$

We can now apply Theorem 8 recursively. Then we have

$$\begin{aligned}\Pi_{cc}^{\Longleftrightarrow,\|}(G,\{N\},4d_0) &\le 4k(\delta+2)N^{1+a+\varepsilon/2}+4d_0+N^{a+\varepsilon/3}C_i\\ &\le 4k(\delta+2)N^{1+a+\varepsilon/2}+4d_0+N^{a+\varepsilon/3}\left(2Nd_k\right)\\ &\le 4k(\delta+2)N^{1+a+\varepsilon/2}+4d_0+N^{a+\varepsilon/3}\left(N^{1+\varepsilon/3}\right)\\ &= O\left(\delta N^{1+a+\varepsilon}\right).\end{aligned}$$

□

Theorem 14. *Suppose that P is a sequential pebbling of a DAG G then there is another sequential pebbling P' of G such that*

(1) (Special Sequential) $\left|P'_i \cup P'_{i-1} \setminus (P'_i \cap P'_{i-1})\right| \le 1$ for all round i,
(2) (Similar ST Cost) $\Pi_{st}(P') \le 6\Pi_{st}(P)$,
(3) (Similar Time) $\Pi_t(P') \le 4\Pi_t(P)$, and
(4) (Similar CC) $\Pi_{cc}(P') \le 6\Pi_{cc}(P)$.

Proof. Let $P=(P_1,\ldots,P_t)$ be a sequential pebbling of G. We first define an intermediate pebbling $A=(A_1,\ldots,A_{2t})$ with $A_{2i-1}=P_i$ for each i and $A_{2i}=P_i\cup(P_{i+1}\setminus P_{i-1})$ for each $i\le t$. Legality of the transition from A_{2i-1} to A_{2i} immediately follows from the legality of the transition P_i to P_{i+1}. Observe that $A_{2i}\supseteq A_{2i-1}=P_i$ and $A_{2i+1}=P_{i+1}\subseteq A_{2i}$. Thus each transition in A can either add one pebble or remove pebbles but not both (it is possible that we neither add nor remove a pebble so that $A_{2i-1}=A_{2i}$). We have $\Pi_s(A)\le\Pi_s(P)+1$ and $\Pi_t(A)\le 2\Pi_t(P)$ so trivially $\Pi_{st}(A)\le 3\Pi_{st}(P)$. We also have $|A_{2i}|\le|P_{i+1}|+1$ so $\Pi_{cc}(A)=\sum_j|A_j|=\sum_{i\le t}(|A_{2i-1}|+|A_{2i}|)\le\sum_{i\le t}(|P_i|+|P_{i+1}|+1)\le 3\Pi_{cc}(P)$.

Now let $a_1<\ldots<a_k$ denote all of the rounds where pebbles are removed and let $r_i=|A_{a_i-1}|-|A_{a_i}|>0$ denote the number of pebbles that are removed during round a_i. We can transform A into a new pebbling P' by replacing each transition $A_{a_i-1}\to A_{a_i}$ with r_i transitions so that we delete at most one pebble at a time. For example, if $A_{a_i-1}\setminus A_{a_i}=\{v_1,\ldots,v_{r_i}\}$ then we can define $A^j_{a_i-1}=A_{a_i-1}\setminus\{v_1,\ldots,v_j\}$ for each $j\le r_j$ so that $A^0_{a_i-1}=A_{a_i-1}$ and $A^{r_j}_{a_i-1}=A_{a_i}$. This transformation adds $\sum_{j=1}^k r_j$ pebbling rounds in total. However, we must have $\Pi_t(A)\ge\sum_{j=1}^k r_j$ because the total number of rounds where we add a pebble must be greater than the total number of pebbles that are removed since the pebbling A is sequential. Thus, $\Pi_t(P')\le\Pi_t(A)+\sum_{j=1}^k r_j\le 2\Pi_t(A)$ and $\Pi_s(P')\le\Pi_s(A)$. It follows that $\Pi_{st}(P')\le 2\Pi_{st}(A)\le 6\Pi_{st}(P)$.

Similarly, we have $\Pi_{cc}(P')-\Pi_{cc}(A)=\sum_{i=1}^k\sum_{j=|A_{a_i}|}^{|A_{a_i}|+r_i} j$ where we can argue that $\sum_{i=1}^k\sum_{j=|A_{a_i}|}^{|A_{a_i}|+r_i} j\le\Pi_{cc}(A)$ — intuitively if we pay cost $\sum_{j=|A_{a_i}|}^{|A_{a_i}|+r_i} j$ to reduce space usage from $|A_{a_i}|+r_i$ down to $|A_{a_i}|$ then we must have previously payed the equivalent cost to increase space from $|A_{a_i}|$ up to $|A_{a_i}|+r_i$. It follows that $\Pi_{cc}(P')\le 2\Pi_{cc}(A)\le 6\Pi_{cc}(P)$. □

Corollary 1. *If $P=(P_1,\ldots,P_t)$ is a special sequential pebbling of a DAG G and L is a reversible sequential pebbling of $\mathcal{L}_t$, then $L\circ P$ is a reversible sequential pebbling of G.*

Proof. Let $Q = L \circ P$. We have already proved that Q is a legal reversible pebbling. It remains to prove that Q is sequential. We have

$$\begin{aligned}
|Q_{i+1} \setminus Q_i| &= \left| \bigcup_{j \in L_{i+1}} P_j \setminus \bigcup_{k \in L_i} P_k \right| \\
&= \left| \bigcup_{j \in L_{i+1} \setminus L_i} P_j \setminus \bigcup_{k \in L_i} P_k \right| \\
&\leq \left| \bigcup_{j \in L_{i+1} \setminus L_i} P_j \setminus P_{j-1} \right| \\
&\leq 1,
\end{aligned}$$

since $|L_{i+1} \setminus L_i| \leq 1$. Furthermore, since P is a special sequential pebbling, we have $|P_j \setminus P_{j+1}| \leq 1$ for all j. Hence,

$$\begin{aligned}
|Q_i \setminus Q_{i+1}| &= \left| \bigcup_{j \in L_i} P_j \setminus \bigcup_{k \in L_{i+1}} P_k \right| \\
&= \left| \bigcup_{j \in L_i \setminus L_{i+1}} P_j \setminus \bigcup_{k \in L_{i+1}} P_k \right| \\
&\leq \left| \bigcup_{j \in L_i \setminus L_{i+1}} P_j \setminus P_{j+1} \right| \\
&\leq 1,
\end{aligned}$$

since $|L_i \setminus L_{i+1}| \leq 1$. Since L is a reversible sequential pebbling, we observe that it must be the case either $|L_{i+1} \setminus L_i| = 0$ or $|L_i \setminus L_{i+1}| = 0$. If $|L_{i+1} \setminus L_i| = 0$ then we have $|Q_{i+1} \setminus Q_i| = 0$, and if $|L_i \setminus L_{i+1}| = 0$ then we have $|Q_i \setminus Q_{i+1}| = 0$. This completes the proof. □

Lemma 2. *Define functions h, f, and g such that for any $0 < c < \sqrt{2}$, $h(N) = 2^{c\sqrt{\log N}}$, $f(N) = N \cdot h(N)$, and $g(N) = 2f\left(\frac{N}{h(N)}\right) + f\left(N - \frac{N}{h(N)}\right)$. There exists $N_0 \geq 1$ such that $f(N) \leq g(N)$ for all $N \geq N_0$.*

Proof of Lemma 2. Let $h(N) = 2^{c\sqrt{\log N}}$, so $f(N) = N \cdot h(N)$ and $g(N) = 2f(N/h(N)) + f(N/h(N))$. It suffices to show

$$\begin{aligned}
&\lim_{N\to\infty} g(N) - f(N) \\
&= \lim_{N\to\infty} N\left(h\left(N - \frac{N}{h(N)}\right) - h(N)\right) + \frac{N}{h(N)}\left(2h\left(\frac{N}{h(N)}\right) - h\left(N - \frac{N}{h(N)}\right)\right) \\
&= \infty.
\end{aligned}$$

In particular, we show that $h(N) - h(N - N/h(N)) = o(1)$ and $2h(N/h(N)) - h(N - N/h(N)) = \Omega(1)$ for all $0 < c < \sqrt{2}$. First, we have

$$\begin{aligned}
\lim_{N \to \infty} \sqrt{\log N} - \sqrt{\log N/h(N)} &= \lim_{N \to \infty} \sqrt{\log N} - \sqrt{\log N - c\sqrt{\log N}} \\
&= \lim_{x \to \infty} \sqrt{x} - \sqrt{x - c\sqrt{x}} \\
&= \lim_{x \to \infty} \frac{c\sqrt{x}}{\sqrt{x} + \sqrt{x - c\sqrt{x}}} \\
&= \frac{c}{2}. \qquad \triangleleft \text{ since } c = O(1)
\end{aligned}$$

Thus, $h(N/h(N))/h(N) \geq 2^{-\frac{c^2}{2} - o(1)}$ for N sufficiently large. This means that $\frac{N}{h(N)}(2h(N/h(N)) - h(N)) \geq N(2^{1-c^2/2-o(1)} - 1)$, which is positive when $c < \sqrt{2}$. Next, We have

$$\begin{aligned}
\lim_{N \to \infty} \sqrt{\log N} - \sqrt{\log(N - N/h(N))} &= \lim_{N \to \infty} \sqrt{\log N} - \sqrt{\log N - \log\left(\frac{1}{1 - 1/h(N)}\right)} \\
&= \lim_{N \to \infty} \sqrt{\log N} - \sqrt{\log N - 0} \\
&= 0,
\end{aligned}$$

meaning $h(N - N/h(N))/h(N) \leq 2^{-o(1)}$. Thus,

$$\begin{aligned}
\lim_{N \to \infty} g(N) - f(N) &= \lim_{N \to \infty} N(2^{1-c^2/2-o(1)} - o(1)) \\
&= \infty \qquad \text{if } 0 < c < \sqrt{2}.
\end{aligned}$$

□

Theorem 3 (Classical vs. Reversible Cumulative Pebbling Complexity). *Let $G = (V = [N], E)$ be a DAG. Then*

$$\Pi_{cc}^{\leftrightarrow, \|}(G) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \Pi_{cc}^{\|}(G),$$

and

$$\Pi_{cc}^{\leftrightarrow}(G) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \Pi_{cc}(G).$$

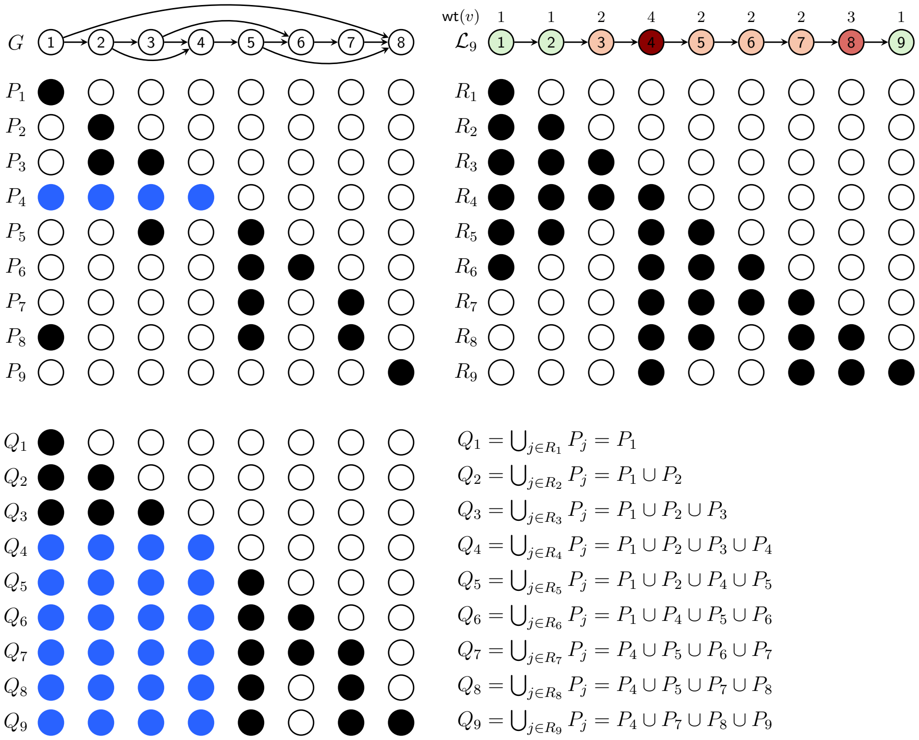
Proof. Let $P = (P_1, \ldots, P_t)$ be a parallel pebbling of G with $\Pi_{cc}(P) = \Pi_{cc}^{\|}(G)$ and note that optimality ensures that $t \leq N^2$. We define a weighted line graph with t nodes and weight $\mathsf{wt}_i = |P_i|$ for each node $i \leq t$. We can apply Theorem 6 to obtain a reversible pebbling $L = (L_1, \ldots, L_{t'})$ of the weighted line graph with weighted cumulative cost $\Pi_{wcc}(L) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \sum_i \mathsf{wt}_i = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \Pi_{cc}^{\|}(G)$. Let $Q = L \circ P$ be the composed reversible pebbling as in Theorem 4. We have $\Pi_{cc}(Q) = \sum_{i \leq t'} \sum_{v \in L_i} |P_i| = \sum_{i \leq t'} \sum_{v \in L_i} \mathsf{wt}_i = \Pi_{wcc}(L) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \Pi_{cc}^{\|}(G)$.

The proof for sequential pebbling is similar. Let $P = (P_1, \ldots, P_t)$ be a minimum cost special sequential pebbling (we can remove at most one node in each round and we cannot remove a pebble during the same round where we add a new pebble) of G. By Theorem 14 we are guaranteed that $\Pi_{cc}(P) \leq 6\Pi_{cc}(G)$. By optimality we also know that $t \leq N^2$ since the naive pebbling is special sequential and has cumulative cost less than N^2. We can apply Corollary 1 to argue that the composed pebbling $Q = L \circ P$ derived as in Theorem 4 is sequential as long the reversible pebbling L is sequential. We define a weighted line graph with t nodes and weight $\mathsf{wt}_i = |P_i|$ for each node $i \leq t$. We can apply Theorem 6 to obtain a reversible sequential pebbling $L = (L_1, \ldots, L_{t'})$ of the weighted line graph with weighted cumulative cost $\Pi_{wcc}(L) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \sum_i \mathsf{wt}_i = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \Pi_{cc}(G)$. Let $Q = L \circ P$ be the composed reversible pebbling as in Theorem 4. Q is a sequential pebbling (by Corollary 1) and we have $\Pi_{cc}(Q) = \sum_{i \leq t'} \sum_{v \in L_i} |P_i| = \sum_{i \leq t'} \sum_{v \in L_i} \mathsf{wt}_i = \Pi_{wcc}(L) = \mathcal{O}\left(N^{\frac{\mathcal{O}(1)}{\sqrt[4]{\log N}}}\right) \cdot \Pi_{cc}(G)$. This completes the proof. □

C.1 Pebbling Composition using Weighted Reversible Line Graph Pebbling

This section gives a concrete example of an efficient transformation from a classical to a reversible pebbling via pebbling composition introduced in Section 3.2. Recall that the transformation works as we take a classical (irreversible) pebbling $P = (P_1, \ldots, P_t)$ and make a composition with a reversible pebbling $R = (R_1, \ldots, R_{t'})$ of the line graph $\mathcal{L}_t$ which yields the pebbling $Q = R \circ P = (Q_1, \ldots, Q_{t'})$ defined by $Q_i = \bigcup_{j \in R_i} P_j$ for $i \in [t']$. Theorem 4 showed that the pebbling composition of a classical pebbling with Blocki et al's reversible line graph pebbling [BHL22], we obtain a legal reversible pebbling that preserves the *space-time cost* within a subpolynomial factor. However, as discussed in Section 3.2.2, the same strategy completely fails to (approximately) preserve the *cumulative pebbling complexity*. Here, we give a concrete example to illustrate why a pebbling composition of a classical pebbling with a reversible line graph pebbling of Blocki et al. [BHL22] could go wrong and fail to preserve the cumulative pebbling complexity. Then we also give an example of a pebbling composition with the *weighted reversible line graph pebbling* $\mathsf{WRevLinePeb}^{\parallel}$ and provide an intuitive explanation of why it is more CC-efficient than the previous strategy and therefore preserves the cumulative pebbling complexity.

Consider a DAG $G = (V = [8], E)$ where the edge set E is given as follows: $E = \{(i, i+1) : i \in [7]\} \cup \{(1,8), (2,4), (3,6), (5,8)\}$. In Figure 3, we give a classical (irreversible) pebbling $P = (P_1, \ldots, P_9)$, and one can easily verify that this is a legal classical pebbling. We remark that the pebbling P is mostly space-efficient *except for* the fourth round P_4 (which takes up half of the entire space), highlighted in blue. Since P consists of 9 rounds, we construct a line graph $\mathcal{L}_9$ with 9 nodes with weight $\mathsf{wt}_i = |P_i|$ for $i \in [9]$. As it is color-coded below, we can see that node 4 in $\mathcal{L}_9$ has the highest weight since P_4 contains the most number of pebbles in G. Now consider the reversible line graph pebbling $R = (R_1, \ldots, R_9)$ of Blocki et al. [BHL22] which modified Bennett's sequential reversible pebbling strategy [Ben89]. Since it did not consider the weight of nodes in $\mathcal{L}_9$, it could maintain pebbles on the high-weight nodes for a long time, just as illustrated in our example (it maintains a pebble on node 4 for 6 rounds). As a result, the pebbling composition Q contains P_4 for a large number of rounds, leading to a non-efficient CC reversible pebbling as shown in Figure 3, i.e., $\Pi^{\parallel}_{cc}(Q) \gg \Pi^{\parallel}_{cc}(P)$.



	1	2	3	4	5	6	7	8
P_1	●	○	○	○	○	○	○	○
P_2	○	●	○	○	○	○	○	○
P_3	○	●	●	○	○	○	○	○
P_4	🔵	🔵	🔵	🔵	○	○	○	○
P_5	○	○	●	○	●	○	○	○
P_6	○	○	○	○	●	●	○	○
P_7	○	○	○	○	●	○	●	○
P_8	●	○	○	○	●	○	●	○
P_9	○	○	○	○	○	○	○	●

	1	2	3	4	5	6	7	8	9
R_1	●	○	○	○	○	○	○	○	○
R_2	●	●	○	○	○	○	○	○	○
R_3	●	●	●	○	○	○	○	○	○
R_4	●	●	●	●	○	○	○	○	○
R_5	●	●	○	●	●	○	○	○	○
R_6	●	○	○	●	●	●	○	○	○
R_7	○	○	○	●	●	●	●	○	○
R_8	○	○	○	●	●	○	●	●	○
R_9	○	○	○	●	○	○	●	●	●

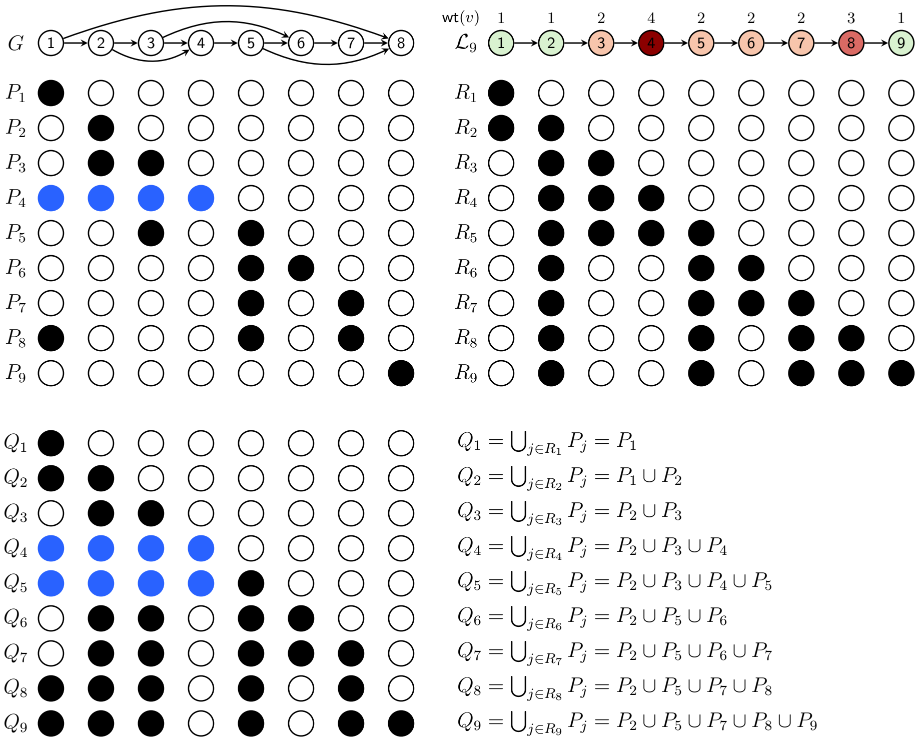
	1	2	3	4	5	6	7	8
Q_1	●	○	○	○	○	○	○	○
Q_2	●	●	○	○	○	○	○	○
Q_3	●	●	●	○	○	○	○	○
Q_4	🔵	🔵	🔵	🔵	○	○	○	○
Q_5	🔵	🔵	🔵	🔵	●	○	○	○
Q_6	🔵	🔵	🔵	🔵	●	●	○	○
Q_7	🔵	🔵	🔵	🔵	●	●	●	○
Q_8	🔵	🔵	🔵	🔵	●	○	●	○
Q_9	🔵	🔵	🔵	🔵	●	○	●	●

$$Q_1 = \bigcup_{j \in R_1} P_j = P_1$$
$$Q_2 = \bigcup_{j \in R_2} P_j = P_1 \cup P_2$$
$$Q_3 = \bigcup_{j \in R_3} P_j = P_1 \cup P_2 \cup P_3$$
$$Q_4 = \bigcup_{j \in R_4} P_j = P_1 \cup P_2 \cup P_3 \cup P_4$$
$$Q_5 = \bigcup_{j \in R_5} P_j = P_1 \cup P_2 \cup P_4 \cup P_5$$
$$Q_6 = \bigcup_{j \in R_6} P_j = P_1 \cup P_4 \cup P_5 \cup P_6$$
$$Q_7 = \bigcup_{j \in R_7} P_j = P_4 \cup P_5 \cup P_6 \cup P_7$$
$$Q_8 = \bigcup_{j \in R_8} P_j = P_4 \cup P_5 \cup P_7 \cup P_8$$
$$Q_9 = \bigcup_{j \in R_9} P_j = P_4 \cup P_7 \cup P_8 \cup P_9$$

Fig. 3: Pebbling composition using reversible line graph pebbling [BHL22] for $\mathcal{L}_9$

On the other hand, Figure 4 depicts the pebbling configuration when we replace Blocki et al's reversible line graph pebbling [BHL22] to our *weighted reversible line graph pebbling* $\mathsf{WRevLinePeb}^{\parallel}([9], \mathcal{S}, 0, L)$. As you see the weighted reversible line graph pebbling $R = (R_1, \ldots, R_9)$ in Figure 4, it focuses on minimizing the number of rounds that keeps pebbles on the high-weight nodes.

As described in Section 3.2.2, $\mathsf{WRevLinePeb}^{\parallel}([9], \mathcal{S}, 0, L)$, where we split $\mathcal{S} = (S_0, S_1, S_2)$ into $S_0 = \{v : 2^0 \leq \mathsf{wt}_v < 2^1\} = \{1, 2, 9\}$, $S_1 = \{v : 2^1 \leq \mathsf{wt}_v < 2^2\} = \{3, 5, 6, 7, 8\}$, and $S_2 = \{v : 2^2 \leq \mathsf{wt}_v < 2^3\} = \{4\}$, works as follows.


G
wt(v) 1 1 2 4 2 2 2 3 1
$\mathcal{L}_9$
1 2 3 4 5 6 7 8 9
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9
R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8 R_9
Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 Q_7 Q_8 Q_9

$Q_1 = \bigcup_{j \in R_1} P_j = P_1$

$Q_2 = \bigcup_{j \in R_2} P_j = P_1 \cup P_2$

$Q_3 = \bigcup_{j \in R_3} P_j = P_2 \cup P_3$

$Q_4 = \bigcup_{j \in R_4} P_j = P_2 \cup P_3 \cup P_4$

$Q_5 = \bigcup_{j \in R_5} P_j = P_2 \cup P_3 \cup P_4 \cup P_5$

$Q_6 = \bigcup_{j \in R_6} P_j = P_2 \cup P_5 \cup P_6$

$Q_7 = \bigcup_{j \in R_7} P_j = P_2 \cup P_5 \cup P_6 \cup P_7$

$Q_8 = \bigcup_{j \in R_8} P_j = P_2 \cup P_5 \cup P_7 \cup P_8$

$Q_9 = \bigcup_{j \in R_9} P_j = P_2 \cup P_5 \cup P_7 \cup P_8 \cup P_9$

Fig. 4: Pebbling composition using $\mathsf{WRevLinePeb}^{\|}([9], \mathsf{wt}, \mathcal{S}, 0, L)$ for $\mathcal{L}_9$

- First, $\mathsf{WRevLinePeb}^{\|}([9], \mathcal{S}, 0, L)$ tries to pebble a subgraph $\mathcal{L}_{S_0} = (S_0, E_{S_0})$ where $E_{S_0} = \{(1,2), (2,9)\}$. Since a pebbling move from node 2 to 9 is indeed *illegal* in $\mathcal{L}_9$, we make a recursive call $\mathsf{WRevLinePeb}^{\|}([3,8], \mathcal{S}, 1, L)$. (Note: in Figure 4, let $\widetilde{R}$ be the pebbling where we ignore the pebbling rounds $R_3, \ldots, R_8$ (which are the rounds for the recursive call $\mathsf{WRevLinePeb}^{\|}([3,8], \mathcal{S}, 1, L)$) and nodes $3, \ldots, 8$, then $\widetilde{R} = (\widetilde{R}_1 = \{1\}, \widetilde{R}_2 = \{1,2\}, \widetilde{R}_9 = \{2,9\})$ is a legal reversible pebbling for $\mathcal{L}_{S_0}$.)
- Next, $\mathsf{WRevLinePeb}^{\|}([3,8], \mathcal{S}, 1, L)$ tries to pebble a subgraph $\mathcal{L}_{S_1} = (S_1, E_{S_1})$ where $E_{S_1} = \{(3,5), (5,6), (6,7), (7,8)\}$. When pebbling $\mathcal{L}_{S_1}$, every pebbling move is legal except for $(3,5)$, hence we need to make a recursive call $\mathsf{WRevLinePeb}^{\|}([4,4], \mathcal{S}, 2, L)$ to bridge the gap. Taken together, we obtain the entire pebbling sequence as illustrated in Figure 4. Note that we only described the *relaxed* parallel reversible pebbling of $\mathcal{L}_9$ in our figure, but it is straightforward to obtain a non-relaxed version by going reverse while keeping a pebble on the sink node, e.g., $R_i = R_{17-i} \cup \{9\}$ for $i \in [10, 16]$.

Now consider the pebbling composition $Q = (Q_1, \ldots, Q_9)$ as illustrated in Figure 4. Since our algorithm, $\mathsf{WRevLinePeb}^{\|}([9], \mathcal{S}, 0, L)$ maintains the minimum number of rounds that keep pebbles on the highest-weight nodes, we can observe that P_4 only appears twice in Q (Q_4 and Q_5, highlighted in blue), which allows us to achieve a CC-efficient reversible pebbling Q. As we discussed in Section 3.2, we can find a reversible pebbling that preserves reversible CC up to subpolynomial factors (see Theorem 3).

D Reversible Recursive Pebbling Attack

Lemma 3. *For any (e_1, d_1)-depth reducible DAG $G = (V = [N], E)$ of depth d_0, target set $T' \subseteq [N]$, and family of pebblings $B(G', T', t')$ for all DAGs $G' = (V', E')$, target sets $T' \subseteq V'$, and $t' \geq 2 \cdot \mathsf{depth}(G')$, the pebbling*

$$P = \mathsf{RRGenPeb}(G, d_0, \{(e_1, d_1, S_1)\}, B)$$

is a legal parallel reversible pebbling of G, where S_1 is a depth-reducing set of size e_1.

Proof. Since $g \geq 2d_1$, each balloon phase is contained in the corresponding light phase. Next, since $\mathsf{LightReq}^c$ is a reversible pebbling sequence and $\mathsf{BalloonReq}^c$ is a reversible pebbling sequence, their union is as well. Now we will consider the transitions between phases. We have that $\mathsf{BalloonReq}^c_{2g} = \mathsf{parents}(I_{c+1}) \backslash I_{c+1}$ and $\mathsf{LightReq}^c_{2g} = S_{\preceq cg}$. Then there is a legal move from $\mathsf{LightReq}^c_{2g} \cup \mathsf{BalloonReq}^c_{2g}$ to $\mathsf{LightReq}^c_1 \cup \mathsf{BalloonReq}^c_1 = S_{\preceq cg+1} \cup \{cg+1\}$. Thus, every step in P is reversible. Since the first half of each light phase pebbles exactly one set D_j per step and the second half takes exactly as many steps as the first half, it follows that $\Pi_t(P) \leq 4d_0$. By the definition of $\mathsf{LightReq}$, it follows that $P_{|P|} = T$. So, P is a legal reversible pebbling. □

Lemma 4. *For any (e_1, d_1)-depth reducible DAG $G = (V = [N], E)$ of depth d_0, target set $T' \subseteq [N]$, and family of pebblings $B(G', T', t')$ for all DAGs $G' = (V', E')$, target sets $T' \subseteq V'$, and $t' \geq 2 \cdot \mathsf{depth}(G')$,*

$$\begin{aligned}&\Pi_{cc}\left(\mathsf{RRGenPeb}(G, d_0, \{(e_1, d_1, S_1)\}, B)\right)\\&\qquad \leq 4d_0(\delta+2)e_1 + 4d_0|T| + \frac{2e_1}{N} \cdot \max_{|T'| \leq \delta e_1} \Pi_{cc}\left(B\left(G - S, T', 2d_1\right)\right),\end{aligned}$$

where S_1 is a depth-reducing set of size e_1.

Proof. During the c^{th} light phase, we have pebbles on at most S, T, $\mathsf{parents}(I_c)$, and I_c, so

$$\Pi_s(\mathsf{LightReq}) \leq e_1 + (\delta+1)g\frac{N}{d_0} + |T| \leq (\delta+2)e_1 + |T|.$$

Thus, the contribution of all of the light phases to the CC of P is at most $4d_0(\delta+2)e_1 + 4d_0|T|$. Next, the contribution to the CC of P of the balloon phases is at most

$$\frac{2d_0}{g} \cdot \max_{|T'| \leq \delta e_1} \Pi_{cc}\left(B(G - S, T, 2d_1)\right) \leq \frac{2e_1}{N} \cdot \max_{|T'| \leq \delta e_1} \Pi_{cc}\left(B\left(G - S, T', 2d_1\right)\right).$$

Putting it all together we get

$$\Pi_{cc}(P) \le 4d_0(\delta+2)e_1 + 4d_0|T| + \frac{2e_1}{N} \cdot \max_{|T'| \le \delta e_1} \Pi_{cc}\left(B\left(G-S, T', 2d_1\right)\right)$$

for $1 \le i \le d_0$. □

E Depth Robustness and Reversible CC

Reminder of Theorem 10. *Let $G = (V = [N], E)$ be a DAG such that $(i, i+1) \in E$ for all $i < N$ and the graph $G_{\mathsf{Trunc},d}$ is (e,d)-depth robust. Then $\widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G) \ge e(2d-1)$.*

Proof of Theorem 10. Let $P_1, \ldots, P_t$ be a *relaxed* reversible pebbling of G. As before for each $i \in [2d-1]$ we let $B_i = P_i \cup P_{i+2d-1} \cup P_{i+2(2d-1)} \cup \ldots \cup P_{i+m(2d-1)}$ where $m = m(i)$ is the largest integer such that $i + m(2d-1) \le t$. As before we note that $\sum_i |B_i| \le \sum_j |P_j| = \widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G)$. It follows that there exists some $B \coloneqq B_i$ with $|B| \le \frac{\widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G)}{2d-1}$.

Now we will show there is no path of length d in $G_{\mathsf{Trunc},d} - B$. Suppose, for contradiction, that there exists a node $v \in [N-d] \backslash B$ such that $\mathsf{depth}(v, G_{\mathsf{Trunc},d} - B) \ge d$. Let $p(v)$ be the first step in which node v is pebbled. Then we observe the following claims:

Claim 1. *$p(v) \le t - d$.*

Proof of Claim 1. Since node $N \in P_t$ which implies $N - d \in P_{t-d}$ (otherwise it would not have been able to place a pebble on node N on round t), which is the last node in $G_{\mathsf{Trunc},d}$. Hence, v must have been pebbled some round on/before P_{t-d}. □

Claim 2. *$p(v) > i + m(2d-1)$.*

Proof of Claim 2. Suppose not. Then there exists some j with $j + 1 \le m$ such that $i + j(2d-1) < p(v) < i + (j+1)(2d-1)$ (here, $p(v) \ne i + j(2d-1)$ and $p(v) \ne i + (j+1)(2d-1)$ since $v \notin P_{i+j(2d-1)} \cup P_{i+(j+1)(2d-1)}$). Since $\mathsf{depth}(v, G_{\mathsf{Trunc},d} - B) \ge d$, it would take at least d steps to place a pebble on node v starting from $P_{i+j(2d-1)}$ and then take at least d steps to remove this pebble before $P_{i+(j+1)(2d-1)}$. This is a contradiction since there are fewer than $2d$ intermediate rounds between $P_{i+j(2d-1)}$ and $P_{i+(j+1)(2d-1)}$. Hence, we can conclude that $p(v) > i + m(2d-1)$. □

Hence, we have $i + m(2d-1) < p(v) \le t - d$. Now by definition of m, we observe that

$$p(v) - (i + m(2d-1)) \le t - d - (i + m(2d-1))$$

$$\begin{aligned} &= [t - (i + m(2d-1))] - d \\ &< (2d-1) - d = d - 1, \end{aligned}$$

since m was the largest integer such that $i + m(2d-1) \leq t$ which implies $t < i + (m+1)(2d-1)$. This implies that there are less than $d-1$ rounds between $P_{i+m(2d-1)}$ and $P_{p(v)}$. However, at time $P_{i+m(2d-1)}$, there is an unpebbled path of length $\geq d$ ending at v, which means that it is impossible to place a pebble on v at time $P_{p(v)}$. Contradiction! (as we defined $p(v)$ to be the first step in which node v is pebbled.) This contradiction was caused due to the assumption $\mathsf{depth}(v, G_{\mathsf{Trunc},d} - B) \geq d$. Hence, we can conclude that there is no path of length d in $G_{\mathsf{Trunc},d} - B$, which implies that $G_{\mathsf{Trunc},d}$ is $(|B|, d)$-reducible. Since $G_{\mathsf{Trunc},d}$ is (e,d)-depth robust, we have $|B| \geq e$. Combining with $|B| \leq \frac{\widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G)}{2d-1}$, we can conclude that $\widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G) \geq e(2d-1)$. □

Remark 1. We can make an improvement on the lower bound of the relaxed parallel reversible cumulative pebbling cost of DRSample by applying Theorem 10. Recall that DRSample [ABH17] is the first practical construction of a data-independent MHF, which is a graph $G = (V = [N], E)$ that has the following edge distribution: $E = \{(i, i+1) : i \in [N-1]\} \cup \{(r(v), v) : i \in [3, N]\}$, where $r(v)$ is picked according to the following random process: (1) randomly select a bucket index $i \leq \log v$, and (2) randomly sample $r(v)$ from the bucket $B_i(v) = \{u : 2^{i-1} < v - u \leq 2^i\}$.

Let $G^{\mathsf{DRS}} = (V^{\mathsf{DRS}} = [N], E^{\mathsf{DRS}})$ be a randomly sampled graph according to the DRSample edge distribution. Then we know that (whp) G^{DRS} is $(c_1 N/\log N, c_2 N)$-depth robust for some constant $c_1, c_2 > 0$, which implies that

$$\widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G^{\mathsf{DRS}}) \geq \Pi_{cc}^{\parallel}(G^{\mathsf{DRS}}) \geq \frac{c_1 c_2 N^2}{\log N},$$

by the previous lower bound [ABP17].

Now we observe that, due to the way that DRSample's edge distribution is defined, $G_{\mathsf{Trunc},d}^{\mathsf{DRS}}$ can simply be viewed as a randomly sampled DRSample graph with $N - d$ nodes. Thus, (whp) $G_{\mathsf{Trunc},d}^{\mathsf{DRS}}$ is $(c_1(N-d)/\log(N-d), c_2(N-d))$-depth robust. To apply Theorem 10, we would need the condition $d = c_2(N-d)$, which can be solved by setting $d = c_2 N/(1+c_2)$. Then we have that $G_{\mathsf{Trunc},\frac{c_2 N}{1+c_2}}^{\mathsf{DRS}}$ is $\left(\frac{c_1 N}{(1+c_2)\log(N/(1+c_2))}, \frac{c_2 N}{1+c_2}\right)$-depth robust. Then by Theorem 10, we have

$$\begin{aligned} \widetilde{\Pi}_{cc}^{\Longleftrightarrow,\parallel}(G^{\mathsf{DRS}}) &\geq \frac{c_1 N}{(1+c_2)\log(N/(1+c_2))}\left(\frac{2c_2 N}{1+c_2} - 1\right) \\ &\geq \frac{c_1 N}{(1+c_2)\log N}\left(\frac{2c_2 N}{1+c_2} - 1\right) \\ &= \frac{\alpha}{(1+c_2)^2} \cdot \frac{c_1 c_2 N^2}{\log N}, \end{aligned}$$

where $\alpha = 2 - \frac{1+c_2}{c_2 N}$. We can observe that as long as we have $\frac{\alpha}{(1+c_2)^2} > 1$, this is an improvement from the classical lower bound which immediately carries over to the reversible case. Since we have $c_2 = 0.03$ [ABP17], we can see that as long as $N > 1.03/(0.03 \times (2 - 1.03^2)) \simeq 35.8$ we achieve an improvement. In particular, if $N \geq 1.03/(0.03 \times (2 - t \cdot 1.03^2))$ then we can achieve an improvement by multiplicative factor of t, e.g., if $N = 10^7$ then we can expect an improvement by multiplicative factor of t up to $t \leq \left(2 - \frac{1.03}{0.03N}\right) \cdot 1.03^{-2} \approx 1.885$. As $N \to \infty$, we have $t \to 2/1.03^2 \approx 1.88519$.