Electronic Colloquium on Computational Complexity, Report No. 11 (2026)

Complete Characterization of Randomness Extraction from DAG-Correlated Sources

Divesh Aggarwal* Zihan Li† Saswata Mukherjee‡ Maciej Obremski§
João Ribeiro¶

Abstract

We introduce the SHEDAG (Somewhere Honest Entropic sources over Directed Acyclic Graphs) source model, a general model for multi-block randomness sources with causal correlations. A SHEDAG source is defined over a directed acyclic graph (DAG) G whose nodes output n-bit blocks. Blocks output by honest nodes are independent (by default uniformly random, more generally having high min-entropy), while blocks output by corrupted nodes are arbitrary functions of their causal views (all predecessors in G). We tightly characterize the conditions under which randomness extraction from SHEDAG sources is possible.

Zero-error extraction: We show that perfect extraction from SHEDAG sources with t corruptions is possible if and only if G contains an "unrelated set" (an antichain under reachability) of size at least $t + 1$. Conversely, if every unrelated set has size at most t, we show that no function can output a perfectly uniform bit. We also provide a polynomial-time algorithm to find a maximum unrelated set, thus efficiently identifying the largest corruption threshold t allowing perfect extraction.

Negligible-error extraction: We identify a quantity that we call "resilience" of a DAG G, denoted $\mathrm{res}(G)$, that characterizes the possibility of randomness extraction with *negligible* error (in the block length). We show that negligible-error extraction is impossible whenever $t > \mathrm{res}(G)$, and, to complement this, for every $t \leq \mathrm{res}(G)$ we construct explicit extractors with polynomial output length and negligible error.

Our results generalize prior online source models studied by (Aggarwal, Obremski, Ribeiro, Siniscalchi, Visconti, Eurocrypt 2020) and (Chattopadhyay, Gurumukhani, Ringach, FOCS 2024), which correspond to the special case of a SHEDAG source whose DAG G is a path.

*National University of Singapore. divesh@comp.nus.edu.sg.
†National University of Singapore. zihan_li_05@u.nus.edu
‡National University of Singapore. saswatamukherjee607@gmail.com
§National University of Singapore. obremski.math@gmail.com.
¶Instituto de Telecomunicações and Departamento de Matemática, Instituto Superior Técnico, Universidade de Lisboa. jribeiro@tecnico.ulisboa.pt

ISSN 1433-8092

Contents

1 Introduction

Randomness is a fundamental resource across computer science. It can help speed up algorithms significantly and simplify and speed up interactive and distributed protocols. Randomness is inherent to cryptography, and many cryptographic tasks are impossible without uniform randomness. Conceptually, two of the most fundamental questions in this area are:

- How to model weak randomness to mimic the behavior of real systems?
- When is it possible to convert such weak randomness into uniform randomness?

Decades of work has established both the power and limitations of deterministic extraction: for many natural models of sources (such as min-entropy sources) exact extraction is impossible, motivating both the study of structured models and relaxed goals that capture useful properties that can still be achieved.

A large body of work has studied sequential and block-structured weak sources. Classical Santha-Vazirani and Chor-Goldreich sources [SV86, CG88] capture bounded unpredictability per bit/symbol. More recently, sources tailored to online settings (e.g., modeling communication in protocols) where honest blocks are mixed with adversarial ones that may depend arbitrarily on the past, were formalized as SHELA sources [AOR+20]. These sources have been studied both from the perspective of building randomness extractors and randomness condensers. For Chor-Goldreich sources, errorless condensing is impossible [GP20], while non-trivial condensing with error was shown for certain regimes with very small block lengths [DMOZ23, GLZ24]. For online and non-oblivious symbol-fixing models (oNOSF/NOSF), sharp condensing thresholds and separations from extraction have recently been established [CGR24, CGRS24].

1.1 Our contributions

We introduce the *SHEDAG* source model, where "SHEDAG" stands for "Somewhere Honest Entropic sources over Directed Acyclic Graphs". A SHEDAG source is parameterized by a DAG $G = (V, E)$ whose nodes each output an n-bit block, and a corruption threshold t. The nodes of G are partitioned into set of *honest* nodes and a set of at most t *corrupted nodes*. The blocks output by honest nodes are independent of each other, while the block output by a corrupted node is an arbitrary function of their causal view (all predecessors along directed paths). Unless otherwise stated, in this work we assume that the blocks output by honest nodes are uniformly distributed. This DAG abstraction generalizes SHELA sources (the corresponding DAG being a path), and captures causal signal propagation and general dependency structures (a special case of Bayesian networks).

We now formally define the SHEDAG source model. Before that, we define some basic notions related to DAGs.

Given a directed acyclic graph (DAG) $G = (V, E)$ we denote the in-degree and out-degree of a vertex $v \in V$ by in-deg(v) and out-deg(v), respectively.

Definition 1 (View of a vertex)**.** *Given a DAG $G = (V, E)$ and a vertex $v \in V$, the* view *of v (in G), denoted* view(v)*, is the set of all vertices u for which there is a path from u to v in G.*

Definition 2 (SHEDAG source)**.** *Fix a DAG $G = (V, E)$ with vertex set $V = [N]$. Then, $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ is said to be an* (n, k, G, t)-SHEDAG source *if there is a subset $S \subseteq V$ of size at most t such that*

1. *$\{\mathbf{X}_i\}_{i \in V \setminus S}$ are independent (n, k)-sources;*[1]
2. *For every $j \in S$ there is a possibly randomized function f_j (using fresh independent randomness) such that*
$$\mathbf{X}_j = f_j(\mathbf{X}_{j_1}, \ldots, \mathbf{X}_{j_\ell}),$$
where $\text{view}(j) = \{j_1, \ldots, j_\ell\}$.

We call G the base graph *of $\mathbf{X}$. When $\mathbf{X}_i$ for $i \notin S$ are uniformly random over $\{0,1\}^n$, we say that $\mathbf{X}$ is an (n, G, t)-SHEDAG source.*

Remark 1. When the base graph $G = (V, E)$ has edge set $E = \{(i, i+1) : i \in [N-1]\}$ (see Figure 1), we recover the SHELA source model from [AOR+20].

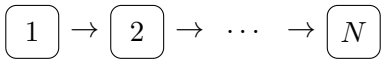

Figure 1: Base graph of a SHELA source from [AOR+20].

Definition 3 (Extractor for SHEDAG sources)**.** *We say that a function $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ is an $(n, k, G, t, \varepsilon)$*-extractor for SHEDAG-sources *if $\Delta(\mathsf{Ext}(\mathbf{X}) \,;\, \mathcal{U}_m) \leq \varepsilon$ for every (n, k, G, t)-SHEDAG source $\mathbf{X}$, where Δ denotes statistical distance and $\mathcal{U}_m$ denotes the uniform distribution over $\{0,1\}^m$.*

We study randomness extraction from SHEDAG sources, with a focus on zero-error and negligible-error extraction (here, "negligible" means negligible in the source block length n, as usual in the randomness extraction literature). Zero-error extraction is of mostly theoretical interest – it is a nice starting point for understanding SHEDAG sources because it leads to a particularly clean landscape. Negligible-error extraction is highly relevant for applications in cryptography. Namely, we are interested in understanding the following:

- For which tuples (n, G, t) of block length, base graph, and corruption threshold is there a zero-error/negligible-error randomness extractor for the class of all (n, G, t)-SHEDAG sources?
- In cases where randomness extraction is possible, can we construct explicit extractors with matching parameters?

In both the zero-error and negligible-error settings, we completely characterize the parameters (n, G, t) for which randomness extraction is possible, and give matching explicit extractors.

[1] We say that $\mathbf{X}$ is an (n, k)-*source* if $\mathbf{X} \in \{0,1\}^n$ and $\mathbf{X}$ has min-entropy $\mathbf{H}_\infty(\mathbf{X}) \geq k$.

Zero-error randomness extraction. When $k < n$, it is not hard to see that zero-error extraction is impossible even when G is the empty graph (and so all blocks are independent (n, k)-sources). Therefore, we focus on the case $k = n$, and obtain a necessary and sufficient condition for the existence of a zero-error extractor for (n, G, t)-SHEDAG sources. Namely, we show that there zero-error extraction is possible if and only if the base graph G can be decomposed into at least $t + 1$ "unrelated sets", which we define next.

Definition 4 (Unrelated set)**.** *Given a DAG $G = (V, E)$, a subset of vertices $U \subseteq V$ is said to be an* unrelated set *if for any two distinct vertices $u, v \in U$ there are no paths from u to v or from v to u in G. In this case, we also say that u and v are* unrelated*.*

Our complete characterization is formalized in the following theorem.

Theorem 1. *For any $n \in \mathbb{N}$, fixed $N \in \mathbb{N}$, $t \leq N$, and N-vertex DAG G, zero-error randomness extraction from (n, G, t)-SHEDAG sources is possible if and only if G has an unrelated set of size at least $t + 1$.*

Randomness extraction with negligible error. If we allow a small extraction error ε, then the landscape for feasibility changes. To state our results we need some additional definitions.

Definition 5 (Head vertex)**.** *Given a DAG $G = (V, E)$, we say that $v \in V$ is a* head vertex *(in G) if* out-deg$(v) = 0$*. We denote the set of all head vertices in G by* Head(G)*.*

In words, our positive result states that if any corruption pattern always leaves some head of the base graph G and its view uncorrupted, then there is a low-error extractor for (n, G, t)-SHEDAG sources. In fact, we are also able to extract with low error from *entropic* SHEDAG sources, as opposed to only SHEDAG sources whose good blocks are uniformly distributed. For simplicity, we focus on the latter task here and leave a discussion of the more general result to Section 4.3.2.

We begin by defining the *resilience* of a DAG G, which captures the scenario mentioned in the previous paragraph.

Definition 6 (Resilience)**.** *Given a DAG $G = (V, E)$, the* resilience *of a subset of vertices $S \subseteq V$, denoted* $\mathrm{res}_S(G)$*, is defined as*

$$\mathrm{res}_S(G) := (|S| - 1) - \max_{s \in S} |\,\mathrm{view}(s) \cap S|.$$

We define the resilience *of G as* $\mathrm{res}(G) := \max_{S \subseteq V} \mathrm{res}_S(G)$*.*

In Remark 3 we give a outline of the proof of $\mathrm{res}(G) \geq 0$ for any DAG G. We show that the feasibility of extracting randomness with negligible error is completely characterized by the resilience of the base graph. This is formalized in the following theorems.

Theorem 2 (Feasibility of negligible-error extraction)**.** *Fix an integer $N \geq 1$ and an N-vertex DAG G. Then, for any $t \leq \mathrm{res}(G)$, there exists an explicit $(n, k = \rho n, G, t, \varepsilon = 2^{-n^{\Omega(1)}})$-SHEDAG extractor, for some constant $\rho < 1$ depending only on N.*

We complement the explicit construction in Theorem 2 with an impossibility criterion. Formally we have the following theorem (restated at Corollary 1).

Theorem 3 (Impossibility of negligible-error extraction)**.** *Fix an integer $N \geq 1$ and an N-vertex DAG G. Then, there is a constant $c > 0$ such that for every function $f : (\{0,1\}^n)^N \to \{0,1\}$ there exists an $(n, G, t = \mathrm{res}(G) + 1)$-SHEDAG source $\mathbf{X}$ with*

$$\Delta(f(\mathbf{X}) \,;\, \mathcal{U}_1) \geq n^{-c}.$$

1.2 Related work

At a high level, our work focuses on randomness extraction from multiple sources of randomness with structured correlations, as opposed to the widely studied setting of randomness extraction from multiple *independent* sources. This direction has seen plenty of interest. We provide a brief survey of previous models and results, and compare our SHEDAG model to other existing models.

The SHEDAG model is an *online* source model. The DAG G induces an ordering of the nodes/blocks, such that if the i-th node is corrupted then its block may only depend on (some of) the blocks of nodes $j < i$. More generally, an online source is a source that outputs a sequence of blocks $B_1, B_2, \ldots, B_N$, where the value of B_i may depend in some way on $B_1, \ldots, B_{i-1}$ only. The study of online sources goes back to the work of Santha and Vazirani [SV86] and Chor and Goldreich [CG88], and extensions of these models, such as almost Chor-Goldreich sources [DMOZ23] and unpredictable sources [DMOZ25], are still being studied. In an orthogonal direction, Aggarwal, Obremski, Ribeiro, Siniscalchi, and Visconti [AOR+20] introduced the model of SHELA sources, which was later also studied by Chattopadhyay, Gurumukhani, Ringach, and Servedio [CGR24, CGRS25], under the alternative name of *online non-oblivious symbol fixing* (oNOSF) sources. There are two important differences between SHELA sources and Chor-Goldreich-type sources: First, in SHELA sources only a subset t of blocks is corrupted, while the other blocks are independent and high-entropic – in Chor-Goldreich-type sources all blocks are (somewhat) dishonest. Second, dishonest blocks in SHELA sources may depend arbitrarily on previous blocks, including being fixed to a "worst-case" value. In contrast, in Chor-Goldreich-type sources the new block always carries some "uncertainty" (the uncertainty measure of interest may vary). Randomness extraction is impossible in all of these models. Because of this, prior work has focused on more relaxed tasks, such as extracting *somewhere-random* sources or deterministic condensing.

Our SHEDAG source model generalizes the SHELA/oNOSF source model by considering correlations described by DAGs other than a path. As we show in this work, there are interesting choices of the underlying DAG that allow for low-error (and sometimes even zero-error) randomness extraction (and we completely characterize all such DAGs).

Finally, we also note that there has been recent work on correlated multi-source models that do not fit the online source model. Examples include adversarial sources [CGGL20], where each corrupted source is allowed to depend arbitrarily on any *bounded* subset of sources, and somewhat correlated sources [BGM22], where there is bounded "dependence" between sources (according to some metric).

1.3 Acknowledgments

João Ribeiro thanks Yuval Ishai for insightful discussions that inspired the definition and study of the SHEDAG source model.

2 Technical Overview

In this section we give an overview of the results and proofs. We begin with a more detailed definition of SHEDAG sources. Given a directed acyclic graph G, each node has n bit value associated with it, and can be *corrupted* or *honest*. For the purpose of this exposition, the *honest* nodes values are simply independently sampled from uniform distribution (in general we allow them to be sampled from weak sources with some min-entropy). A *corrupted* node value can depend arbitrarily on the nodes it has in its *view*, the *view* of node v is defined as all nodes that have a path towards v (we exclude v from its own *view*). The direction of arrows corresponds to signal propagation, if there is an edge from u to v one should think of v "talking" later, i.e., picking its value after it saw value in node u. The $(n, G, t)-$SHEDAG source is defined on a directed acyclic graph G where honest nodes are uniform n bit strings, and there are at most t corrupted nodes.

We ask for which DAGs extraction of uniform randomness from SHEDAG sources is possible, how many corruptions the best extractor can withstand, and if we can give the explicit efficient construction of such optimal extractor. We resolve all of the above questions in two variants: *perfect* extraction (where output is perfectly uniform), and *statistical* extraction (where output is negligibly close to uniform).

For the simplicity of the exposition we focus on the scenario with *single bit output*, but we obtain strongest possible results: impossibility for a single bit output and a matching constructions for multi-bit output.

2.1 Perfect extraction

Let us begin with the definition: a set of nodes is *unrelated* if there is *no* path in the graph that leads from one node of the set to another node in this set. This can also be expressed in terms of *view*: for each node in the set its *view* does not contain any other node in the set. This means that in *unrelated* set all *honest* nodes are independent of each other, and importantly, independent of *corrupted* nodes (as each *corrupted* node can only depend on nodes in its *view*).

Extraction. Let us consider a DAG G with a large *unrelated* set: since all nodes in such set are independent we can simply XOR them, if at least one of them is *honest* we are done-output will be uniform. Extraction is therefore trivially possible if number of corruptions is strictly smaller than the size of largest *unrelated* set (as this clearly guarantees that at least one node in the set will be *honest*).

Naturally, the question is, can we do better? The answer is negative, above condition is tight. A detailed and formal statement can be found in Theorem 5.

The algorithm. We also provide an efficient algorithm to find the largest *unrelated* set, this completes the above construction. The key observation is that "reachability" in a DAG defines a partial order on its vertex set, and any set of unrelated vertices forms an *antichain* in this poset.

Impossibility. We show that if number of corruptions is greater or equal than the size of the largest *unrelated* set then it is not possible to extract even a single uniform bit. For a formal statement see Lemma 5.

Idea behind the proof: we start with an observation that the extraction from $(n, G, |G|)$-SHEDAG source (i.e. all of the nodes are *corrupted*) is clearly not possible. And then we inductively reduce the number of corruptions: we show that as long as the number of corruptions is greater than the size of the largest *unrelated* set, we can drop one corruption at a time and the output of extractor will remain biased. The intuition for the inductive step is following: since we have more corruptions than the size of the largest *unrelated* set, we know that there are two corrupted nodes u and v such that there is a path from u to v. We show that we can "uncorrupt" one of those two nodes and the extractor will remain biased (although the bias might get smaller). The gist of the proof is that node v can depend on the node u, and if we "uncorrupt" u (i.e. set it to something uniform), adversary can pick v accordingly and maintain some bias. One should note here that there is a small caveat: simply "uncorrupting" u might not work, as extractor might not depend on the node v at all, in that scenario we have to "uncorrupt" v - nevertheless corruption of both of the nodes is not necessary to bias the extractor. We can apply this inductive step as long as we have more corruptions than the size of the largest *unrelated* set. The argument stops exactly at number of corruptions being equal to the largest *unrelated* set, which tightly matches the extractor discussed earlier.

What if *honest* nodes are entropic instead of uniform. In this exposition we focus on *honest* nodes being uniform, but one can also consider honest nodes being sampled from some entropic distributions with guarantee that the output has at least k bits of entropy. We show that even if DAG has no edges i.e. all nodes are independent, and there is no corruptions, it is not possible to obtain perfect randomness in such scenario. Proof is inductive over number of sources. For single weak source it's a well known fact that extraction is not possible (simply pick X uniform over $\mathsf{Ext}^{-1}(i)$ for $i = 0$ or 1, the extraction output is fixed, and X has almost full entropy). Then we proceed with inductive step: given sources $X_1, ..., X_{t+1}$, and assume there exists an extractor Ext that produces perfect randomness. Fix any $x \in \mathsf{supp}(X_{t+1})$. By the inductive hypothesis we know that $\mathsf{Ext}(X_1, ..., X_t, x)$ has to be biased else we'd have perfect extractor for t weak sources. Consider two distributions: $\mathsf{Ext}(X_1, ..., X_t, U)$ and $\mathsf{Ext}(X_1, ..., X_t, U \setminus \{x\})$ where U is uniform over all n bit strings, and $U \setminus \{x\}$ is uniform over all n bit strings except x (both distributions have very high entropy). Since $\mathsf{Ext}(X_1, ..., X_t, x)$ is biased it is not possible that both $\mathsf{Ext}(X_1, ..., X_t, U)$ and $\mathsf{Ext}(X_1, ..., X_t, U \setminus \{x\})$ are perfect. In Observation 1 we give a more formal proof sketch.

2.2 Statistical extraction

Let us start with a definition: given a directed acyclic graph G, we call vertex v a *head* if there is no other vertex that has v in its *view*. Given any subset of nodes S we also consider a DAG G^S which is a graph with node set S and preserved paths, more precisely: for any $u, v \in S$ if there was a path from u to v in G then there will be a path from u to v in G^S. We can also consider *head* vertices with respect to S, which are vertices that do not have any node in S that would have them in their view (i.e. they are *head* vertices for graph G^S).

Let us first consider a DAG G that has a single *head* vertex v. This means that every other node of G is in the *view* of v. There is a temptation to just corrupt v, since it can see all the nodes, we can change its value accordingly and bias the output of extractor towards 0 or 1. But the extractor can just "ignore" v, i.e. not depend on v at all, or depend on it in a very "weak" way. To capture this we define the notion of influence. The influence of node v with respect to extractor/function f is defined as follows: imagine we sample every node but v uniformly at random: $\vec{x}_{\neg v} \leftarrow U$, and then we sample two independent uniform version of node v: $x_0, x_1 \leftarrow U$, we measure influence $\mathrm{Inf}_v^G(f)$ as (for formal definition see Definition 14):

$$\mathrm{Inf}_v^G(f) = \Pr\left[f(x_0, \vec{x}_{\neg v}) \neq f(x_1, \vec{x}_{\neg v})\right],$$

this probability is taken over randomness in choice of $x_0, x_1, \vec{x}_{\neg v}$. The reason why we resample x_0, x_1 from uniform distribution, instead of simply checking if there exist x_0, x_1 for which the value of the extractor changes, will be apparent later on.

We will proceed with impossibility result first:

Impossibility. Given DAG G, we will show that there exists a resilience threshold, which once exceeded allows to bias any function f.

The intuition behind the earlier, naive attack was that, if a node v has a view of all other nodes and it has non-negligible influence over the extractor, then we are done. We set $X_v = x_0$ or $X_v = x_1$ depending on which way we want to bias the output of the extractor f.

Notice that simply corrupting one node with influence does not guarantee bias: the problem is that even if v has a full power to flip output of the extractor, it has to know which way he is biasing the output. Simply imagine a DAG without any edges, and extractor just XORs all nodes: each node has a power to flip the output of the function, but it has to know all other nodes' inputs to actually bias the output of f.

Given function/extractor f let us consider its influence set V^f: set of nodes with influence non-negligible in n, where n is the size of the string each node produces.For now, let us assume that all other nodes have influence 0. Consider graph G^{V^f}, as defined before it is a graph defined on nodes V^f that maintains the *view* structure of the original graph. If there is a *head* node v in graph G^f that has all nodes from V^f in its view then we are done - just corrupt single node v, and since by definition v has non-negligible influence, and it sees all inputs, it has the power to bias the output of the function whichever way he wants.

What if there are multiple *head* vertices in G^{V^f} and none of them has full view? We can simply pick v to be one of the *heads* with the largest $|\mathrm{view}(v)|$, corrupt v and all nodes outside its view that still have influence: $R := V^f \setminus (\mathrm{view}(v) \cup \{v\})$. The idea is simple:

v has influence, and knows the values of all other nodes with influence[2], so it can bias the output of the function. One has to be slightly careful here, influence of v is defined over uniform distribution of everything else, so even that nodes in R are corrupted they have to be set to uniform values- the only purpose of corrupting those nodes is to *know* their value. As it will become apparent soon this is the best attack possible and we recommend keeping it in mind throughout the technical introduction.

There are three issues to resolve: 1. this strategy seems to be function specific, 2. we assumed that nodes outside V^f had zero influence, 3. is there a better strategy? Let us address the first issue, by defining resilience of the graph as follows:

$$\mathrm{res}(G) = \max_{S \subseteq V} \left[|S| - \max_{v \in S} |S \cap \mathrm{view}(v)| - 1 \right],$$

where V is a set of all nodes in graph G. If one looks at the attack above, it required exactly

$$|V^f| - \max_{v \in V^f} |V^f \cap \mathrm{view}(v)|$$

corruptions. Thus if number of corruptions is greater than $\mathrm{res}(G)$ we can bias any function/extractor.

For the second issue-the case where nodes outside of V^f have negligible but not 0 influence. The matter seems quite obvious: execute the attack as earlier, and since the influence of each node outside of V^f is negligible, they can only impact the bias of the output distribution by negligible factor, and when corrupting v we can predict the output of the function with $1 - \mathsf{negl}$ probability. More precisely, if we group the inputs of f into four input classes: the value at node v, values in the $\mathrm{view}(v)$, values in corruptions of the rest of influence set[3] r and the values of remaining nodes with negligible influence, one would be tempted to write: $\Pr[f(v, \mathrm{view}, r, U) = f(v, \mathrm{view}, r, U')] > 1 - \mathsf{negl}(n)$, and thus one would like to conclude that output of the function is basically known to v, even that v does not know the exact values outside of V^f. However, there is a delicate caveat here: influence of the nodes outside of V^f is defined with respect to uniform distribution of all nodes. This is precisely the reason why we pick values in $V^f \setminus (\mathrm{view}(v) \cup \{v\})$ as uniform, and even value in the node v is picked as choice between two uniform samples (one can think of picking a value at random, and we have a choice to reset it once). The distribution in v is not quite uniform, but we show that such "single-reset" source does not impact the influence of nodes outside V^f too much - to be precise we show that if the influence of the node measured over uniform distribution is ε, then the influence of that node counted over "single-reset" distribution is at most 2ε. And thus, we can still obtain that the output of the function is basically already determined from the point of view of node v, i.e. $\Pr[f(v, \mathrm{view}, r, U) = f(v, \mathrm{view}, r, U')] > 1 - \mathsf{negl}(n)$, even if v, view, r are sampled from this not-quite-uniform distribution. This concludes that the strategy of corrupting $|V^f| - \max_{v \in V^f} |V^f \cap \mathrm{view}(v)|$ nodes works. By the previous discussion it is also clear that if number of corruptions exceeds $\mathrm{res}(G)$ then every function can be biased, and thus it is not possible to extract from such source. Formal statement can be found in Corollary 1.

[2]Node v "sees" all *honest* nodes in his *view*, and all remaining nodes (set R) are corrupted, and thus known to v.

[3]That is values in $V^f \setminus (\mathrm{view}(v) \cup \{v\})$.

Finally we address the third issue: can we do better? The answer is negative. We can build extractor that is resilient to res(G) corruptions, which completes the picture.

Extraction. We have discussed set of nodes with non-negligible influence V^f, and we have established that for any function it suffices to corrupt $|V^f - \max_{v \in V^f} |V^f \cap \text{view}(v)|$ many nodes to bias it. Let us begin the quest for building the extractor with finding[4] the set S that maximizes resilience, i.e.

$$|S| - \max_{v \in S} |S \cap \text{view}(v)| - 1 = \text{res}(G).$$

When building extractor, S has to be the set of nodes that have high impact on the output, while nodes outside of S should have negligible impact. Let us simplify this task a bit: we will make extractor depend only on nodes in S and completely ignore nodes outside of S.

The corrupted nodes introduce a lot of correlations, and a natural tool to break them is the two-source non-malleable extractor: As long as X, Y are independent and have high entropy, output of $\mathsf{2nmExt}(X, Y)$ remains indistinguishable from uniform distribution even given the outputs of the extractor on correlated/tampered inputs. More precisely, let $X', X'', ..., X^{(t)}$ be arbitrarily correlated with X but not equal to X, symmetrically define $Y', ..., Y^{(t)}$, importantly $X', ..., X^{(t)}$ do not depend on Y, and $Y', ..., Y^{(t)}$ do not depend on X, then

$$\mathsf{2nmExt}(X, Y) \approx U \text{ even given } \left[\mathsf{2nmExt}(X', Y'), \mathsf{2nmExt}(X'', Y''), .., \mathsf{2nmExt}(X^{(t)}, Y^{(t)})\right].$$

Using the above tool, a naive idea would be to run the extractor over all pairs of nodes in the set S (defined earlier as set that maximizes resilience) and just XOR the outputs. However, this approach has a number of technical issues, which will become apparent soon.

One issue is: if $u \in \text{view}(v)$ then $\mathsf{2nmExt}(X_u, X_v)$ does not bring much to the table, as a single corruption of v gives full control over such pair. Similarly, if $u, w \in \text{view}(v)$ then v depends on both u, w and therefore $\mathsf{2nmExt}(u, w) \oplus \mathsf{2nmExt}(u, v) \oplus \mathsf{2nmExt}(v, w)$ is fully controlled by a single corruption in v. Strictly speaking, these are not attacks, but soon it will be clear they cause unnecessary technical difficulties.

To make technical proofs simpler instead of XOR-ing over all pairs, let us just XOR over *unrelated* pairs, that is, w, u such that $w \notin \text{view}(u)$ and $u \notin \text{view}(w)$ and $w \neq u$.

Notice that if we want to match impossibility result then we do not have much to work with, adversary is controlling most of the nodes in S. We are only guaranteed that there is one pair of nodes a, b, such that adversary does not have both a, b in a single view, but potentially "sees" each of them separately. This is a crucial information: if adversary can corrupt only res(G) many nodes, and S maximizes resilience, then there must be one node outside[5] of view(v) that is not corrupted, thus our single pair $a, b \in S$ for which $\forall v \in S, (a \notin \text{view}(v))$ or $(b \notin \text{view}(v))$.

But this pair a, b is a great news, it will work perfectly with the $\mathsf{2nmExt}$. We want to XOR over all unrelated pairs:

$$\bigoplus_{(w,u) \text{ unrelated}} \mathsf{2nmExt}(X_w, X_u).$$

[4] We'll discuss the task of actually finding the set later on.

[5] Please recall the corruption strategy: given set S we singled out one node $v \in S$ with the largest $|\text{view}(v) \cap S|$, and proceeded to corrupt v, and everything outside its view $S \setminus \text{view}(v)$.

Let us consider a few groups of elements in above sum with respect to their relationship to our uncorrupted pair a, b.

1. First group consists only of nodes a, b, those nodes are independent and not corrupted, X_a, X_b are high entropic and $\mathsf{2nmExt}(X_a, X_b)$ will provide us with uniform output. IMPORTANTLY note that we do not need to know a, b, it is sufficient to know that they are somewhere in the XOR.

2. All $w, u \in S$ such that $a, b \notin \text{view}(w) \cup \text{view}(u) \cup \{w, u\}$, those will bring nothing important to the table as X_w, X_u will be independent of X_a, X_b and thus XOR of $\mathsf{2nmExt}(X_w, X_u)$ over all pairs in this group will remain independent of $\mathsf{2nmExt}(X_a, X_b)$

3. For the third group consider w, u such that $a \in \text{view}(u) \cup \{u\}$, and $b \notin \text{view}(u) \cup \text{view}(w) \cup \{w, u\}$. For thisms group $\mathsf{2nmExt}(X_w, X_u)$ depends only on X_a, and thus

$$\bigoplus_{\text{elements in this group}} \mathsf{2nmExt}(X_w, X_u)$$

   can be seen as single small leakage from X_a, we can reveal it and fix it, which will reduce entropy of X_a slightly, but well within the parameters that allow us to use two-source non-malleable extractor. Note that we will reveal whole

$$L(X_a) = \bigoplus_{\text{elements in this group}} \mathsf{2nmExt}(X_w, X_u)$$

   as one single leakage instead of revealing each extractor output separately:

$$\left\{ \mathsf{2nmExt}(X_w, X_u) \right\}_{(u,w) \text{ in this group}}$$

   else the loss in the entropy of X_a would be too large. We have to be delicate here, and reveal X_w as it randomizes the leakage function, and not revealing it would lead to correlations between leakages from X_a and X_b (see next group). However, we only need to reveal X_w when w cannot "see" a, i.e if $a \notin \text{view}(w) \cup \{w\}$, else we don't need to reveal X_u or X_w, as in this case since both u, w can "see" a and thus neither u, w or their parents can "see" b (since $\forall v \in S, (a \notin \text{view}(v))$ or $(b \notin \text{view}(v))$) and thus none of those variables will show up in the leakage from X_b group and there won't be any correlation between leakage caused by those variables.

4. Same happens for the forth group consisting of w, u such that $b \in \text{view}(u) \cup \{u\}$, and $a \notin \text{view}(u) \cup \text{view}(w) \cup \{w, u\}$. This group constitutes leakage from X_b. And since the same X_w may be used to randomize both leakage from X_a and X_b, those two random variables will not be independent given the leakages unless we reveal all X_w such that $a, b \notin \text{view}(w) \cup \{w\}$.

5. Finally, the fifth group: w, u such that $a \in \text{view}(w) \cup \{w\}$ and $b \in \text{view}(u) \cup \{u\}$. Note that since $\forall_{v \in S}(a \notin \text{view}(v))$ or $(b \notin \text{view}(v))$ we know that there is no "cross-over", i.e $a \notin \text{view}(u) \cup \{u\}$ and $b \notin \text{view}(w) \cup \{w\}$. This group is taken care of by two-source non-malleable extractor that guarantees that $\mathsf{2nmExt}(X_w, X_u)$ are all independent of $\mathsf{2nmExt}(X_a, X_b)$.

6. Note that there is no other pairs in our XOR. In particular, we are not having that $\mathsf{2nmExt}(X_a, X_v) + \mathsf{2nmExt}(X_b, X_v)$ where $a, b \in \text{view}(v)$ as X_v could depend on both X_a and X_b and we could not model it as a leakage or use the non-malleable extractor property.

We are nearly done, there are two issues to take care of, one is actually quite serious.

First, remember that two-source non-malleable extractor requires $X' \neq X$ and $Y' \neq Y$. This can be taken care of by a simple trick: just append id of the node:

$$\bigoplus_{(w,u) \text{ unrelated}} \mathsf{2nmExt}(X_w \| w \ , \ X_u \| u).$$

The second issue is far more serious. Remember the definition says:

$$\mathsf{2nmExt}(X, Y) \approx U \text{ even given } \left[\mathsf{2nmExt}(X', Y'), \mathsf{2nmExt}(X'', Y''), .., \mathsf{2nmExt}(X^{(t)}, Y^{(t)})\right].$$

Note that variables correlated to X are always first input, while variables correlated to Y are second input, we have *no* guarantee on

$$\mathsf{2nmExt}(X, Y) \overset{?}{\approx} U \text{ given } \mathsf{2nmExt}(Y', X').$$

To combat this issue we need one more definition and a crucial observation. For $w, u \in S$ we will call (w, u) a *headless pair* if there does not exist $v \in S$ such that $w, u \in \text{view}(v) \cup \{v\}$. We call this pair headless because in particular there is no *head* v in S that has both nodes in its *view*. Notice that we are guaranteed an honest pair a, b for which $\forall v \in S, (a \notin \text{view}(v))$ or $(b \notin \text{view}(v))$, which means that a, b is *headless* pair. Crucial observation: if we only XOR over *headless* pairs instead of *unrelated* pairs, we have the following property: for every (w, u) *headless* pair and any *head* vertices in S - h, g such that $w \in \text{view}(h) \cup \{h\}$ and $u \in \text{view}(g) \cup \{g\}$ and there is no cross-over: u is not "seen" by h and w is not "seen" by g. We can enforce ordering on all *heads* of S, and this ordering will translate into an ordering of elements within each *headless* pair: given (w, u) the smaller of w, u is the one with the smaller *head* (note that each of them can have multiple heads, just pick the smallest of them).

Our extractor is XOR-ing over *headless* pairs (w, u) such that $w \prec u$. As this is a subset of *unrelated* pairs the previous discussion about leakages and non-malleable extractors still holds. For formal statement and proof one can refer to Theorem 9, followed by Corollary 2 and 3.

A small remark, when comparing this to the perfect extraction case: if S is the largest *unrelated* set then $\text{res}(G) \geq |S| - \max_{v \in S} |S \cap \text{view}(v)| - 1 = |S| - 1$, but $\text{res}(G)$ might be much larger than that, which leads to a clear gap between perfect and statistical extraction. Simple example would be: Figure 2. The largest *unrelated* set there has two elements so we can withstand at most 1 corruption if we want perfect coin. However if we take $S = \{B, C, D, F, G, H\}$ the *resilience* of this graph is 3, we are resilient to 3 corruptions.

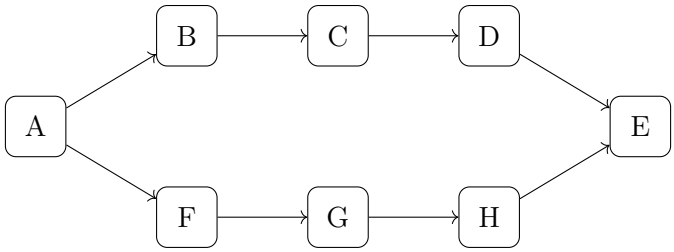

Figure 2: DAG with a small *unrelated* set (size 2), and larger resilience (equal 3).

Algorithm to find the most resilient subset of nodes. The above extractor relied on finding the set that maximizes the resilience of the graph. We show how to do it efficiently. There might be multiple such maximal sets. We show that among them there must be a set that does not truncate the *view* of nodes, to be more precise: There exists a set S that maximizes resilience of the graph and S can be written as union of complete *views* of its heads: $\exists_{v_1,\ldots,v_k},\ S = \bigcup_i (\text{view}\, v_i \cup \{v_i\})$. This significantly narrows down the space we have to search, and we can do it in time linear in the size of the graph.

What if nodes are not uniform? Notice that at no point we really needed uniformness of the nodes, we just needed honest *headless* pair X_a, X_b to have high enough entropy to work with non-malleable extractor, this puts a constant entropy rate requirement on the sources (constant is quite close to 1 and depends on size of the graph, but it clearly is far from uniform requirement). Also, the size of the output will have an impact on the entropy requirements (note that we have to handle both extraction with a longer output, and the entropy loss of sources as leakages become longer).

3 Preliminaries

3.1 Notation

We use uppercase roman letters such as X and Y to denote random variables. $\mathcal{U}_n$ denotes the uniform distribution over $\{0,1\}^n$. For any $T \subseteq \{0,1\}^n$, we use $\mathcal{U}_T$ to refer to the uniform distribution over the set T. For a set S and random variable $\mathbf{X}$, we write $\mathbf{X} \sim S$ to denote that $\mathbf{X}$ is supported on S and $r \sim \mathbf{X}$ to denote that r is sampled according to $\mathbf{X}$. Finally, $r \leftarrow S$ denotes that r is uniformly sampled from the set S. For any $m \in \mathbb{N}$, we write $[m]$ for the set $\{1, 2, \ldots, m\}$. For any set $S \subseteq [N]$ and $r \in (\{0,1\}^n)^{|S|}$ we use the notation $f(X_S = r, X_{[N]\setminus S})$ to define another function on variables $\{X_i : i \in [N] \setminus S\}$, obtained by fixing $X_S = r$ in f. For any two strings $x, y \in \{0,1\}^n$, we use the notation $x \| y$ to denote their concatenation.

3.2 Basic probability theory

Now we will proceed to introduce a few useful definitions and related lemmas. Let us start by stating a Markov like inequality which will be useful.

Lemma 1. *Let Z be a random variable that takes values from the range $[0,1]$ and its expectation $\mathbb{E}[Z] \geq \mu$. Then for any $0 \leq p < 1$ we have $\Pr[Z \leq p] \leq (1-\mu)/(1-p)$.*

Proof. Consider the random variable $Y = 1 - Z$. Note, Y is also a random variable that takes value from $[0,1]$ and $\mathbb{E}[Y] \leq 1 - \mu$. Applying Markov's inequality on Y we get $\Pr[Y \geq 1-p] \leq (1-\mu)/(1-p)$ and from here replacing $Y = 1 - Z$ our proof follows. □

Definition 7 (Support)**.** *For a random variable $\mathbf{X} \sim \{0,1\}^n$, we say support of $\mathbf{X}$,*

$$\mathsf{supp}(\mathbf{X}) := \left\{x \in \{0,1\}^n : \Pr[\mathbf{X} = x] \neq 0\right\} .$$

Definition 8 (Min-entropy)**.** *For a random source $\mathbf{X} \sim \{0,1\}^n$, min-entropy of $\mathbf{X}$ (denote it as $\mathbf{H}_\infty(\mathbf{X})$) is defined as,*

$$\mathbf{H}_\infty(\mathbf{X}) := \min_{x \in \mathsf{supp}(\mathbf{X})} \log \frac{1}{\Pr[\mathbf{X} = x]} .$$

We say $\mathbf{X}$ is an (n,k) source if $\mathbf{X} \sim \{0,1\}^n$ and $\mathbf{H}_\infty(\mathbf{X}) \geq k$.

Next we will state a lemma on conditional min-entropy of a distribution. Informally, the lemma asserts that for any two distributions $\mathbf{X}$ and $\mathbf{Y}$, if $\mathbf{X}$ has some entropy, then conditioning on a random $y \sim \mathbf{Y}$ does not significantly reduce the entropy with high probability.

Lemma 2 (Min-entropy chain rule [MW97, Lemma 5])**.** *$\mathbf{X} \sim \Omega$ and $\mathbf{Y} \sim \Omega'$ be two distributions so that $\mathbf{Y}$ takes at most ℓ values from Ω'. Then, for any $\varepsilon > 0$,*

$$\Pr_{y \sim \mathbf{Y}}[\mathbf{H}_\infty(\mathbf{X} \mid \mathbf{Y} = y) \geq \mathbf{H}_\infty(\mathbf{X}) - \log \ell - \log(1/\varepsilon)] \geq 1 - \varepsilon .$$

The statistical distance is a standard measure for the proximity of two random variables sampled from the same set.

Definition 9 (Statistical Distance)**.** *Given two random variables $\mathbf{X}, \mathbf{Y} \sim \Omega$, we define the statistical distance as*

$$\Delta(\mathbf{X}\ ;\ \mathbf{Y}) := \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[\mathbf{X} = \omega] - \Pr[\mathbf{Y} = \omega]|.$$

We shorthand $\Delta((\mathbf{X}, \mathbf{Z})\ ;(\mathbf{Y}, \mathbf{Z}))$ by $\Delta(\mathbf{X}\ ;\ \mathbf{Y} \mid \mathbf{Z})$ and $\Delta(\mathbf{X}\ ;\ \mathbf{Y}) \leq \varepsilon$ by $\mathbf{X} \approx_\varepsilon \mathbf{Y}$.

The following lemma asserts that if $\mathbf{X}, \mathbf{Y}$ are statistically close, then $f(\mathbf{X}), f(\mathbf{Y})$ are also statistically close, for any function f.

Lemma 3 (Data processing inequality [Vad12, Lemma 6.3])**.** *For any possibly randomized function $f : \{0,1\}^n \to \{0,1\}^*$ and random sources $\mathbf{X}, \mathbf{Y} \sim \{0,1\}^n$, we have $\Delta(f(\mathbf{X})\ ;\ f(\mathbf{Y})) \leq \Delta(\mathbf{X}\ ;\ \mathbf{Y})$.*

3.3 Extractors and non-malleable extractors

In this section we define extractors and non-malleable extractors.

Extractors are deterministic functions that take a weak source as input and outputs a distribution that is close to uniform. Formally the definition is as follows.

Definition 10 (Extractor for a class of sources)**.** *Let $\mathcal{X}$ be a class of sources supported on a set $\mathcal{S}$. The function $\mathsf{Ext} : \mathcal{S} \to \{0,1\}^m$ is a ε-*extractor for $\mathcal{X}$ *if for all $\mathbf{X} \in \mathcal{X}$ we have $\mathsf{Ext}(\mathbf{X}) \approx_\varepsilon \mathcal{U}_m$.*

Two source non-malleable extractors were defined by Cheraghchi and Guruswami in [CG14]. Informally, a two-source non-malleable extractor is a function that, on two *weak* input sources, outputs a distribution that stays close to uniform even when the output on any tampered version of the inputs is known. We will need a multi-tampering version of the above which was first introduced in [CGL20].

Definition 11 (Two source non-malleable extractor [CGL20])**.** *A function $\mathsf{2nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is called an $(\ell, k_1, k_2, \varepsilon)$-two source non-malleable extractor if for every pair of independent sources $\mathbf{X}, \mathbf{Y} \sim \{0,1\}^n$ so that $\mathbf{H}_\infty(\mathbf{X}) \geq k_1$ and $\mathbf{H}_\infty(\mathbf{Y}) \geq k_2$ and for every family of tampering functions $g_{i1}, g_{i2} : \{0,1\}^n \to \{0,1\}^n$ where for all $i = 1, \ldots, \ell$ at least one of g_{i1} and g_{i2} has no fixed points,*

$$\Delta(\mathsf{2nmExt}(\mathbf{X}, \mathbf{Y}) \; ; \; \mathcal{U}_m \mid \mathsf{2nmExt}(g_{11}(\mathbf{X}), g_{12}(\mathbf{Y})), \ldots, \mathsf{2nmExt}(g_{\ell 1}(\mathbf{X}), g_{\ell 2}(\mathbf{Y}))) \leq \varepsilon \; .$$

If $k_1 = k_2 = k$, we call it a (ℓ, k, ε)-two source non-malleable extractor.

Lemma 4 ([AOR+22, Lemma 4])**.** *Let $\mathsf{2nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ be an $(\ell, k_1, k_2, \varepsilon)$-two source non-malleable extractor and $\boldsymbol{R}$ be an arbitrary distribution on some set $\mathcal{R}$. Then for every family of functions $g_{i1}, g_{i2} : \{0,1\}^n \times \mathcal{R} \to \{0,1\}^n$ so that for every $R \in \mathcal{R}$ at least one of $g_{i1}(., R)$ and $g_{i2}(., R)$ has no fixed points, it holds that,*

$$\Delta(\mathsf{2nmExt}(\mathbf{X}, \mathbf{Y}); \mathcal{U}_m \mid \mathsf{2nmExt}(g_{11}(\mathbf{X}, \boldsymbol{R}), g_{12}(\mathbf{Y}, \boldsymbol{R})), \cdots, \mathsf{2nmExt}(g_{\ell 1}(\mathbf{X}, \boldsymbol{R}), g_{\ell 2}(\mathbf{Y}, \boldsymbol{R})), \boldsymbol{R})$$

is at most ε for every independent sources $\mathbf{X}, \mathbf{Y} \sim \{0,1\}^n$ with $\mathbf{H}_\infty(\mathbf{X}) \geq k_1$, $\mathbf{H}_\infty(\mathbf{Y}) \geq k_2$ so that both $\mathbf{X}$ and $\mathbf{Y}$ are independent of $\boldsymbol{R}$.

Finally we will state a few constructions of two source non-malleable extractors.

Proposition 1 ([CGL20])**.** *There exists a constant $\gamma > 0$ so that for all $n > 0$ and $\ell \leq n^\gamma$ there is an explicit (ℓ, k, ε)-two source non-malleable extractor $\mathsf{CGL} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ where $k \geq n - n^\gamma$, $\varepsilon \leq 2^{-n^{\Omega(1)}}$ and $m = n^{\Omega(1)}$.*

Proposition 2 ([ACO23])**.** *For every constant ℓ, for all $n > 0$ there exists an explicit $(\ell, k_1, k_2, \varepsilon)$-two source non-malleable extractor $\mathsf{ACO} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ with $k_1 \geq (\log n)^{O(1)}$, $k_2 \geq \left(1 - \frac{1}{2\ell+3}\right)n$, $m = \Omega(k_1)$ and $\varepsilon \leq 2^{-\Omega(k_1^c)}$ for some $c < 1/2$.*

4 Randomness extraction from SHEDAG sources

4.1 Head vertices, and parents

In this section we present a few more definitions related to DAGs that will be useful in our impossibility arguments and extractor constructions.

Definition 12 (Head vertex)**.** *Given a DAG $G = (V, E)$, we say that $v \in V$ is a* head vertex *(in G) if* out-deg$(v) = 0$*. We denote the set of all head vertices in G by* Head(G)*.*

Definition 13 (Parents of a vertex)**.** *Given a DAG $G = (V, E)$ and a vertex $v \in V$, the set of* parents *of v, denoted* parents(v)*, is the set of vertices u such that there is a path from v to u.*

Note that for any $u, v \in V$ we have $u \in \text{view}(v)$ if and only if $v \in \text{parents}(u)$.

4.2 Zero-error randomness extraction

Firstly, we proceed to give a complete characterization of the scenario under which perfect-coin extraction with zero error is achievable. The key idea is that, if we have a large number of nodes that cannot *see* each other, meaning that none of the nodes are in view of the rest, then the adversary has no way to corrupt the nodes in such a way that extraction is impossible.

Observation 1. *For every function $f : (\{0,1\}^n)^N \to \{0,1\}$ there exists a source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ so that each $\mathbf{X}_i$ are independent (n, k) source with $k < n$. By induction over N we can show it. If $N = 1$ it is indeed true because if $f : \{0,1\}^n \to \{0,1\}$ be any function then one of $|f^{-1}(0)|$ and $|f^{-1}(1)|$ is at least 2^{n-1}. For $a \in \{0,1\}$ say $|f^{-1}(a)| \geq 2^{n-1}$ and $\mathbf{X}_1$ to be flat source over $|f^{-1}(a)|$. Then $f(\mathbf{X}_1)$ is constant and $\mathbf{X}_1$ has min-entropy at least $n - 1$.*

Let $f : (\{0,1\}^n)^{\ell+1} \to \{0,1\}$ be a function. Pick any $x \in \{0,1\}^n$ and note that by induction hypothesis, there exists a source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_\ell)$ so that $\Delta(f(\mathbf{X}, x)\ ;\ \mathcal{U}_1) > 0$. Consider the source $\mathbf{X}' = (\mathbf{X}'_{[\ell]} = \mathbf{X}, \mathbf{X}'_{\ell+1} = \mathcal{U}_n)$. Note that

$$\Pr[f(\mathbf{X}') = 0] = 2^{-n} \Pr[f(\mathbf{X}, x) = 0] + (1 - 2^{-n}) \Pr[f(\mathbf{X}, \mathcal{U}_S) = 0]\ ,$$

where $S = \{0,1\}^n \setminus \{x\}$. Since, $\Pr[f(\mathbf{X}, x) = 0] \neq 1/2$, it is not possible that both $\Pr[f(\mathbf{X}') = 0] = 1/2$ and $\Pr[f(\mathbf{X}, \mathcal{U}_S) = 0] = 1/2$. Hence by induction our proof follows.

As we proved that perfect extraction from general (n, k, G, t)-SHEDAG sources is impossible for $k < n$, even when G is an empty graph (i.e. each block is independent) we start with analyzing the condition when the zero error extraction from (n, G, t)-SHEDAG source (when honest sources are independent and uniform) is possible.

4.2.1 Feasibility

At first we will show that when the base graph G has a unrelated set of size at least $t + 1$, it is possible to extract from any SHEDAG source with at most t corruptions. Formally we state our first main theorem as follows.

Theorem 4. *For all $n \in \mathbb{N}$, fixed $N \in \mathbb{N}$ and $t \leq N-1$ the following holds: Let $G = (V, E)$ be any directed acyclic graph with $V = [N]$ and G has a unrelated set of size at least $t+1$. Then there exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^n$ so that for every (n, G, t)-SHEDAG source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$, we have $\mathsf{Ext}(\mathbf{X}) = \mathcal{U}_n$.*

Proof. Let, $T \subseteq V$ be an unrelated set of G of size $t+1$. By our assumption such T would exist and say $T = \{v_1, \ldots, v_{t+1}\}$. Define,

$$\mathsf{Ext}(\mathbf{X}_1, \ldots, \mathbf{X}_N) = \bigoplus_{u \in T} \mathbf{X}_u \ .$$

Since there are at most t many corrupted blocks, at least one of $\mathbf{X}_{v_1}, ..., \mathbf{X}_{v_{t+1}}$ is honest. Without loss of generality, say $\mathbf{X}_{v_1} = \mathcal{U}_n$ and independent from $\mathbf{X}_{v_2}, \ldots, \mathbf{X}_{v_{t+1}}$.

This implies, under every fixing of $(\mathbf{X}_{v_2}, \ldots, \mathbf{X}_{v_{t+1}}) = (y_2, \ldots, y_{t+1})$ for $y_2, \ldots, y_{t+1} \in \{0,1\}^n$, we have $\bigoplus_{u \in T} \mathbf{X}_u | (\mathbf{X}_{v_2} = y_2, \ldots, \mathbf{X}_{v_{t+1}} = y_{t+1})$ is still uniform. Hence, $\bigoplus_{u \in T} \mathbf{X}_u$ is uniform over $\{0,1\}^n$ and it completes our proof. □

4.2.2 Impossibility

We will next prove that this bound on size of unrelated set is actually tight for zero error extraction from (n, G, t)-SHEDAG source. That is, if maximum unrelated set of the base graph G has size at most t, then it is impossible to extract from SHEDAG source with t corruptions.

Since, Head(G) is a unrelated set of G, the number of head vertices of G is at most t when the size of maximum unrelated set of G is at most t . Now the main idea is, if the number of corrupted vertices is t then the adversary can *control* vertices of the maximum unrelated set and make the output of the extractor *biased*.

At first, we make an observation to the adversary behavior. If $\mathcal{A}$ corrupts a vertex v which is under view of another corrupted vertex u, then by corrupting only u the adversary can induce non-zero bias on the extractor's output.

Lemma 5. *For all $n \in \mathbb{N}$, fixed $N \in \mathbb{N}$ and $t < \ell \leq N$ the following holds: Let $G = (V, E)$ be any directed acyclic graph with $V = [N]$ and G has at most t many unrelated sets. Let $f : (\{0,1\}^n)^N \to \{0,1\}$ be any function. Then the following two are equivalent:*

(i) There exists a (n, G, ℓ)-SHEDAG source $\mathbf{X}$ so that $\Delta(f(\mathbf{X}) \ ; \ \mathcal{U}_1) > 0$.

(ii) There exists a (n, G, t)-SHEDAG source $\mathbf{X}$ so that $\Delta(f(\mathbf{X}) \ ; \ \mathcal{U}_1) > 0$.

Proof. We focus on proving that (i) $\implies$ (ii), as the other direction is immediate. Let $\mathbf{X}$ be a (n, G, ℓ)-SHEDAG source, for which $\Delta((f(\mathbf{X}) \ ; \ \mathcal{U}_1) > 0$. Without loss of generality we can assume that $\Pr[f(\mathbf{X}) = 1] > 1/2$. Define a set, $V_{\mathcal{A}} \subseteq V$, with $|V_{\mathcal{A}}| \leq \ell$, to be the set of vertices in the source $\mathbf{X}$ which are corrupted by the adversary $\mathcal{A}$.

Assume $|V_{\mathcal{A}}| > t$. Then there must exist two distinct vertices $i, j \in V_{\mathcal{A}}$, such that $j \in \text{view}(i)$. Define set $R = V \setminus \{i, j\}$, and let $\mathbf{X}_R$ be a marginal distribution of source $\mathbf{X}$ on coordinates in the set R. Let us consider two cases:

Case 1: There exist $y_0, y_1, z \in \{0,1\}^n$, such that[6]

$$\Pr_{r\sim \mathbf{X}_R}[f(X_i = y_0, X_j = z, X_R = r) = 1] \neq \Pr_{r\sim \mathbf{X}_R}[f(X_i = y_1, X_j = z, X_R = r) = 1] .$$

Therefore both quantities of the above cannot be equal to $1/2$. Without loss of generality let $\Pr_r[f(X_i = y_0, X_j = z, X_R = r) = 1] = p \neq 1/2$. Now consider the $(n, G, \ell-1)$-SHEDAG source $\tilde{\mathbf{X}}$ defined as:

$$\tilde{\mathbf{X}} = (\tilde{\mathbf{X}}_i = \mathcal{U}_n, \tilde{\mathbf{X}}_j = \mathcal{U}_S, \tilde{\mathbf{X}}_R = \mathbf{X}_R) \tag{1}$$

where $S = \{0,1\}^n \setminus \{z\}$, note that the node i is not corrupted here thus $\ell - 1$ corruptions. If we have, $\Pr[f(\tilde{\mathbf{X}}) = 1] = \Pr[f(\mathcal{U}_n, \mathcal{U}_S, \mathbf{X}_R) = 1] \neq 1/2$, then we are done. Assume $\Pr[f(\mathcal{U}_n, \mathcal{U}_S, \mathbf{X}_R) = 1] = 1/2$, consider the source $\mathbf{X}'$ defined as follows:

$$\mathbf{X}'_i = \begin{cases} y_0 & \text{if } \mathbf{X}'_j = z \\ \mathcal{U}_n & \text{otherwise} \end{cases}$$

and $\mathbf{X}'_j = \mathcal{U}_n,\ \mathbf{X}'_R = \mathbf{X}_R$, notice that since $j \in \text{view}(i)$, $\mathbf{X}'_i$ can depend arbitrarily on $\mathbf{X}_j$. Note that it's a $(n, G, \ell-1)$-SHEDAG source as now the node j is not corrupted. We have:

$$\begin{aligned}
&\Pr\big[f(\mathbf{X}'_i, \mathbf{X}'_j, \mathbf{X}'_R) = 1\big] \\
&= \Pr\big[\mathbf{X}'_j = z\big] \cdot \Pr\big[f(\mathbf{X}'_i, \mathbf{X}'_j, \mathbf{X}_R) = 1 \mid \mathbf{X}'_j = z\big] \\
&\quad + \Pr\big[\mathbf{X}'_j \neq z\big] \cdot \Pr\big[f(\mathbf{X}'_i, \mathbf{X}'_j, \mathbf{X}_R) = 1 \mid \mathbf{X}'_j \neq z\big] \\
&= 2^{-n} \cdot \Pr[f(y_0, z, \mathbf{X}_R) = 1] + (1 - 2^{-n}) \cdot \Pr[f(\mathcal{U}_n, \mathcal{U}_S, \mathbf{X}_R) = 1] \\
&= 2^{-n} \cdot p + (1 - 2^{-n}) \cdot \tfrac{1}{2} \\
&\neq \tfrac{1}{2}
\end{aligned}$$

The second equality holds because $(\mathbf{X}'_j \mid \mathbf{X}'_j \neq z) \equiv \mathcal{U}_S$ where $S = \{0,1\}^n \setminus \{z\}$ The third equality follows from our assumption that $\Pr[f(\tilde{\mathbf{X}}) = 1] = 1/2$, where $\tilde{\mathbf{X}}$ is as defined in 1. Therefore, we have $\Delta(f(\mathbf{X}') \,;\, \mathcal{U}_1) > 0$.

Case 2: For all $z \in \{0,1\}^n$ and for all $y_0, y_1 \in \{0,1\}^n$ we have,

$$\Pr_{r\sim \mathbf{X}_R}[f(X_i = y_0, X_j = z, X_R = r) = 1] = \Pr_{r\sim \mathbf{X}_R}[f(X_i = y_1, X_j = z, X_R = r) = 1] .$$

This implies, for all $y \in \{0,1\}^n$ the value of $\Pr_{r\sim \mathbf{X}_R}[f(X_i = y, X_j = z, X_R = r) = 1]$ depends only on z. Now define the $(n, G, \ell-1)$-SHEDAG source $\mathbf{X}'$ in the following way:

$$\mathbf{X}' = (\mathbf{X}'_i = \mathcal{U}_n,\ \mathbf{X}'_j = \mathbf{X}_j,\ \mathbf{X}'_R = \mathbf{X}_R).$$

Then clearly we will have $\Pr[f(\mathbf{X}) = 1] = \Pr[f(\mathbf{X}') = 1] \neq \frac{1}{2}$.

Iteratively we can apply the above process, until we will have that $|V_{\mathcal{A}}| \leq t$ and $V_{\mathcal{A}}$ is a set of independent vertices in G. This concludes our proof. □

[6]Note that such triple might not exist, the crucial observation is that the influence of the node is defined over uniform distribution of nodes in the view, while $\mathbf{X}_R$ might have arbitrary distribution.

Using the above lemma, we are ready to prove the main result of impossibility of extraction with zero error.

Theorem 5. *For all $n \in \mathbb{N}$, fixed $N \in \mathbb{N}$ and $t \leq N$ the following holds: Say $G = (V, E)$ be any directed acyclic graph with $V = [N]$ and $f : (\{0,1\}^n)^N \to \{0,1\}$ be any function. Further assume that maximum unrelated set of G has size t. Then there exists a (n, G, t)-SHEDAG source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$, such that $\Delta(f(\mathbf{X}) \; ; \; \mathcal{U}_1) > 0$.*

Proof. Fix a (n, G, N)-source $\mathbf{X}$ so that $\Delta(f(\mathbf{X}) \; ; \; \mathcal{U}_1) = 1$. We can always find such a source by picking $x \in (\{0,1\}^n)^N$ so that $f(x) = 0$ (if such x does not exist, pick x so that $f(x) = 1$) and fixing $\mathbf{X} = x$. Then by applying Lemma 5 the proof follows.

□

4.2.3 Algorithm for locating unrelated sets with maximum size

As discussed previously, to extract randomness with zero-error against as many corruptions as possible, the primary goal is to find the largest number of k such that there exist unrelated set of size k.

Observation 2. *If $G = (V, E)$ be a directed acyclic graph, then reachability in G induces a partial order in V. Formally $(V, \preccurlyeq)$ is a partially ordered set, where for $u, v \in V$ we have $u \preccurlyeq v$ if and only if there is a path from u to v in G. Here $A \subseteq V$ is an anti-chain if and only if for all $x, y \in S$ there is no path from x to y and from y to x. Hence, finding maximum unrelated set in G is same as finding maximum anti-chain in $(V, \preccurlyeq)$.*

In [FRS99], a polynomial time algorithm was given to find the maximum antichain in a partially ordered set (poset).

Theorem 6 (Locating maximum antichain [FRS99])**.** *Let $(P, \preccurlyeq)$ be a partially ordered set. There is an $O(k|P|^2)$ time algorithm deciding whether the size of maximum antichain of P is at most k. Additionally, if the size of maximum antichain is exactly k, this algorithm can be adapted to find the maximum antichain in $O(k|P|^2)$ time.*

The construction is based on Dilworth's Theorem [Dil50], which states that the maximum size of an antichain of a poset P equals the minimum number of chains required to cover P.

At first corresponding to each vertex $u \in V$ we make a list L_u so that $v \in L_u$ if and only if there is a path from u to v. This can be done in $O(|V|^3)$ time (by BFS from each vertex). Then by Theorem 6 we find the maximum antichain $\mathcal{C}$, the vertices in $\mathcal{C}$ thus form an unrelated set with maximum size. If the number of corruptions is less than $|\mathcal{C}|$, by Theorem 4, it is possible to extract from sources at vertices in $\mathcal{C}$ to obtain perfect randomness.

4.3 Randomness extraction with negligible error

4.3.1 Impossibility

In this section we will see the condition when we cannot extract from SHEDAG source with *small* error. Before diving into the main theorem, we will first define the notion of influence of a node of G on a function.

Definition 14 (Influence)**.** *$G = (V, E)$ be a directed acyclic graph with $V = [N]$ and $f : (\{0,1\}^n)^N \to \{0,1\}^n$ be any function. Then influence of i-th coordinate (or i-th vertex) on f in presence of G, denoted as $\mathrm{Inf}_i^G(f)$ is:*

$$\Pr_{\substack{r \leftarrow (\{0,1\}^n)^{N-1} \\ x_0 \leftarrow \{0,1\}^n \\ x_1 \leftarrow \{0,1\}^n}} \left[f(X_i = x_0, X_{S^i} = r) \neq f(X_i = x_1, X_{S^i} = r) \right]$$

where $S^i = V \setminus \{i\}$.

Now, we need another definition of the graph induced by influence vertices with respect to reachability in the original graph. We will state a more general definition.

Definition 15 (View preserving graph with a subset of vertices)**.** *Given a directed $G = (V, E)$ acyclic graph with $V = [N]$ and $S \subseteq V$ be any non-empty subset of the vertices. We construct a graph H with the vertex set S (denote it by $H(S) = (S, E(S))$) by the following Algorithm 1.*

Simply speaking, we remove vertices that is not in S, and we add edges in such a way that if one vertex was in the view of another, this relation will be preserved in the new graph.

Algorithm 1 Construction of the view preserving graph with vertex set S.

1: Start with (V', E') where $V' = V$ and $E' = E$.
2: **while** There is $v \in V'$ so that $v \notin S$ **do**
3: Define $\mathbf{in}(v) = \{u \in V' : (u, v) \in E'\}$ and $\mathbf{out}(v) = \{w \in V' : (v, w) \in E'\}$.
4: $V' \leftarrow V' \setminus \{v\}$ and from E' remove all the edges connected to v.
5: Define $E'' = \{(u, w) : u \in \mathbf{in}(v) \text{ and } w \in \mathbf{out}(v)\}$.
6: $E' \leftarrow E' \cup E''$.
7: **end while**
8: $E(S) \leftarrow E'$.
9: Return $H(S) = (S, E(S))$.

Remark 2. *Note that, for any $S \subseteq V$ and $u, v \in S$, we have $u \in \mathrm{view}(v)$ in $H(S)$ if and only if $u \in \mathrm{view}(v)$ in G. Hence, number of unrelated sets of $H(S)$ is at most the number of unrelated sets in G.*

We need a definition of negligible function to quantify the influence of vertices on a function.

Definition 16 (Negligible function)**.** *A function $\delta : \mathbb{N} \to [0, 1]$ is called a negligible function if for all constant $c > 0$ we have $\delta(n) \in o(n^{-c})$. We define $\mathsf{negl}(n)$ as the set of all negligible functions. Any function that is not in $\mathsf{negl}(n)$ we call it a non-negligible function.*

Now we need another definition of resilience of a directed acyclic graph G.

Definition 17 (Resilience of a graph)**.** *Given a directed acyclic graph $G = (V, E)$ and a subset $S \subseteq V$ we defined resilience of S as*

$$\mathrm{res}(S) := |S| - \max_{s \in S} |\operatorname{view}(s) \cap S| - 1$$

Further we define resilience of G as $\mathrm{res}(G) := \max_{S \subseteq V} \mathrm{res}(S)$.

Remark 3. *Note that for every $S \subseteq V$, number of head vertices in G^S is at least 1 and*

$$\max_{s \in S} |\operatorname{view}(s) \cap S| = \max_{u \in \mathrm{Head}(G^S)} |\operatorname{view}(u) \cap S| \ .$$

Hence, from definition of view it follows that $\mathrm{res}(S) \geq 0$ *for all $S \subseteq V$. Therefore,* $\mathrm{res}(G) \geq 0$ *for every directed acyclic graph G.*

Now we are ready to state our results on impossibility of extraction from SHEDAG source. At first we will show that in a given directed cyclic graph G and a function f if there is a vertex with non-negligible influence then it is impossible to extract from SHEDAG source with $\mathrm{res}(G)$ many corruptions. Prior to stating the theorem we will introduce a few notation.

let $G = (V, E)$ be a directed acyclic graph and $f : (\{0,1\}^n)^N \to \{0,1\}$ be a function. $V^f \subseteq V$ be the set of vertices v so that $\mathrm{Inf}_v^G(f) \notin \mathsf{negl}(n)$. If V^f is non-empty define the graph $G^f = H(V^f)$ to be the induced graph we have from the algorithm mentioned in Definition 15. For every vertex $u \in V^f$ we define $\operatorname{view}^f(u) = \operatorname{view}(u) \cap V^f$.

Theorem 7 (Impossibility of extraction with negligible error)**.** *For all large enough $n \in \mathbb{N}$ and fixed N the following holds: Let $G = (V, E)$ be a directed acyclic graph with $V = [N]$ and $f : (\{0,1\}^n)^N \to \{0,1\}$ be any function. Assume V^f is non-empty. Then there exists a non-negligible function $\varepsilon_0(n)$ and a (n, G, t)-SHEDAG source $\mathbf{X}$ with $t = \mathrm{res}(G) + 1$ so that $\Delta(f(\mathbf{X}) \ ; \ \mathcal{U}_1) \geq \varepsilon_0(n)$.*

Proof. By our assumption V^f is non-empty. Without loss of generality let us assume that number of $a \in (\{0,1\}^n)^N$ so that $f(a) = 0$ is at least 2^{nN-1}. Say $S = V \setminus V^f$ and $|S| = m$.

Let $u_0 \in \mathrm{Head}(G^f)$ so that for all $u \in \mathrm{Head}(G^f)$ we have $|\operatorname{view}^f(u_0)| \geq |\operatorname{view}^f(u)|$. Let R be the set of vertices v in V^f such that $v \notin \operatorname{view}^f(u_0) \cup \{u_0\}$. Now, we define the source $\mathbf{X}$ as: $\mathbf{X}_R = (\mathcal{U}_n)^{|R|}$, $\mathbf{X}_{\operatorname{view}^f(u_0)} = (\mathcal{U}_n)^{|\operatorname{view}^f(u_0)|}$, then sample x, x' uniformly and independently from $\{0,1\}^n$, when $\mathbf{X}_R = y_1$ and $\mathbf{X}_{\operatorname{view}^f(u_0)} = y_2$ define $\mathbf{X}_{u_0}$ as:

$$\mathbf{X}_{u_0} = \begin{cases} x & \text{if } E(x, y_1, y_2) \text{ occures} \ . \\ x' & \text{otherwise} \ . \end{cases}$$

where $E(x, y_1, y_2)$ is the event that: number of $z' \in (\{0,1\}^n)^m$ so that $f(X_{u_0} = x, X_{\operatorname{view}(u_0)} = y_2, X_R = y_1, X_{V \setminus V^f} = z') = 0$ is at least 2^{nm-1}. Finally set $\mathbf{X}_{V \setminus V^f} = (\mathcal{U}_n)^m$.

Notice that $\mathbf{X}$ is a (n, G, t)-SHEDAG source with $t = \mathrm{res}(G) + 1$. As, $|\operatorname{view}^f(u_0)|$ is at least $|\operatorname{view}^f(u)|$ for every $u \in G^f$ and every vertex of G^f is in view of some head vertex, we have

$$|\operatorname{view}^f(u_0)| = \max_{u \in V^f} |\operatorname{view}^f(u)| \ .$$

Hence, $\mathrm{res}(G^f) = |V^f| - |\mathrm{view}^f(u_0)| - 1$ and number of corrupted blocks is $|V^f| - |\mathrm{view}^f(u_0)|$[7] which is at most $\mathrm{res}(G) + 1$.

Sample z_0 uniformly from $(\{0,1\}^n)^m$ and set $f_{z_0} = f(X_{V^f}, X_S = z_0)$. Note that,

$$\Pr[f(\mathbf{X}) = 0] \geq \Pr[f_{z_0}(\mathbf{X}_{V^f}) = 0 \mid z_0 \in \text{Good}] \ \Pr[z_0 \in \text{Good}] \tag{2}$$

where, Good is the set of all $z \in (\{0,1\}^n)^m$ so that,

- $\mathrm{Inf}^G_{u_0}(f_z) \notin \mathsf{negl}(n)$.
- $\Pr_y[f_z(y) = 0] \geq 1/2 - \delta'(n)$ for some $\delta'(n) \in \mathsf{negl}(n)$.

Claim 1. $\Pr[z_0 \in \textit{Good}\,] \geq 1 - \tilde{\delta}(n)$ *for some* $\tilde{\delta}(n) \in \mathsf{negl}(n)$.

At first we continue proving Theorem 7 assuming the preceding claim and prove the claim after that. Let $E_i(x, y_1, y_2)$ be the event that $f_{z_0}(X_{u_0} = x, X_{\mathrm{view}(u_0)} = y_2, X_R = y_1) = i$ and $E_i(x', y_1, y_2)$ be the event that $f_{z_0}(X_{u_0} = x', X_{\mathrm{view}(u_0)} = y_2, X_R = y_1) = i$ for $i = 0, 1$. Notice that, when $z_0 \in$ Good , u_0 has non-negligible influence on f_{z_0}. Hence, from Definition 14 we have,

$$\begin{aligned}&\Pr_{x,x',y_1,y_2}[E_0(x, y_1, y_2) \wedge E_1(x', y_1, y_2) \mid z_0 \in \text{Good}] \\ &+ \Pr_{x,x',y_1,y_2}[E_0(x', y_1, y_2) \wedge E_1(x, y_1, y_2) \mid z_0 \in \text{Good}] = \varepsilon(n)\end{aligned}$$

where $\varepsilon(n) \notin \mathsf{negl}(n)$. As both x, x' are uniformly chosen, by symmetry, we have that $\Pr_{x,x',y_1,y_2}[E_0(x, y_1, y_2) \wedge E_1(x', y_1, y_2) \mid z_0 \in \text{Good}] = \varepsilon(n)/2$. Now, notice that

$$\begin{aligned}&\Pr[f_{z_0}(\mathbf{X}_{V^f}) = 0 \mid z_0 \in \text{Good}] \\ &= \Pr_{x,y_1,y_2}[E_0(x, y_1, y_2) \mid z_0 \in \text{Good}] + \Pr_{x,x',y_1,y_2}[E_0(x', y_1, y_2) \wedge E_1(x, y_1, y_2) \mid z_0 \in \text{Good}]\ .\end{aligned}$$

As, $\Pr_{x,y_1,y_2}[E_0(x, y_1, y_2) \mid z_0 \in \text{Good}] \geq 1/2 - \delta'(n)$ we have,

$$\Pr[f(\mathbf{X}_{V^f}) = 0 \mid z_0 \in \ \text{Good}\ ] \geq (1/2 + \varepsilon(n)/2 - \delta'(n))\ .$$

Finally from Claim 1 and Equation (2) we can conclude that,

$$\Pr[f(\mathbf{X}) = 0] \geq (1/2 + \varepsilon(n)/2 - \delta'(n))(1 - \tilde{\delta}(n))\ .$$

Since ε is non-negligible function and $\delta'(n) \in \mathsf{negl}(n)$ we have $\varepsilon(n)/2 - \delta'(n)$ is non-negligible. Also, $\tilde{\delta}(n)$ is a negligible function. Hence $\Pr[f(\mathbf{X}) = 0] \geq 1/2 + \varepsilon_0(n)$ for some $\varepsilon_0(n) \notin \mathsf{negl}(n)$ and from the definition of statistical distance our proof follows. □

Proof of Claim 1. For a vertex $u \in V^f$ define $S^u = V^f \setminus \{u\}$. As all the vertices in $V \setminus V^f$ has negligible influence, by hybrid argument we have: there is a function $\delta(n) \in \mathsf{negl}(n)$ so that for large enough n,

$$\Pr_{z_0,x,y,z}\left[f_{z_0}(X_u = x, X_{S^u} = y) \neq f(X_u = x, X_{S^u} = y, X_S = z)\right] \leq m \cdot \delta(n) \tag{3}$$

[7] As the adversary is fixing $\mathbf{X}_R$ by uniformly sampled y_1, it may seem like we are only corrupting $\mathbf{X}_{u_0}$. But notice that the adversary needs to see y_1 to sample $\mathbf{X}_{u_0}$ which it cannot do unless it corrupts $\mathbf{X}_R$.

$$\implies \Pr_{z_0,x,y,z}[f_{z_0}(X_u = x, X_{S^u} = y) = f(X_u = x, X_{S^u} = y, X_S = z)] \geq 1 - m\delta(n) \tag{4}$$

Consider the uniform random variable Z over $(\{0,1\}^n)^m$, defined as follows,

$$Z(z) = \Pr_{x,y,z}[f_{z_0}(X_u = x, X_{S^u} = y) = f(X_u = x, X_{S^u} = y, X_S = z)] \,.$$

Clearly from Equation (3), $\mathbb{E}[Z] \geq 1 - m\delta(n)$. By Lemma 1 we have, $\Pr_{z_0}[Z \leq 1 - \sqrt{m\delta(n)}] \leq \sqrt{m\delta(n)}$. We can rewrite this as,

$$\Pr_{z_0}\left(\Pr_{x,y,z}[f_{z_0}(x,y) = f(x,y,z)] > 1 - \sqrt{m\delta(n)}\right) \geq 1 - \sqrt{m\delta(n)} \,. \tag{5}$$

Define the set Great $\subseteq (\{0,1\}^n)^m$ as, for every $z' \in (\{0,1\}^n)^m$ we have $z' \in$ Great if $\Pr_{x,y,z}[f_{z'}(x,y) = f(x,y,z)] > 1 - \sqrt{m\delta(n)}$. Also, recall that $u_0 \in V^f$ hence $\mathrm{Inf}^G_{u_0}(f) = \varepsilon_{u_0}(n)$ where $\varepsilon_{u_0}(n) \notin \mathsf{negl}(n)$. Now by union bound we have,

$$\Pr_{x_0,x_1,y,z}\left[\begin{array}{c} f_{z_0}(X_{u_0} = x_0, X_{S^{u_0}} = y) = f(X_{u_0} = x_0, X_{S^{u_0}} = y, X_S = z) \\ \wedge \quad f_{z_0}(X_{u_0} = x_1, X_{S^{u_0}} = y) = f(X_{u_0} = x_1, X_{S^{u_0}} = y, X_S = z) \end{array} \middle| z_0 \in \text{Great}\right]$$

is at least $1 - 2\sqrt{m\delta(n)}$. From the definition of influence (see Definition 14) and again by union bound we have,

$$\Pr_{x_0,x_1,y}[f_{z_0}(X_{u_0} = x_0, X_{S^{u_0}} = y) \neq f_{z_0}(X_{u_0} = x_1, X_{S^{u_0}} = y) \mid z_0 \in \text{Great}] \geq \varepsilon_{u_0}(n) - 2\sqrt{m\delta(n)} \,. \tag{6}$$

Since $\Pr[z_0 \in \text{Great}] \geq 1 - 2\sqrt{m\delta(n)}$, combining with 6 we get: with probability at least $1 - 2\sqrt{m\delta(n)}$ over the choice of z_0 it holds that $\mathrm{Inf}^G_{u_0}(f_{z_0}) \geq \varepsilon_{u_0}(n) - 2\sqrt{m\delta(n)}$.

Next note that from our assumption, $\Pr_{x,y,z}[f(X_u = x, X_{S^u} = y, X_S = z)]$ is at least $1/2$. Therefore from 5 the following holds: With probability at least $1 - \sqrt{m\delta(n)}$ over the choice of z_0, the following holds:

$$\Pr_y[f_{z_0}(y) = 0] \geq \tfrac{1}{2} - \tfrac{1}{2}\sqrt{m\delta(n)} \,.$$

As, $m \leq N$ and N is fixed, we have $m\delta(n) \in \mathsf{negl}(n)$ which further implies $c\sqrt{m\delta(n)} \in \mathsf{negl}(n)$ for any constant c. Since $\varepsilon_{u_0}(n)$ is non-negligible, so is $\varepsilon_{u_0}(n) - 2\sqrt{m\delta(n)}$. Finally by union bound we have $\Pr[z_0 \in \text{Good}] \geq 1 - 4\sqrt{m\delta(n)}$. Setting $\tilde{\delta}(n) = 4\sqrt{m\delta(n)}$ yields the proof. □

Next we will proceed to show that if in the directed acyclic graph $G = (V, E)$ and function f we have, $\mathrm{Inf}^G_u(f) \in \mathsf{negl}(n)$, for all $u \in V$, then with *high* probability over the uniformly random inputs the function f is constant.

Theorem 8. *For all large enough $n \in \mathbb{N}$ and fixed $N \in \mathbb{N}$ the following holds: Given direct acyclic graph $G = (V, E)$ with $V = [N]$ and $f : (\{0,1\}^n)^N \to \{0,1\}$ be any function. Further assume that for all $u \in V$, $\mathrm{Inf}^G_u(f) \in \mathsf{negl}(n)$. Then there exists $a \in \{0,1\}$ so that $\Pr_{x \leftarrow (\{0,1\}^n)^N}[f(x) = a] \geq 1 - \delta_0(n)$ for some $\delta_0(n) \in \mathsf{negl}(n)$.*

Proof. Note that, by our assumption and Definition 14, for all $u \in V$ the following holds: There exists a function $\delta(n) \in \mathsf{negl}(n)$ so that,

$$\Pr_{x_0,x_1,y}[f(X_u = x_0, X_{S^u} = y) \neq f(X_u = x_1, X_{S^u} = y)] \leq \delta(n)$$

where $S^u = V \setminus \{u\}$. Now, note that, by hybrid argument we have,

$$\Pr_{\substack{x \leftarrow (\{0,1\}^n)^N \\ x' \leftarrow (\{0,1\}^n)^N}}[f(x) \neq f(x')] \leq N\delta(n) \tag{7}$$

which implies that $\Pr_{x,x'}[f(x) = f(x')] \geq 1 - N\delta(n)$. Say, $\Pr[f(\mathcal{U}_{nN}) = 1] = p$ and $\Pr[f(\mathcal{U}_{nN}) = 0] = 1 - p$. Without loss of generality we can assume $p \geq 1/2$. As, collision probability of $f(\mathcal{U}_{nN})$ is more than $1 - N\delta(n)$, we have $p^2 + (1-p)^2 \geq 1 - N\delta(n)$. That implies,

$$2p - 2p^2 \leq N\delta(n) \implies 2p(1-p) \leq N\delta(n) \implies p \geq 1 - N\delta(n).$$

The last implication holds because of our assumption that $p \geq 1/2$. As, N is fixed and $\delta(n) \in \mathsf{negl}(n)$ we have $N\delta(n) \in \mathsf{negl}(n)$. Setting $\delta_0(n) = N\delta(n)$ the proof follows. □

Combining Theorem 7 and Theorem 8 we get the following corollary.

Corollary 1. *For all large enough $n \in \mathbb{N}$ and fixed $N \in \mathbb{N}$ the following holds: Let $G = (V, E)$ be a directed acyclic graph with $V = [N]$. Then for every function $f : (\{0,1\}^n)^N \to \{0,1\}$ there exists a (n, G, t)-SHEDAG source with $t = \mathrm{res}(G) + 1$ so that $\Delta(f(\mathbf{X}) \,;\, \mathcal{U}_1) \geq n^{-c}$ for some constant c.*

4.3.2 Explicit extractor

We will start by recalling the definition of resilience of a directed acyclic graph.

Definition 18 (Resilience of a graph)**.** *Given a directed acyclic graph $G = (V, E)$ and a subset $S \subseteq V$ we defined resilience of S as*

$$\mathrm{res}(S) := |S| - \max_{s \in S} |\,\mathrm{view}(s) \cap S| - 1$$

Further we define resilience of G as $\mathrm{res}(G) := \max_{S \subseteq V} \mathrm{res}(S)$.

In the previous section we proved that in a given directed acyclic graph $G = (V = [N], E)$ if we allow number of corruptions is strictly more than the resilience of the graph then it is impossible to extract from SHEDAG sources with negligible error. In this section we will show that in fact it is tight in the sense that if number of corrupted vertices is at most $\mathrm{res}(G)$ then there exists an explicit function that can extract from SHEDAG sources with negligible error.

Before stating the construction of the extractor we need a few definitions which will be useful.

Definition 19 (Headless pair of vertices)**.** *In a given directed acyclic graph $G = (V, E)$ with $V = [N]$ a pair $\{x, y\} \subseteq V$ is called headless pair of vertices if x and y are unrelated (see Definition 4) and there does not exist any $h \in \text{Head}(G)$ so that $x, y \in \text{view}(h)$.*

Next we need to define a order between headless vertices of G.

Definition 20. *Given a directed cyclic graph $G = (V = [N], E)$, and $\text{Head}(G)$ is the set of head vertices of G. Take any arbitrary order $\preceq$ of the head vertices. We extend this order for any pair of headless vertices in the following way:*
For any two headless pair of vertices $\{u, v\} \subseteq V$ define,

$$S_u = \begin{cases} \text{parents}(u) \cap \text{Head}(G) & \text{if } u \notin \text{Head}(G) \\ \{u\} & \text{otherwise .} \end{cases}$$

Similarly we define S_v. Say u' is the smallest head vertex in S_u and v' is the smallest vertex in S_v in the order $\preceq$. Then, we say $u \preceq v$ if $u' \preceq v'$.

Remark 4. *If $\{u, v\} \subseteq V$ be a pair of headless vertices then, $\text{parents}(u) \cap \text{parents}(v)$ is empty. Otherwise since every vertex is in view of some head vertex, it contradicts the fact that u, v are headless. Moreover, for any headless pair $\{w_1, w_2\}$ so that $w_1 \in \text{parents}(u)$ and $w_2 \in \text{parents}(v)$, we have $w_1 \preceq w_2$.*

Next we will define set of ordered pair of headless vertices based on the order defined before.

Definition 21 (Set of ordered pair of headless vertices)**.** *Given a directed acyclic graph $G = (V, E)$ with $V = [N]$. $\preceq$ is the order defined between headless vertex pair. The set of ordered pair of independent vertices of G is defined as*

$$I := \left\{(u, v) : \{u, v\} \text{ is headless pair of vertices of } G \text{ and } u \preceq v\right\} .$$

We now proceed to define the construction of our extractor. Our construction is built upon two source non-malleable extractors (see Definition 11). At first we will prove a general statement and after that plugging the current state-of-the-art constructions of non-malleable extractors we will give explicit extractors for SHEDAG sources in various parameter regimes.

Theorem 9 (Explicit extractor for SHEDAG source with negligible error)**.** *For all $n, m \in \mathbb{N}$, fixed $N \in \mathbb{N}$, $k \leq n$ and $\hat{\varepsilon} > 0$ the following holds: Let $G = (V, E)$ be a directed acyclic graph with $V = [N]$. Suppose there exists an explicit (N^2, k, ε)-two source non-malleable extractor $\mathsf{2nmExt} : \{0,1\}^{n+\log N} \times \{0,1\}^{n+\log N} \to \{0,1\}^m$. Then we have an explicit $(n, k', G, t, \varepsilon')$-SHEDAG extractor with $t = \text{res}(G)$, $k' \geq k + m + \log(1/\hat{\varepsilon})$, $\varepsilon' \leq \varepsilon + 2\hat{\varepsilon}$.*

Proof. Without loss of generality we can assume that $\text{res}(G) \geq 1$ because otherwise number of corrupted blocks is 0.

Since $t = \text{res}(G)$, by definition of resilience, there exists $S \subseteq V$ so that $t = \text{res}(S)$. In the next section (Section 4.3.3) we will show that we can find this subset S in $\text{poly}(N)$ time. Let, $G^S = H(S)$ be the graph that we can find by the algorithm mentioned in Definition 15.

Observe that, for every $s \in S$ we have $|\operatorname{view}(s) \cap S|$ is at most $\max_{u \in \operatorname{Head}(G^S)} |\operatorname{view}(u) \cap S|$, hence formally,

$$\operatorname{res}(S) = |S| - \max_{u \in \operatorname{Head}(G^S)} |\operatorname{view}(u) \cap S| - 1 \ .$$

Since $\operatorname{res}(G^S) \geq 1$, we have $|\operatorname{Head}(G^S)| \geq 2$. Moreover, as $t = \operatorname{res}(S)$, from the above observation and Remark 4 it follows that, there exist a pair of headless vertices $\{i_0, j_0\} \subseteq S$ so that $\mathbf{X}_{i_0}, \mathbf{X}_{j_0}$ are independent (n, k') sources.

Let I^S be set of ordered pair of headless vertices of G^S (see Definition 21). Define

$$\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) := \bigoplus_{(i,j) \in I^S} \mathsf{2nmExt}(\mathbf{X}_i \| \mathsf{bit}(i), \mathbf{X}_j \| \mathsf{bit}(j)) \ , \tag{8}$$

where $\mathsf{bit}(i), \mathsf{bit}(j)$ are the binary representations of i, j respectively. Notice that by Remark 4, for any $i', j' \in S$ so that $i' \in \operatorname{parents}(i_0)$ and $j' \in \operatorname{parents}(j_0)$, we have $\{i', j'\}$ is also a headless vertex pair with $i' \preceq j'$. Define the partition of $I^S = \sqcup_{\alpha=1}^{4} A_\alpha$ where,

- $A_1 = \big\{(i', j') \in I^S : i' \in \operatorname{parents}(i_0) \cup \{i_0\} \text{ and } j' \in \operatorname{parents}(j_0) \cup \{j_0\}\big\}$.
- $A_2 = \big\{(i', j') \in I^S : i' \in \operatorname{parents}(i_0) \cup \{i_0\} \text{ but } j' \notin \operatorname{parents}(j_0) \cup \{j_0\}\big\}$.
- $A_3 = \big\{(i', j') \in I^S : i' \notin \operatorname{parents}(i_0) \cup \{i_0\} \text{ but } j' \in \operatorname{parents}(j_0) \cup \{j_0\}\big\}$.
- $A_4 = \big\{(i', j') \in I^S : i' \notin \operatorname{parents}(i_0) \cup \{i_0\} \text{ and } j' \notin \operatorname{parents}(j_0) \cup \{j_0\}\big\}$.

Define $\mathbf{Z}_\alpha = \bigoplus_{(i,j) \in A_\alpha} \mathsf{2nmExt}(\mathbf{X}_i \| \mathsf{bit}(i), \mathbf{X}_j \| \mathsf{bit}(j))$ for $\alpha = 1, 2, 3, 4$ and we rewrite 8 as,

$$\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) = \mathbf{Z}_1 \oplus \mathbf{Z}_2 \oplus \mathbf{Z}_3 \oplus \mathbf{Z}_4 \ .$$

At first we define the set $A' = S \setminus (\operatorname{parents}(i_0) \cup \operatorname{parents}(j_0) \cup \{i_0, j_0\})$. Now for every $v \in A'$ and for $x_v \sim \mathbf{X}_v$ we will fix the random variable $\mathbf{X}_v$ to x_v. Note that this will fix $\mathbf{Z}_4$ and it will not cause any entropy loss to $\mathbf{X}_u$ for $u = i_0, j_0$ and $u \in \operatorname{parents}(i_0) \cup \operatorname{parents}(j_0)$. Moreover, for every $(i, j) \in A_1$, the random variables $\mathbf{X}_i, \mathbf{X}_j$ will remain independent.

Next, for $z_2 \sim \mathbf{Z}_2$ we next fix $\mathbf{Z}_2$ to z_2. Observe that $\mathbf{X}_{j_0}$ and $\mathbf{Z}_2$ are independent. Hence this fixing will not cause entropy loss to $\mathbf{X}_{j_0}$. And by Lemma 2 we have $\mathbf{H}_\infty(\mathbf{X}_{i_0} | \mathbf{Z}_2 = z_2) \geq k' - m - \log(1/\hat{\varepsilon})$ with probability at least $1 - \hat{\varepsilon}$ over the choice of z_2.

Similarly we fix $\mathbf{Z}_3$ by some $z_3 \sim \mathbf{Z}_3$, by Lemma 2, $\mathbf{H}_\infty(\mathbf{X}_{j_0} | \mathbf{Z}_3 = z_3) \geq k' - m - \log(1/\hat{\varepsilon})$ with probability at least $1 - \hat{\varepsilon}$ over the choice of z_3. Since $\mathbf{Z}_3$ and $\mathbf{X}_{i_0}$ are independent this fixing will not cause entropy loss to $\mathbf{X}_{i_0}$.

After the fixings of $\mathbf{X}_v$ for all $v \in A'$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$, we have $\mathbf{Z}_2 \oplus \mathbf{Z}_3 \oplus \mathbf{Z}_4$ is fixed and

$$\mathbf{Z}_1 = \mathsf{2nmExt}(\mathbf{Y}_{i_0} \| \mathsf{bit}(i_0), \mathbf{Y}_{j_0} \| \mathsf{bit}(j_0)) \oplus \left(\bigoplus_{\substack{(i,j) \in A_1 \\ (i,j) \neq (i_0, j_0)}} \mathsf{2nmExt}(\mathbf{Y}_i \| \mathsf{bit}(i), \mathbf{Y}_j \| \mathsf{bit}(j)) \right)$$

where $\mathbf{Y}_i = \mathbf{X}_i | (\mathbf{Z}_2 = z_2, \{\mathbf{X}_v = x_v\}_{v \in A'})$ and $\mathbf{Y}_j = \mathbf{X}_j | (\mathbf{Z}_3 = z_3, \{\mathbf{X}_v = x_v\}_{v \in A'})$, for $(i, j) \in A_1$. Moreover, for every $(i, j) \in A_1$ we have $\mathbf{Y}_i, \mathbf{Y}_j$ will still remain independent and there does not exist any u so that $\mathbf{X}_u$ depends on both $\mathbf{X}_i, \mathbf{X}_j$ since $\{i, j\}$ is headless pair. Note that

for $(i,j) \neq (i_0,j_0)$, there exist randomized tampering functions $g_1^{(i,j)}, g_2^{(i,j)} : \{0,1\}^n \to \{0,1\}^n$ so that all of them are sharing same randomness independent of $\mathbf{Y}_i, \mathbf{Y}_j$ for all $(i,j) \in A_1$ and we can write

$$\mathbf{Y}_i \| \mathsf{bit}(i) = g_1^{(i,j)}(\mathbf{Y}_{i_0} \| \mathsf{bit}(i_0)) \text{ and } \mathbf{Y}_j \| \mathsf{bit}(j) = g_2^{(i,j)}(\mathbf{Y}_{j_0} \| \mathsf{bit}(j_0)) \ .$$

Moreover for every fixing of the randomness of the tampering functions, for all $(i,j) \neq (i_0,j_0)$ we have at least one of $g_1^{(i,j)}$ or $g_2^{(i,j)}$ has no fixed point. As, $\mathsf{2nmExt}$ is (ℓ, k, ε) two source non-malleable extractor with $\ell = N^2$ and $\mathbf{H}_\infty(\mathbf{Y}_{i_0} \| \mathsf{bit}(i_0)), \mathbf{H}_\infty(\mathbf{Y}_{j_0} \| \mathsf{bit}(j_0)) \geq k$, from Lemma 4 we have,

$$\Delta\big(\mathsf{2nmExt}(\mathbf{Y}_{i_0} \| \mathsf{bit}(i_0), \mathbf{Y}_{j_0} \| \mathsf{bit}(j_0)) \ ; \ \mathcal{U}_m | \{\mathsf{2nmExt}(\mathbf{Y}_i \| \mathsf{bit}(i), \mathbf{Y}_j \| \mathsf{bit}(j))\}_{A_1 \ni (i,j) \neq (i_0,j_0)}\big) \leq \varepsilon \ .$$

From here by Lemma 3, we have

$$\Delta\left(\bigoplus_{(i,j)\in A_1} \mathsf{2nmExt}(\mathbf{Y}_i \| \mathsf{bit}(i), \mathbf{Y}_j \| \mathsf{bit}(j)) \ ; \ \mathcal{U}_m\right) \leq \varepsilon \ .$$

Therefore, $\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) \approx_{\varepsilon'} \mathcal{U}_m$ with $\varepsilon' \leq \varepsilon + 2\hat{\varepsilon}$. □

The above theorem combining with two source non-malleable extractor construction of [CGL20] leads to the following corollary.

Corollary 2. *There exists a constant $\gamma \in (0,1)$ so that for all $n \in \mathbb{N}$ large enough and fixed $N \in \mathbb{N}$ the following holds: Let, $G = (V,E)$ be a directed acyclic graph with $V = [N]$. Then for $t = \mathrm{res}(G)$ there exists an explicit $\left(n, n - n^{\gamma/2}, G, t, 2^{-n^{\gamma/2}}\right)$-SHEDAG extractor with output length $n^{\gamma/2}$.*

Proof. Let $\gamma > 0$ be the constant from Proposition 1 and $\mathsf{CGL} : \{0,1\}^{\tilde{n}} \times \{0,1\}^{\tilde{n}} \to \{0,1\}^{\tilde{m}}$ is $(\ell, \tilde{k}, \tilde{\varepsilon})$-two source non-malleable extractor where $\tilde{n} = n + \log N$, $\ell \leq \tilde{n} - (\tilde{n})^\gamma$, $\tilde{k} \geq \tilde{n} - (\tilde{n})^\gamma, \tilde{m} = (\tilde{n})^\gamma$ and $\tilde{\varepsilon} \leq 2^{-(\tilde{n})^\gamma}$.

Note that for any constant c and large enough n we have $(n+c)^\gamma \geq 3n^\gamma/4$ as $(n+c)^\gamma - n^\gamma \in o(n^\gamma)$. So, there exists $n_0 \in \mathbb{N}$ so that for $n \geq n_0$,

$$(n + \log N) - (n + \log N)^\gamma \leq n^\gamma/4 - \log N \leq n - n^\gamma/2 \ .$$

Finally in Theorem 9 substituting $m = n^{\gamma/2}$ and $\hat{\varepsilon} = 2^{-n^{\gamma/2}}$ we get $(n + \log N) - (n + \log N)^\gamma + 2 \cdot n^{\gamma/2} \leq n - n^{\gamma/2}$ for large n and from here the proof follows. □

We can derive another corollary from the construction of [ACO23] and Theorem 9.

Corollary 3. *For all large enough $n \in \mathbb{N}$ and fixed $N \in \mathbb{N}$ the following holds: Let $G = (V,E)$ be a directed acyclic graph with $V = [N]$. Then there exists $c < 1/2$ so that for $t = \mathrm{res}(G)$ there exists an explicit $\left(n, \left(1 - \frac{1}{5(2N^2+3)}\right)n, G, t, 2^{-\Omega(n^c)}\right)$-SHEDAG extractor with output length $\Omega(n)$.*

Proof. Let $\mathsf{ACO} : \{0,1\}^{\tilde{n}} \times \{0,1\}^{\tilde{n}} \to \{0,1\}^{\tilde{m}}$ be the $(\ell, \tilde{k}, \tilde{\varepsilon})$-two source non-malleable extractor from Proposition 2 with $\tilde{n} = n + \log N$, $\tilde{k} \geq \left(1 - \frac{1}{2N^2+3}\right)\tilde{n}$, $\tilde{m} = \Omega(\tilde{k})$ and $\tilde{\varepsilon} \leq 2^{-\tilde{k}^c}$ where $c < 1/2$.

Note that as N is fixed, $\left(1 - \frac{1}{2(2N^2+3)}\right)n \geq \left(1 - \frac{1}{2N^2+3}\right)\tilde{n}$. Substituting $\hat{\varepsilon} = 2^{-n^c}$ and $m = \frac{n}{4(2N^2+3)}$ in Theorem 9 we have, $\left(1 - \frac{1}{5(2N^2+3)}\right)n \geq \left(1 - \frac{1}{2N^2+3}\right)\tilde{n} + n^c + \frac{n}{4(2N^2+3)}$ for large enough n. From here our proof follows. □

4.3.3 Algorithm for locating subset with highest resilience

Definition 22 (Resilience)**.** *Given a graph $G = (V, E)$, we say that a subset of vertices $S \subseteq V$ has resilience r if*

$$|S| - \max_{s \in S}(|\operatorname{view}(s) \cap S|) - 1 = r$$

As we discussed in the previous section, if S has the *resilience* value r, our extractor can take S as its influence set and such extractor would produce uniform output under r corruptions. Therefore, the natural goal is to find the subset that maximizes *resilience*.

First, note that it suffices to consider the maximum view of only the head vertices in S.

$$|S| - \max_{s \in S}(|\operatorname{view}(s) \cap S|) - 1 = |S| - \max_{s \in \mathrm{Head}(G^S)}(|\operatorname{view}(s) \cap S|) - 1$$

where $G^S = H(S)$ is the graph induced by the set S defined in Definition 15. Since for all $v \in S$, if there exists $u \in \mathrm{parents}(v) \cap S$, then $\operatorname{view}(v) \cap S \subset \operatorname{view}(u) \cap S$.

The next lemma states that when locating subset with maximum resilience, it suffices to look for sets that do not truncate the view of the nodes, this means that if $v \in S$ then $\operatorname{view}(v) \subseteq S$.

Lemma 6. *For any $S \subseteq V$, denote the set of heads in S as $\mathrm{Head}(S)$. Consider S' defined as follows:*

$$S' = S \cup \{u \in V : \exists v \in \mathrm{Head}(S),\ s.t.\ u \in \operatorname{view}(v)\}$$

Then the resilience of S' is at least the resilience of S.

Proof. By the construction, the sets of heads of S and S' are the same. Let us consider any vertex h in set S', then:

$$|\operatorname{view}(h) \cap S'| \leq |\operatorname{view}(h) \cap S| + |S' \setminus S|.$$

Therefore, we can get the following bound:

$$|S'| - |\operatorname{view}(h) \cap S'| \ \geq\ |S'| - \left(|\operatorname{view}(h) \cap S| + |S' \setminus S|\right) \ =\ |S| - |\operatorname{view}(h) \cap S|.$$

It follows that

$$|S'| - \max_{h \in S'} |\operatorname{view}(h) \cap S'| - 1 \ \geq\ |S| - \max_{h \in S} |\operatorname{view}(h) \cap S| - 1,$$

Thus, the resilience of S' is at least as large as the resilience of S. □

Following the above lemma, we can always assume that our resilient set S is determined by the set of head vertices S_H, simply by taking $S = S_H \cup \{u \in V : \exists h \in S_H, u \in \text{view}(h)\}$. This means we can optimize resilience by iteratively removing nodes that are currently heads, without changing non-head vertices.

Definition 23. *(Intact Set) When searching for the most resilient set, we will be looking at sets with the following property $v \in S \Rightarrow \text{view}(v) \subseteq S$. We shall call such sets* intact.

The next lemma indicates we can remove head vertices in a greedy way. Recall that by the Lemma 6 it suffices to consider sets that are *intact*.

Lemma 7. *Suppose there exists set H that maximizes resilience, and $H \subseteq S \subseteq V$, and that H, S are both intact. Consider the set of head vertices in S the have maximum view: $S_0 := \{h \in S : |\text{view}(h) \cap S| = \max_{p \in S}(|\text{view}(p) \cap S|)\}$. Then either S maximizes resilience, or $H \cap S_0 = \emptyset$.*

Proof. If S maximizes resilience, lemma is proven. Otherwise, assume $\exists v \in H \cap S_0$. Then by the Definition 23, we have $\text{view}(v) = \text{view}(v) \cap H = \text{view}(v) \cap S$. Therefore, by the definition of set S_0 we have

$$|\text{view}(v) \cap H| = |\text{view}(v) \cap S| = \max_{h \in S}(|\text{view}(h) \cap S|),$$

but $|H| < |S|$, and $|\text{view}(v) \cap H| = \max_{h \in H}(|\text{view}(h) \cap H|)$ thus:

$$|H| - \max_{h \in H}(|\text{view}(h) \cap H|) - 1 = |H| - (|\text{view}(v) \cap H|) - 1 <$$
$$< |S| - |\text{view}(v) \cap H| - 1 = |S| - \max_{h \in S}(|\text{view}(h) \cap S|) - 1.$$

This is a contradiction with H optimizing resilience. Therefore, for all *intact* H that maximize resilience and are contained in S, we must have $H \cap S_0 = \emptyset$. Again, we stress here that by Lemma 6 it suffices to consider *intact* sets that maximize resilience. □

Following this lemma, we can design the algorithm as follows. First, preprocess the graph to compute and store the view size of each vertex. Then, iteratively remove the head with the largest view, together with its incident edges. At each step, update and record the maximum resilience attained. After this elimination phase terminates, repeat the process: continue removing heads until the current resilience equals the previously recorded optimum. Finally, collect the heads present at this stage, and perform a traversal (e.g., BFS or DFS) from these heads to recover the entire optimal subset.

We will present the pseudocode for finding maximum resilience of subsets. The rest of algorithm is similar to the same routine.

Algorithm 2 Greedy head removal

Input: DAG $G = (V, E)$
Output: Set with maximum resilience : *best*
Compute and store $|view(h)|$ for all $h \in V$
$s \leftarrow |V|$
Create array A with length $|V|$ (0-index), elements initialized to empty linked list
For all heads h of G, append h to $A[|\operatorname{view}(h)|]$
$best \leftarrow 0$
Maintain a graph structure G
$i \leftarrow |V| - 1$
while $i \geq 0$ **do**
  if $A[i]$ is empty **then**
    $i \leftarrow i - 1$
    continue
  end if
  Remove head of $A[i]$, denote by v
  $best \leftarrow \max\{best, s - |\operatorname{view}(v)|\}$
  for u children of v in G **do**
    if u has no parent except v **then**
      Append u to $A[|view(u)|]$
    end if
  end for
  Remove v and adjacent edges from G
  $s \leftarrow s - 1$
end while
return *best*

For the first step, we can perform a toposort and then compute and store the size of view of vertices in order of toposort. This takes time linear in $|V| + |E|$. After this, we use an array to store the current head vertices in the same order as their size of view, along with a graph G. At each step, we remove the head with largest view, compute new resilience, and then insert new heads into the array. Note the new heads inserted will always have a smaller view, thus inserted to an earlier index in the array.

At the second iteration, we use same routine until we find the state of array that results in largest resilience. Then we record all head vertices and use Lemma 6 to retrieve whole optimal subset.

Correctness:

Loop invariant: Before *best* is updated to actual optimal resilience, at the start of each iteration of outer while-loop, either the remaining vertices in G form an optimal resilient subset, or there exists an optimal intact subset of vertices in G.

Initialization: observe that the optimal $H \subseteq V$ must exist, and that we can assume H is intact by Lemma 6.

Maintenance: Suppose at the start of an iteration, $G = (V', E')$ and loop invariant is satisfied. If V' is an optimal resilience subset itself, this implies *best* will be updated to optimal

in current iteration. Otherwise, suppose the intact subset guaranteed by loop invariant is $H \subset V'$. Note since we remove only heads, after removal step, the remaining subset is still intact. By Lemma 7, we know that $H \cap \{h \in V' : |\operatorname{view}(h) \cap V'| = \max_{p \in V'}(|\operatorname{view}(p) \cap V'|) = \emptyset$. Since the removed head has largest possible view in subset, it is not in H. Therefore, after removal, H is still a subset of the remaining vertices in G

Termination: Since the algorithm ends after removing all vertices, upon termination, *best* must be correctly updated, otherwise this means that the empty set itself is an optimal resilience subset.

Afterwards, we re-run the algorithm to find the heads of the intact subset associated with maximal resilience. Since intact subsets are determined uniquely by head vertices, we are able to retrieve the entire optimal subset.

Runtime: Preprocessing costs $O(|V| + |E|)$ with toposort. Each vertex is added and removed from array A at most twice, each costing $O(1)$ time. Keeping and modifying graph structure in algorithm takes $O(|V| + |E|)$ time. Thus total runtime is $O(|V| + |E|)$, which is asymptotically same as simply reading through the graph G.

References

[ACO23] Divesh Aggarwal, Eldon Chung, and Maciej Obremski. Extractors: Low entropy requirements colliding with non-malleability. In *Advances in Cryptology – CRYPTO 2023: 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20–24, 2023, Proceedings, Part II*, page 580–610, Berlin, Heidelberg, 2023. Springer-Verlag. doi:10.1007/978-3-031-38545-2_19.

[AOR+20] Divesh Aggarwal, Maciej Obremski, João Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to extract useful randomness from unreliable sources. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 343–372, Cham, 2020. Springer International Publishing.

[AOR+22] Divesh Aggarwal, Maciej Obremski, João Ribeiro, Mark Simkin, and Luisa Siniscalchi. Privacy amplification with tamperable memory via non-malleable two-source extractors. *IEEE Transactions on Information Theory*, 68(8):5475–5495, 2022. doi:10.1109/TIT.2022.3167404.

[BGM22] Marshall Ball, Oded Goldreich, and Tal Malkin. Randomness extraction from somewhat dependent sources. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pages 12:1–12:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.12, doi:10.4230/LIPIcs.ITCS.2022.12.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988. arXiv:https://doi.org/10.1137/0217015, doi:10.1137/0217015.

[CG14] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In Yehuda Lindell, editor, *Theory of Cryptography*, pages 440–464, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 1184–1197, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384339.

[CGL20] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Nonmalleable extractors and codes, with their many tampered extensions. *SIAM Journal on Computing*, 49(5):999–1040, 2020. doi:10.1137/18M1176622.

[CGR24] Eshan Chattopadhyay, Mohit Gurumukhani, and Noam Ringach. On the existence of seedless condensers: Exploring the terrain. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1451–1469, 2024. doi:10.1109/FOCS61266.2024.00093.

[CGRS24] Eshan Chattopadhyay, Mohit Gurumukhani, Noam Ringach, and Rocco Servedio. Condensing and extracting against online adversaries. *arXiv preprint arXiv:2411.04115*, 2024.

[CGRS25] Eshan Chattopadhyay, Mohit Gurumukhani, Noam Ringach, and Rocco Servedio. Condensing and extracting against online adversaries, 2025. URL: https://arxiv.org/abs/2411.04115, arXiv:2411.04115.

[Dil50] R. P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950. URL: http://www.jstor.org/stable/1969503.

[DMOZ23] Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Almost chor-goldreich sources and adversarial random walks. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1–9, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585134.

[DMOZ25] Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Online Condensing of Unpredictable Sources via Random Walks. In *40th Computational Complexity Conference (CCC 2025)*, pages 30:1–30:17, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CCC.2025.30, doi:10.4230/LIPIcs.CCC.2025.30.

[FRS99] Stefan Felsner, Vijay Raghavan, and Jeremy P. Spinrad. Recognition algorithms for orders of small width and graphs of small dilworth number, 1999. doi:10.17169/refubium-22108.

[GLZ24] Jesse Goodman, Xin Li, and David Zuckerman. Improved condensers for chor-goldreich sources. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1513–1549. IEEE, 2024.

[GP20] Dmitry Gavinsky and Pavel Pudlák. Santha-vazirani sources, deterministic condensers and very strong extractors. *Theory of Computing Systems*, 64(6):1140–1154, 2020.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burt Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer, Berlin, Heidelberg, 1997. doi:10.1007/BFb0052244.

[SV86] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986. URL: https://www.sciencedirect.com/science/article/pii/0022000086900449, doi:10.1016/0022-0000(86)90044-9.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 2012. URL: http://dx.doi.org/10.1561/0400000010.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il