Electronic Colloquium on Computational Complexity, Revision 1 of 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. The blocks outputted by honest nodes are independent (by default uniformly random, more generally having high min-entropy), while the blocks outputted 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 $G$ contains an "unrelated set" (an antichain under reachability) of size at least $t+1$. Whereas when 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 prove the existence of the extractor with polynomial output length and negligible error.

Our results generalize prior online source model SHELA 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

# 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 some 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, error-less condensing is impossible [GP20], while non-trivial condensing with error was shown for certain regimes of parameters [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[1]. 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).

SHELA sources model the communication pattern where messages arrive in chronological order and malicious parties output fully depends on whatever was said in the past. Many real systems can have more non-trivial dependence that is not limited to chronological total order: measurements arrive through pipelines, devices sample at scheduled times, and information propagates subject to communication delays and causality. A convenient abstraction is to discretize time into events and draw a directed edge $u \to v$ whenever the output produced at event $u$ can reach and influence event $v$ before $v$ occurs. This yields a DAG of causal influence (a standard view in distributed computing and Bayesian-network style modelling [KF09]). In such settings, even a powerful adversary that compromises some components is limited by propagation: a corrupted node can only correlate its output with what lies in its causal past, while outputs generated at causally unrelated events remain

[1] In the perfect extraction case this is necessary, while in statistical extraction all our claims extend to high entropy case in straightforward fashion.

independent. SHEDAG sources capture exactly this "adversarial but delay-bounded" regime: honest nodes contribute fresh entropy independently, and corrupted nodes may be arbitrary functions of their causal views. Our results show that the topology of causal propagation sharply determines whether one can extract uniform randomness from the entire system.

Below we formally define the SHEDAG source model along with some basic notions related to DAGs.

**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* $\mathrm{view}(v)$*, 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;*[2]
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* $\mathrm{view}(j) = \{j_1, \ldots, j_\ell\}$.

*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].

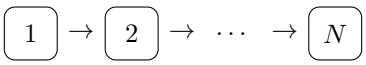


Figure 1: Base graph of a SHELA source from [AOR+20].

**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 $\Delta$ denotes statistical distance and $\mathcal{U}_m$ denotes the uniform distribution over $\{0,1\}^m$.*

**Remark 2.** *Note that while constructing an extractor for the $(n, k, G, t)$-SHEDAG source, the base graph $G$ is known to the extractor, but the positions of the good blocks are unknown.*

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 as cryptographic protocols often require uniform randomness to provide security guarantees, and instantiating such a protocol with a source that is negligibly close to uniform would imply essentially the same security guarantees. We are interested in understanding the following:

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

**Question 1.** *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 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 impossibility results.

**Zero-error randomness extraction.** When the entropy of honest blocks $k$ is less than $n$ (i.e. $k < n$), it is not hard to see[3] 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 zero-error extraction is possible if and only if the base graph $G$ contains at least $t + 1$ "unrelated vertices", 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 $G$, a zero-error randomness extraction from the $(n, G, t)$-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 for a small extraction error $\varepsilon$, then the landscape of feasibility changes drastically. To state our results, we need some additional definitions.

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

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

For our positive result, we shall consider different subgraphs of $G$ and the *head vertices* of these subgraphs. We will show that to bias the output of the extractor, corruption pattern can leave two vertices uncorrupted only if they are in the *view* of the same *head vertex*. If there is a subgraph for which adversary does not have enough corruption budget to leave at most *view* of one *head vertex* uncorrupted, then there is a low-error extractor for such $(n, G, t)$-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 5.2.

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 $\text{res}_S(G)$, is defined as*

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

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

[3] We refer to Section 2.1, paragraph "What if honest nodes are entropic.." for a sketch of the proof, and **??** for formal statement and proof.

In Remark 5 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 a (possibly non-explicit) $(n, k = \log^C n, G, t, \varepsilon = 2^{-k^{\Omega(1)}})$-SHEDAG extractor, for some absolute constant $C$.*

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 a lot 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 a (partial) 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 before $i$.

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 (or some subset of those blocks). 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).

**Comparison to [CGGL20]** Another recent interesting work that studies adversarial source model is [CGGL20]. In their work the authors define $d$-local sources where the corrupted blocks are allowed to depend arbitrarily on any subset of size at most $d$ of good blocks and give explicit construction of extractors with negligible error when $d = (\text{Number of good blocks})^{\Omega(1)}$. These local sources are not comparable to SHEDAG sources. In the SHEDAG model there might exist nodes with a full view of all nodes. It would seem that extraction in the presence of such "full view" nodes should not be possible; however, the potential dependency pattern is known (induced by the network signal propagation etc.), and when building the extractor we can just discard such "full view" nodes. In contrast, if the locality of local sources is not bounded and we do not know anything about which nodes can depend on which, the extraction is clearly not possible.

Finally, we also note that there has been recent work on somewhat correlated sources [BGM22], where there is bounded "dependence" between sources (according to some metric).

### 1.3 Future Work

Our construction of a SHEDAG extractor is semi-explicit, we take less structured object: existential generalized non-malleable multi-source extractors, and build SHEDAG extractor using those objects as blackboxes. Clearly an open question remains to construct those multi-source extractors (see Definition 12).

## 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 overview, 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 establish matching impossibility results. 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 feasibility 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 (see Section 4.3) 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 Theorem 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 are no corruptions, it is not possible to obtain perfect randomness in such scenario. Proof is inductive over the number of sources. For a single weak source it is a well known fact that perfect extraction is not possible (simply bias the output distribution of $X_1$ slightly towards $\mathsf{Ext}^{-1}(i)$ for $i = 0$ or 1). Then we proceed with the inductive step: given sources $X_1, ..., X_{t+1}$, and assume there exists an extractor $\mathsf{Ext}$ that produces perfect randomness. Fix any $n$-bit string $x$. By the inductive hypothesis we know that $\mathsf{Ext}(X_1, ..., X_t, x)$ has to be biased, otherwise we would have a perfect extractor for $t$ weak sources. Consider two distributions: $\mathsf{Ext}(X_1, ..., X_t, U)$ and $\mathsf{Ext}(X_1, ..., X_t, U')$ where $U$ is uniform over all $n$-bit strings, and $U'$ 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 $\mathsf{Ext}(X_1, ..., X_t, U')$ are perfectly uniform. Note that this may require our source to have entropy at

most $n-1$, but this can be pushed to arbitrarily close to $n$ by a slight modification of the above idea. In Lemma 5 we give a more formal statement and proof.

### 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*, that is out-deg$(v) = 0$. 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 need the notion of influence. The usual definition of an $i$th coordinate influence on a function defined over $n$-bit strings is: the fraction of inputs on which the value of the function changes if the $i$th bit is flipped. Here we need influence of a vertex which itself is a block instead of a 0/1 bit. Therefore we need a generalized version of influence and we define it in the following way. 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 15):

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

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 the adversary 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? Simply corrupting all heads will not be sufficient. Imagine a simple scenario of 4 nodes: $A, B, C, D \in \{0,1\}^n$ arranged in the $A \to B, C \to D$ way (i.e. $B$ can depend on $A$, and $D$ can depend on $C$). Let $B_1, D_1$ be first bits of $B, D$ respectively, consider a following function $f(A, B, C, D) = \langle A, C\rangle \oplus B_1 \oplus D_1$, where $\langle \cdot, \cdot \rangle$ stands for the inner product. In this example all nodes have influence and the graph has two heads: $B$ and $D$, but it is not sufficient to corrupt them, as $B$ does not carry enough information about $A$, and the whole scenario breaks down to the leakage resilience of inner product[4], and as long as $A, C$ carry enough entropy $f(A, B, C, D)$ will be uniform even if $B$ and $D$ are arbitrary functions of $A, C$ respectively. Above example illustrates the issue, *heads* might not have a way to pass all information about their *views* to other heads: $B$ cannot pass enough information about $A$ to $D$ and vice versa.

Instead, we can pick $v$ to be one of the *heads* with the largest $|\operatorname{view}(v)|$, corrupt $v$ and all nodes outside its view that still have influence: $R := V^f \setminus (\operatorname{view}(v) \cup \{v\})$. The idea is simple: $v$ has influence, and knows the values of all other nodes with influence[5], 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:

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

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

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

corruptions. Thus if number of corruptions is greater than $\operatorname{res}(G)$ 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 $\operatorname{view}(v)$, values in corruptions of the rest of influence set[6] $r$ and the values of remaining nodes with negligible influence, one would be tempted to write: $\Pr[f(v, \text{view}, r, U) = f(v, \text{view}, r, U')] > 1 - \mathsf{negl}(n)$, 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 $V^f \setminus (\operatorname{view}(v) \cup \{v\})$ 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

[4] Alternatively, one can think of the fact that the inner product has high communication complexity.

[5] Node $v$ "sees" all *honest* nodes in his *view*, and all remaining nodes (set $R$) are corrupted, and thus known to $v$. This eliminates the problem of passing all information to $v$ by other heads as there is no information to be passed: all other nodes are corrupted.

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

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 $\varepsilon$, then the influence of that node counted over "single-reset" distribution is at most $2\varepsilon$. 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. $\Pr[f(v, \text{view}, r, U) = f(v, \text{view}, r, U')] > 1 - \mathsf{negl}(n)$, even if $v, \text{view}, r$ are sampled from this not-quite-uniform distribution. This concludes that the strategy of corrupting $|V^f| - \max_{v \in V^f} |V^f \cap \text{view}(v)|$ nodes works. By the previous discussion it is also clear that if number of corruptions exceeds $\text{res}(G)$ 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.

Finally we address the third issue: can Adversary do better? The answer is negative. We can build extractor that is resilient to $\text{res}(G)$ 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[7] the set $S$ that maximizes resilience, i.e.

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

When building the 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 the extractor depend only on nodes in $S$ and completely ignore nodes outside of $S$. In fact $S$ will be the subset of vertices that have influence on our extractor.

Recall that, we use the notation $G^S$ to denote the graph on vertex set $S$ so that for each pair of vertices $u, v \in S$ we have $u$ has a path to $v$ in $G^S$ if and only if $u$ has a path to $v$ in $G$. From now on we will restrict our view only to $G^S$ instead of $G$. We start by noticing that if number of corrupted vertices $t \leq \text{res}(G)$, then number of honest vertices is at least $N - |S| + \max_{v \in S} |S \cap \text{view}(v)| + 1$ which can further be lower bounded by $\max_{v \in S} |S \cap \text{view}(v)| + 1$.

Since for every vertex $v \in G^S$ we have number of honest vertices is at least $|S \cap \text{view}(v)| + 1$, it implies that there does not exist any vertex $v$ such that, all the honest vertices are contained in $\text{view}(v)$ in $G^S$. From the definition of SHEDAG source, we know that for each vertex $v \in G^S$, output of $v$ can depend on output of $u \in G^S$ only if $u \in \text{view}(v)$. Hence, we obtain the following observation.

**Observation 1** (Informal)**.** *There does not exist any $v \in G^S$ so that $X_v$ depends non-trivially on output of all the honest vertices. Here $X_v$ denotes the output distribution of the vertex $v$.*

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

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

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 idea will not work due to the following issues:

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

First, 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 $X_v$ depends on both $X_u, X_w$ and therefore $\mathsf{2nmExt}(X_u, X_w) \oplus \mathsf{2nmExt}(X_u, X_v) \oplus \mathsf{2nmExt}(X_v, X_w)$ is fully controlled by a single corruption in $v$. These will cause fatal technical difficulties. Hence, to make progress it seems that instead of XOR-ing over all pairs, if we just XOR over pairs $(u, w)$, so that $u \neq w$, $u \notin \text{view}(w)$, $w \notin \text{view}(u)$, and hope that for at least one honest pair $(u, w)$ there does not exist any corrupted $v$ so that $u, w \in \text{view}(v)$ (the hope could be that we had so many different pairs that would exhaust the corruption budget of the adversary). The main flaw of this approach is that the existence of such pairs $(u, w)$ cannot be guaranteed.

Consider the following example:

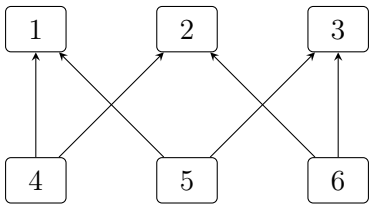


Figure 2: DAG with each pair of vertices with a common head

It is easy to check that resilience of the graph in Figure 2 is 3 and the maximum resilience is attained by the full vertex set, that is $\{1, 2, 3, 4, 5, 6\}$. Since number of corrupted vertex is at most $\text{res}(G)$, the adversary is allowed to corrupt at most 3 vertices and it chooses to corrupt vertex $1, 2, 3$. Then for every vertex pair $(i, j)$ for $i, j \in \{4, 5, 6\}$ there exists $v \in \{1, 2, 3\}$ so that $i, j \in \text{view}(v)$. From Observation 1 it can be only guaranteed that all the honest vertices cannot be *controlled* by a single corrupted vertex (which indeed is true, none of $1, 2, 3$ corrupted vertex sees all three remaining honest nodes).

Let $S$ be a set that maximizes the resilience of a graph $G$, and $\alpha = \max_{u \in G^S} |\text{view}(u) \cap S| + 1$ (in other words $\alpha = |S| - \text{res}(G^S)$). We know that when the number of corruption is at most $\text{res}(G)$, then $G^S$ has at least $\alpha$ many honest vertices and each corrupted vertex can depend only on at most $\alpha - 1$ many of the honest ones. To break this correlation we need a more general non-malleable extractor, that takes $\alpha$ many independent and entropic sources $(X_1, \ldots, X_\alpha)$ as input and it produces *close to* uniform distribution even when its output on $\ell$ many corrupted $\alpha$-tuple of sources $\{(X_1^{(i)}, \ldots, X_\alpha^{(i)}) : i = 1, \ldots, \ell\}$ is given, where for all $1 \leq i \leq \ell$ and $1 \leq j \leq \alpha$ we have $X_j^{(i)}$ depends arbitrarily on at most $\alpha - 1$ of $X_1, \ldots, X_\alpha$. More formally, we want a function $\alpha$-$\mathsf{nmExt}$ so that, for every $\alpha$ many independent entropic sources $X_1, \ldots, X_\alpha$,

$$\alpha\text{-}\mathsf{nmExt}(X_1, \ldots, X_\alpha) \approx U, \quad \text{Given}\left[\alpha\text{-}\mathsf{nmExt}(X_1^{(1)}, \ldots, X_\alpha^{(1)}), \ldots, \alpha\text{-}\mathsf{nmExt}(X_1^{(\ell)}, \ldots, X_\alpha^{(\ell)})\right],$$

where for all $i$ and $j$, we have $X_j^{(i)}$ depends on at most $\alpha - 1$ many of the input sources.

[CGGL20] gives a non-explicit construction of such generalized multi-source non-malleable extractor for all $\alpha, \ell$ (for formal statement one can look at Definition 12 and Proposition 1). Now we will define our SHEDAG extractor as: Take a $\alpha$-size subset of $S$, we call it $B$. Let $B = \{i_1 < \cdots < i_\alpha\}$. Then take $\alpha$-$\mathsf{nmExt}(X_{i_1}, \ldots, X_{i_\alpha})$ and finally take XOR over all such $\alpha$-size subsets. In our case it is enough to choose $\ell = \binom{|S|}{\alpha}$.

We know that there exists a $\alpha$-size subset $B_0 \subseteq S$ so that $(X_i)_{i \in B_0}$ is independent, entropic and for all $B \neq B_0$ if $j \in B$ then from Observation 1 we have $X_j$ depends on at most $\alpha - 1$ honest

sources. Hence, this will be extractor for SHEDAG sources when number of corrupted vertices is at most $\mathrm{res}(G)$.

We are almost done except one last issue to take care of. Similar to two source non-malleable extractor, for generalized multi-source non-malleable extractor requires, for all $i \in \{1, \ldots, \ell\}$, $(X_1^{(i)}, \ldots, X_\alpha^{(i)}) \neq (X_1, \ldots, X_\alpha)$. We can handle this by adding id of the node in the input of the extractor, that is $\alpha$-$\mathsf{nmExt}(X_{i_1} \| i_1, \ldots, X_{i_\alpha} \| i_\alpha)$. So finally our extractor will be,

$$\bigoplus_{\substack{B \subseteq S \\ |B| = \alpha \\ B = \{i_1 < \cdots < i_\alpha\}}} \alpha\text{-}\mathsf{nmExt}(X_{i_1} \| i_1, \ldots, X_{i_\alpha} \| i_\alpha) \ .$$

For formal statement and proofs one can look at Section 5.2.

A small remark, when comparing this to the perfect extraction case: if $S$ is the largest *unrelated* set then $\mathrm{res}(G) \geq |S| - \max_{v \in S} |S \cap \mathrm{view}(v)| - 1 = |S| - 1$, but $\mathrm{res}(G)$ might be much larger than that, which leads to a clear gap between perfect and statistical extraction. Simple example would be: Figure 3. 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.

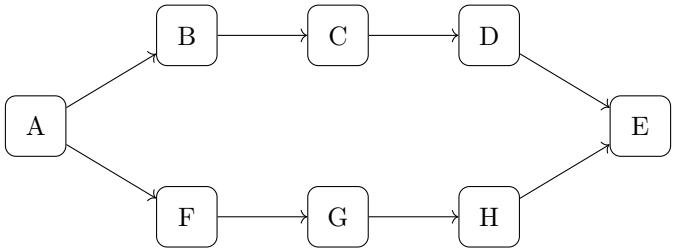


Figure 3: DAG with a small *unrelated* set (size 2), and larger resilience (equal 3).

**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: $\exists_{v_1, \ldots, v_k},\ S = \bigcup_i (\mathrm{view}\, v_i \cup \{v_i\})$. 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 $\ell$ values from $\Omega'$. 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 $\varepsilon$-*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,*

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

*If $k_1 = k_2 = k$, we call it a $(\ell, k, \varepsilon)$-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 $g_{i1}(\cdot, R)$ and $g_{i2}(\cdot, R)$ has no fixed points, it holds that,*

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

*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 $\boldsymbol{R}$.*

In the end we will define a more generalized version of two source non-malleable extractor.

**Definition 12** (Generalized $(s, \ell)$ non-malleable extractor). *A function $s\text{-}\mathsf{nmExt} : (\{0,1\}^n)^s \to \{0,1\}^m$ is called generalized $(s, \ell)$ non-malleable extractor for entropy $k$, output length $m$ and error $\varepsilon$ if the following holds: Let $\mathbf{X}_1, \ldots, \mathbf{X}_s$ be any $s$ independent $(n, k)$ sources and $g_1, \ldots, g_\ell : \{0,1\}^{ns} \to \{0,1\}^{ns}$ be any $\ell$ tampering functions so that for all $i \in [\ell]$, $g_i$ does not have any fixed point and each $g_i$ is of the form $(g_{i1}, \ldots, g_{is})$ so that each of $g_{ij}$ arbitrarily depends on at most $s-1$ many of the sources. Then,*

$$\Delta\Big(s\text{-}\mathsf{nmExt}(\mathbf{X}_1, \ldots, \mathbf{X}_s) ; \mathcal{U}_m \mid s\text{-}\mathsf{nmExt}(g_1(\mathbf{X}_1, \ldots, \mathbf{X}_s)), \ldots, s\text{-}\mathsf{nmExt}(g_\ell(\mathbf{X}_1, \ldots, \mathbf{X}_s))\Big) \leq \varepsilon$$

[CGGL20] shows existence of generalized $(s, \ell)$ non-malleable extractor for poly-logarithmic entropy. Formally the statement is as follows.

**Proposition 1** ([CGGL20, Theorem A.2]). *For all $n, k, s, \ell, m \in \mathbb{N}$ and $\varepsilon > 0$ so that $k > f(n, s, \ell, m, \varepsilon)$ and $s > 1$ there exists a generalized $(s, \ell)$ non-malleable extractor for entropy $k$, output length $m$ and error $\varepsilon$ where,*

$$f(n, s, \ell, m, \varepsilon) = \frac{m(\ell + 1)}{s} + \log n + 2\log(1/\varepsilon) + 2\log(\ell^2 + \ell) + \log s + 3 \, .$$

### 3.4 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 13** (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 14** (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 Zero-error randomness extraction from SHEDAG sources

Firstly, we proceed to give a complete characterization of the scenario under which 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 and bias the output.

To begin with, we introduce a lemma to show that it is impossible to build a zero-error extractor if input sources do not have full entropy, even when no adversary is present. The main idea is that, if the function $f$ is biased even given uniformly random inputs, then there is nothing to prove; otherwise, we slightly modify the output distributions of the sources one by one, while ensuring that their entropy never drops below $k$. Although $k$ may be arbitrarily close to $n$, these small changes suffice to introduce a nonzero bias in the output of $f$.

**Lemma 5.** *For all $n, N \in \mathbb{N}$ the following is true: for every function $f : (\{0,1\}^n)^N \to \{0,1\}$, and any $0 \le k < n$, there exists a source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ such that each $\mathbf{X}_i$ is an independent $(n, k)$-source and $\Delta(f(\mathbf{X}) \, ; \, \mathcal{U}_1) > 0$.*

*Proof.* We will prove the lemma by induction on number of blocks, that is $N$. When $N = 1$, we can assume that both $f^{-1}(0)$ and $f^{-1}(1)$ have equal size, otherwise we can take $\mathbf{X} = \mathcal{U}_n$ and $f(\mathbf{X})$ is biased. Now take $s \in f^{-1}(1)$ and define a source $\mathbf{X} \sim \{0,1\}^n$ in the following way:

$$\Pr[\mathbf{X} = s] = 2^{-k} \;\text{ and for all } s' \neq s, \; \Pr[\mathbf{X} = s'] = \frac{1 - 2^{-k}}{2^n - 1} \, .$$

Then,

$$\begin{aligned} \Pr[f(\mathbf{X}) = 1] = \Pr[f(\mathbf{X}) = 1 \mid \mathbf{X} = s] \Pr[\mathbf{X} = s] + \\ \Pr[f(\mathbf{X}) = 1 \mid \mathbf{X} \neq s] \Pr[\mathbf{X} \neq s] \end{aligned}$$

$$
\begin{aligned}
&= 2^{-k} + \frac{2^{n-1}-1}{2^n-1} \cdot (1-2^{-k}) && \text{(since, } |f^{-1}(1)| = 2^{n-1}\text{)} \\
&= \left(1 - \frac{2^{n-1}-1}{2^n-1}\right) 2^{-k} + \frac{2^{n-1}-1}{2^n-1} \\
&> \left(1 - \frac{2^{n-1}-1}{2^n-1}\right) 2^{-n} + \frac{2^{n-1}-1}{2^n-1} && \text{(since, } k < n\text{)} \\
&= \frac{2^{n-1}}{2^n-1} \cdot 2^{-n} + \frac{2^{n-1}-1}{2^n-1} = \frac{1}{2}
\end{aligned}
$$

Therefore, $f(\mathbf{X}) \not\equiv \mathcal{U}_1$ and for all $x$ we have $\Pr[\mathbf{X} = x] \leq 2^{-k}$. Hence, we are done.

Suppose for $N = \ell$ our lemma is true. Now for any function $f : (\{0,1\}^n)^{\ell+1} \to \{0,1\}$ we need to find a source $\mathbf{X} \sim (\{0,1\}^n)^{\ell+1}$ so that $f(\mathbf{X})$ is biased. Consider the function $f(X_1, \ldots, X_\ell, 0^n)$ and by inductive hypothesis, there exist $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_\ell) \sim (\{0,1\}^n)^\ell$ such that $\mathbf{Y}_1, \ldots, \mathbf{Y}_\ell$ are independent $(n,k)$-sources and $f(\mathbf{Y}, 0^n)$ is not uniform. Without loss of generality we can assume $\Pr[f(\mathbf{Y}, 0^n) = 1] > 1/2$.

Note that if $f(\mathbf{Y}, \mathcal{U}_n)$ is not uniform, our conclusion follows immediately. Otherwise, let $s \in \{0,1\}^n$ be such that

$$\Pr[f(\mathbf{Y}, s) = 1] \geq \Pr[f(\mathbf{Y}, s') = 1] \,,$$

for all $s' \in \{0,1\}^n$ and $s' \neq s$. Denote $\Pr[f(\mathbf{Y}, s) = 1] = p$ and notice $p > 1/2$ since by our choice of $s$, $\Pr[f(\mathbf{Y}, s) = 1]$ is at least $\Pr[f(\mathbf{Y}, 0^n) = 1]$. Define the source $\mathbf{Z} \sim \{0,1\}^n$ in the following way,

$$\Pr[\mathbf{Z} = s] = 2^{-k} \;\text{ and for } s' \neq s, \Pr[\mathbf{Z} = s'] = \frac{1-2^{-k}}{2^n-1} \,.$$

Then we have,

$$
\begin{aligned}
&\Pr[f(\mathbf{Y}_1, \ldots, \mathbf{Y}_\ell, \mathbf{Z}) = 1] \\
&= \Pr[f(\mathbf{Y}, \mathbf{Z}) = 1 \mid \mathbf{Z} = s] \Pr[\mathbf{Z} = s] + \Pr[f(\mathbf{Y}, \mathbf{Z}) = 1 \mid \mathbf{Z} \neq s] \Pr[\mathbf{Z} \neq s] \\
&= 2^{-k} p + \Pr[f(\mathbf{Y}, \mathbf{Z}) = 1 \mid \mathbf{Z} \neq s] \cdot (1 - 2^{-k}) \\
&= 2^{-k} p + (1 - 2^{-k}) \Pr[f(\mathbf{Y}, \mathcal{U}_T) = 1] && \text{(where } T = \{0,1\}^n \setminus \{s\}\text{)} \\
&= 2^{-k} p + \frac{1-2^{-k}}{2^n-1} \sum_{s' \neq s} \Pr[f(\mathbf{Y}, s') = 1] \\
&= 2^{-k} p + 2^{-n} \sum_{s' \neq s} \Pr[f(\mathbf{Y}, s') = 1] - \left(\frac{1}{2^n} - \frac{1-2^{-k}}{2^n-1}\right) \sum_{s' \neq s} \Pr[f(\mathbf{Y}, s') = 1]
\end{aligned}
$$

Note that we assumed $f(\mathbf{Y}_1, \ldots, \mathbf{Y}_\ell, \mathcal{U}_n)$ is uniform which implies

$$2^{-n} \left( \Pr[f(\mathbf{Y}, s) = 1] + \sum_{s' \neq s} \Pr[f(\mathbf{Y}_1, \ldots, \mathbf{Y}_\ell, s')] \right) = 1/2 \,. \tag{1}$$

Thus we have $2^{-n} \sum_{s' \neq s} \Pr[f(\mathbf{Y}, s') = 1] = 1/2 - 2^{-n} p$. Moreover, by our choice of $s$, for all $s' \neq s$ we have $\Pr[f(\mathbf{Y}, s') = 1] \leq \Pr[f(\mathbf{Y}, s) = 1] = p$. If for all $s' \neq s$ the equality holds true, from Equation (1) that will imply $p = 1/2$ which contradicts the fact that $p > 1/2$. Therefore, there exists $y \neq s$ so that $\Pr[f(\mathbf{Y}, y) = 1] < \Pr[f(\mathbf{Y}, s) = 1]$ and hence,

$$\sum_{s' \neq s} \Pr[f(\mathbf{Y}, s') = 1] < (2^n - 1) p \,.$$

Finally, combining everything we get,

$$\Pr[f(\mathbf{Y},\mathbf{Z})=1] > 2^{-k}p + \frac{1}{2} - 2^{-n}p - \left(\frac{1}{2^n} - \frac{1-2^{-k}}{2^n-1}\right)(2^n-1)p = 1/2$$

And each of $\mathbf{Y}_1,\ldots,\mathbf{Y}_\ell$ and $\mathbf{Z}$ are independent $(n,k)$ sources. Hence, by induction our lemma follows. □

This implies that we should only consider perfect extraction in SHEDAG sources with honest blocks being uniformly random. In the next two sections we will show that we can extract from $(n,G,t)$-SHEDAG sources with zero-error if and only if $t$ (i.e. number of corrupted blocks) is strictly less than size of maximum unrelated set in $G$.

### 4.1 Feasibility

We will first show that when the base graph $G$ has an unrelated set of size at least $t+1$, it is always possible to extract from any SHEDAG source over $G$ 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 $t\le N-1$ the following holds: Let $G=(V,E)$ be any directed acyclic graph with $V=[N]$ and $G$ has an 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 $T$ exists, and we denote $T=\{v_1,\ldots,v_{t+1}\}$. 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. □

**Remark 3.** *From Theorem 4 one can notice that, to efficiently construct the extractor, we need to find the maximum unrelated set of $G$ efficiently. In Section 4.3 we give an algorithm that takes an $N$-vertex DAG as input and outputs the maximum unrelated set of $G$ in $\mathrm{poly}(N)$ time.*

### 4.2 Impossibility

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

The main idea is that, when the adversary controls some related vertices and successfully corrupts the output, then it should be able to achieve the same thing by controlling only an unrelated subset of those vertices.

**Lemma 6.** *For all $n\in\mathbb{N}$, fixed $N\in\mathbb{N}$ and $t<\ell\le N$ the following holds: Let $G=(V,E)$ be any directed acyclic graph with $V=[N]$ and any unrelated set of $G$ has size at most $t$. Let $f:(\{0,1\}^n)^N\to\{0,1\}$ be any function. Then the following two are equivalent:*

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

*(ii) There exists a $(n, G, t)$-SHEDAG source $\mathbf{X}$ such 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 $(n, G, \ell)$-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 $V_{\mathcal{A}} \subseteq V$ to be the set of vertices in the source $\mathbf{X}$ which are corrupted by the adversary $\mathcal{A}$. Then $|V_{\mathcal{A}}| \leq \ell$.

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 the marginal distribution of source $\mathbf{X}$ on coordinates in the set $R$. We will first focus on allowing one of the sources at vertex $i$ or $j$ to become uncorrupted, while still keeping the output biased. Let us consider two cases:

**Case 1**: There exist $y_0, y_1, z \in \{0,1\}^n$, such that[8]

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

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

$$\mathbf{X}' = (\mathbf{X}'_i = \mathcal{U}_n, \mathbf{X}'_j = \mathcal{U}_S, \mathbf{X}'_R = \mathbf{X}_R) \tag{2}$$

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

Otherwise, assume $\Pr[f(\mathcal{U}_n, \mathcal{U}_S, \mathbf{X}_R) = 1] = 1/2$, we modify the output distribution of the source $\mathbf{X}'$ as follows:

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

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

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

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 $\Pr[f(\mathcal{U}_n, \mathcal{U}_S, \mathbf{X}_R) = 1] = 1/2$. Therefore, we have $\Delta(f(\mathbf{X}') ; \mathcal{U}_1) > 0$ and that $\mathbf{X}'$ has one less corruptions.

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

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

[8]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.

This implies that 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 $\Pr[f(\mathbf{X}) = 1] = \Pr[f(\mathbf{X}') = 1] \neq \frac{1}{2}$.

We can apply the above process repeatedly, each time reducing the number of corruptions by one, until $V_{\mathcal{A}}$ becomes a set of unrelated vertices in $G$, and thus $|V_{\mathcal{A}}| \leq t$. This concludes our proof. □

Using the above lemma, we are ready to prove the main impossibility result 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: let $G = (V, E)$ be any directed acyclic graph with $V = [N]$ and $f : (\{0,1\}^n)^N \to \{0,1\}$ be any function. Further assume that the 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 $(n, G, N)$-SHEDAG source $\mathbf{X}$ such that $\Delta(f(\mathbf{X})\ ;\ \mathcal{U}_1) = 1$. We can always find such a source by picking $x \in (\{0,1\}^n)^N$ with $f(x) = 0$ and fixing $\mathbf{X} = x$ (if such $x$ does not exist, pick $x$ so that $f(x) = 1$). Then by applying Lemma 6 the result follows. □

### 4.3 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 $\mathcal{C}$, thus the vertices in $\mathcal{C}$ 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 the sources at vertices in $\mathcal{C}$ to obtain perfect randomness.

# 5 Negligible-error randomness extraction from SHEDAG source

## 5.1 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 15** (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 16** (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 4.** *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 17** (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 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*

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

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

**Remark 5.** *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 16. 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:

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

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, $\mathrm{res}(G^f) = |V^f| - |\operatorname{view}^f(u_0)| - 1$ and number of corrupted blocks is $|V^f| - |\operatorname{view}^f(u_0)|$[9] which is at most $\mathrm{res}(G) + 1$.

[9] 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 $\mathbf{X}_R$.

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

$$\Pr[f(\mathbf{X}) = 0] \geq \Pr[f_{z_0}(\mathbf{X}_{V^f}) = 0 \mid z_0 \in \text{Good}] \ \Pr[z_0 \in \text{Good}] \tag{3}$$

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

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

**Claim 1.** $\Pr[z_0 \in \text{Good}] \geq 1 - \tilde{\delta}(n)$ *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 $f_{z_0}(X_{u_0} = x, X_{\text{view}(u_0)} = y_2, X_R = y_1) = i$ and $E_i(x', y_1, y_2)$ be the event that $f_{z_0}(X_{u_0} = x', X_{\text{view}(u_0)} = y_2, X_R = y_1) = i$ for $i = 0, 1$. Notice that, when $z_0 \in$ Good , $u_0$ has non-negligible influence on $f_{z_0}$. Hence, from Definition 15 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 and independently chosen, by symmetry, we have $\Pr_{x,x',y_1,y_2}[E_0(x, y_1, y_2) \wedge E_1(x', y_1, y_2) \mid z_0 \in \text{Good}] = \varepsilon(n)/2$. 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 (3) we can conclude that,

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

Since $\varepsilon$ 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 $n$,

$$\Pr_{z_0,x,y,z}\left[f_{z_0}(X_u = x, X_{S^u} = y) \neq f(X_u = x, X_{S^u} = y, X_S = z)\right] \leq m \cdot \delta(n) \tag{4}$$

$$\Rightarrow \Pr_{z_0,x,y,z}[f_{z_0}(X_u = x, X_{S^u} = y) = f(X_u = x, X_{S^u} = y, X_S = z)] \geq 1 - m\delta(n) \tag{5}$$

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 (4), $\mathbb{E}[Z] \geq 1 - m\delta(n)$. 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,

$$\Pr_{z_0}\left(\Pr_{x,y,z}[f_{z_0}(x, y) = f(x, y, z)] > 1 - \sqrt{m\delta(n)}\right) \geq 1 - \sqrt{m\delta(n)} \ . \tag{6}$$

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 15) and again by union bound we have,

$$\Pr_{x_0,x_1,y}[f_{z_0}(X_{u_0}=x_0, X_{S^{u_0}}=y) \neq f_{z_0}(X_{u_0}=x_1, X_{S^{u_0}}=y) \mid z_0 \in \text{Great}] \geq \varepsilon_{u_0}(n) - 2\sqrt{m\delta(n)}\,. \tag{7}$$

Since $\Pr[z_0 \in \text{Great}] \geq 1-2\sqrt{m\delta(n)}$, combining with Equation (7) 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 Equation (6) we have: With probability at least $1-\sqrt{m\delta(n)}$ over the choice of $z_0$, the following holds:

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

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 15, 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,

$$\Pr_{\substack{x \leftarrow (\{0,1\}^n)^N \\ x' \leftarrow (\{0,1\}^n)^N}}\big[f(x) \neq f(x')\big] \leq N\delta(n) \tag{8}$$

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 $c$.*

## 5.2 Extractor

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

**Definition 19** (Resilience of a graph)**.** *Given a directed acyclic graph $G = (V, E)$ and a subset $S \subseteq V$ we defined resilience of a set $S$ 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 to be strictly greater 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 this is a tight characterization, and if the number of corrupted vertices is at most $\mathrm{res}(G)$ then there exists a function (possibly non-explicit) that can perform negligible error extraction from SHEDAG sources with honest blocks having poly-log entropy.

**Theorem 9.** *There exists a universal constant $C > 1$ so that the following holds: Let $N$ be any fixed natural number and $G = (V = [N], E)$ be any DAG. Then there exists a $\left(n, k = \log^C n, G, t, 2^{-k^{\Omega(1)}}\right)$-SHEDAG extractor with output length $k^{\Omega(1)}$ when $t \leq \mathrm{res}(G)$.*

*Proof.* Without loss of generality we can assume that $\mathrm{res}(G) \geq 1$ because otherwise number of corrupted blocks is 0. By definition of resilience, this implies there exists $S \subseteq V$ such that $\mathrm{res}(S) \geq \mathrm{res}(G) \geq 1$. In Section 5.3 we will show that we can find this subset $S$ of maximum resilience in $\mathrm{poly}(N)$ time. Let $G^S = H(S)$ be the graph that we can find by the algorithm mentioned in Definition 16.

**Claim 2.** *$G^S$ has at least 2 head vertices.*

*Proof.* Observe that, for every $s \in S$ we have $|\,\mathrm{view}(s) \cap S| \leq \max_{u \in \mathrm{Head}(G^S)} |\,\mathrm{view}(u) \cap S|$, hence formally,

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

Now let's say, if possible $|\,\mathrm{Head}(G^S)| = 1$ and $h$ be the only head vertex in $G^S$. Then for all $s \in S$ so that $s \neq h$, we have $s \in \mathrm{view}(h)$. Hence, $|\,\mathrm{view}(h)| = |S| - 1$ and therefore, $\mathrm{res}(S) = 0$ which contradicts the fact that $\mathrm{res}(S) \geq 1$. Hence, $|\,\mathrm{Head}(G^S)| \geq 2$. □

Now let, $\mathbf{X} \sim \{0,1\}^{nN}$ be any $(n, k, G, t)$-SHEDAG source with $t \leq \mathrm{res}(G)$. Notice that $\mathrm{res}(G) = \mathrm{res}(S)$. We define the set

$$A_{\mathbf{X}} = \{i \in S : \mathbf{X}_i \text{ is a honest vertex}\} \,.$$

Say, $\alpha := \max_{h \in \mathrm{Head}(G^S)} |\,\mathrm{view}(h) \cap S| + 1$.

**Claim 3.** *$|A_{\mathbf{X}}| \geq \alpha$.*

*Proof.* Since number of corrupted blocks $t \leq \mathrm{res}(S)$ and by definition of resilience we have $\mathrm{res}(S) = |S| - \alpha$, our claim follows from it. □

Before stating the extractor construction we need a final observation.

**Observation 3.** *Since* $\alpha = \max_{h \in \mathrm{Head}(G^S)} |\,\mathrm{view}(h) \cap S| + 1$, *for every vertex* $v$ *in* $G^S$, *we have* $\mathrm{view}(v) < \alpha$.

We define our extractor in the following way,

$$\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) := \bigoplus_{\substack{B \subseteq S: \\ |B| = \alpha \\ B = \{i_1 < \cdots < i_\alpha\}}} \alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\alpha} \| \mathsf{bit}(i_\alpha)) \,, \tag{9}$$

where $\mathsf{bit}(i)$ denotes binary representation of $i$ for $i \in [N]$, and $\alpha$-$\mathsf{nmExt}$ is generalized $(\alpha, \ell)$ two source non-malleable extractor from Proposition 1 for $\ell = \binom{|S|}{\alpha}$ .

Recall that $A_{\mathbf{X}} = \{i \in S : \mathbf{X}_i \text{ is honest}\}$ is the set of honest vertices. From Claim 3 we have $|A_{\mathbf{X}}| \geq \alpha$. We can assume $|A_{\mathbf{X}}| = \alpha$ since if it is more than $\alpha$ we can fix all the other honest vertices while retaining only $\alpha$ many. From Observation 3 it follows that, $\mathbf{X}_j$ depends on at most $\alpha - 1$ many honest blocks if $j \notin A_{\mathbf{X}}$. Now Equation (9) looks like, $\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) =$

$$\alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1^0} \| \mathsf{bit}(i_1^0), \ldots, \mathbf{X}_{i_\alpha^0} \| \mathsf{bit}(i_\alpha^0)) \;\oplus \left( \bigoplus_{\substack{\tilde{B} \subseteq S \\ |\tilde{B}| = \alpha \\ \tilde{B} \neq A_{\mathbf{X}} \\ \tilde{B} = \{i_1 < \cdots < i_\alpha\}}} \alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\alpha} \| \mathsf{bit}(i_\alpha)) \right)$$

where $A_{\mathbf{X}} = \{i_1^0, \ldots, i_\alpha^0\}$. Notice that, for each $\tilde{B} \neq A_{\mathbf{X}}$ with $\tilde{B} = \{i_1 < \cdots < i_\alpha\}$ we can view $(\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\alpha} \| \mathsf{bit}(i_\alpha))$ in the following way: Consider the randomized function $g_{\tilde{B}} : \{0,1\}^{n\alpha} \to \{0,1\}^{n\alpha}$ so that,

$$g_{\tilde{B}}(\mathbf{X}_{i_1^0} \| \mathsf{bit}(i_1^0), \ldots, \mathbf{X}_{i_\alpha^0} \| \mathsf{bit}(i_\alpha^0)) = (\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\alpha} \| \mathsf{bit}(i_\alpha)) \,.$$

We have that each $\mathbf{X}_{i_j} \| \mathsf{bit}(i_j)$ depends on at most $\alpha - 1$ many of the input sources. Also, since $\tilde{B} \neq A_{\mathbf{X}}$, we have $(\mathsf{bit}(i_1), \ldots, \mathsf{bit}(i_\alpha)) \neq (\mathsf{bit}(i_1^0), \ldots, \mathsf{bit}(i_\alpha^0))$, or in other words, $g_{\tilde{B}}$ does not have any fixed point. Hence, by definition of generalized $(\alpha, \ell)$ non-malleable extractor (see Definition 12) we have,

$$\Delta\Bigg( \alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1^0} \| \mathsf{bit}(i_1^0), \ldots, \alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_\alpha^0}) \| \mathsf{bit}(i_\alpha^0)) \;;\; \mathcal{U}_m \;| \\ \{\alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\alpha} \| \mathsf{bit}(i_\alpha))\}_{\substack{\tilde{B}: A_{\mathbf{X}} \neq \tilde{B} \subseteq S, |\tilde{B}| = \alpha, \\ \tilde{B} = \{i_1 < \cdots < i_\alpha\}}} \Bigg) \leq \varepsilon$$

when $k > f(n, \alpha, \ell, m, \varepsilon)$, where $f$ is the function from Proposition 1. Taking XOR over all the distributions in left and right separately, from Lemma 3 we have, $\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) \approx_\varepsilon \mathcal{U}_m$. Since $\alpha, \ell$ are fixed, for some suitable choice of constant $C$, if $k \geq \log^C n$ then $\varepsilon \leq 2^{-k^{\Omega(1)}}$ and $m = k^{\Omega(1)}$ and this completes our proof.

□

### 5.3 Algorithm for locating subset with optimal resilience

**Definition 20** (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 previously, 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 $S$.

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

where $G^S = H(S)$ is the graph induced by the set $S$ defined in Definition 16. Since for all $v \in S$, if there exists $u \in \operatorname{parents}(v) \cap S$, 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 7.** *For any $S \subseteq V$, denote the set of heads in $S$ as $\operatorname{Head}(S)$. Consider $S'$ defined as follows:*

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

*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:

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

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 $S = S_H \cup \{u \in V : \exists h \in S_H, u \in \operatorname{view}(h)\}$. This means we can optimize resilience by iteratively removing nodes that are currently heads, without changing non-head vertices.

**Definition 21.** *(Intact Set) When searching for the most resilient set, we will look at sets with the following property: $v \in S \Rightarrow \operatorname{view}(v) \subseteq S$. We shall call such sets* intact*.*

The next lemma indicates that we can remove head vertices in a greedy way. Recall that by Lemma 7, it suffices to consider sets that are *intact*.

**Lemma 8.** *Suppose there exists a set $H$ that maximizes resilience, with $H \subseteq S \subseteq V$, and that $H, S$ are both intact. Consider the set of head vertices in $S$ that have maximum view: $S_0 := \{h \in S : |\operatorname{view}(h) \cap S| = \max_{p \in S}(|\operatorname{view}(p) \cap S|)\}$. 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 21, 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 in contradiction with $H$ optimizing resilience. Therefore, for any *intact* $H$ that maximizes resilience and is contained in $S$, we must have $H \cap S_0 = \emptyset$. Again, we stress here that by Lemma 7 it suffices to consider *intact* sets that maximize resilience. □

Following this lemma, we can design an algorithm as shown in Algorithm 2. Here, we present only the pseudocode to find the maximum resilience among the subsets. The rest of algorithm follows in a similar way.

For the first step (line 3), we perform a toposort and then compute and store $|\,\text{view}(h)|$ for $h \in V$ in sorted order. This takes time $O(|V| + |E|)$. Afterwards, we use an array $A$ to store the current head vertices in the same order as the size of their view, along with a graph $G$. In each step, we remove the head $h$ from $G$ with the largest view and compute the new resilience. Then we insert new head vertices, created by removing $h$, into $A$. Note that the new head vertices will always have a smaller view than $h$, thus inserted to a smaller index in the array.

In the second iteration, we use the same routine until we find the state of the array that results in the largest possible resilience. Then we record all head vertices and use Lemma 7 to retrieve the entire optimal subset by a graph traversal (e.g., BFS or DFS).

It is clear from above that the running time will be $O(|V| + |E|)$, which is asymptotically the same as simply reading the graph $G$.

### 5.3.1 Proof of correctness for Algorithm 2

We track the following *loop invariant*: before *best* is updated to the optimal resilience, at the start of each iteration of the outer while-loop, either the remaining vertices or an intact subset of them form a subset with optimal resilience.

*Initialization*: observe that the optimal $H \subseteq V$ must exist, and that we can assume $H$ is intact by Lemma 7.

*Maintenance*: let $G = (V', E')$ at the start of an iteration. If $V'$ has the optimal resilience, then we are done. Otherwise, by loop invariant, there exists an intact subset $H \subset V'$ with optimal resilience. Since we remove only head vertices, the remaining vertices in $G$ will always be *intact*. By Lemma 8, we know that $H \cap \{h \in V' : |\,\text{view}(h) \cap V'| = \max_{p \in V'}(|\,\text{view}(p) \cap V'|)\} = \emptyset$. Since the removed head vertex has the largest possible view in $V'$, 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 have been correctly updated. This concludes the proof of correctness for Algorithm 2.

**Algorithm 2** Greedy head removal

**Input:** DAG $G = (V, E)$
**Output:** The maximum resilience among subsets of $V$
Compute and store $|\operatorname{view}(h)|$ for all $h \in V$
Create array $A$ with length $|V|$ (0-index), elements initialized to empty linked list
For all heads $h$ of G, append $h$ to $A[|\operatorname{view}(h)|]$
$s \leftarrow |V|$ , $best \leftarrow 0$ , $i \leftarrow |V| - 1$
**while** $i \geq 0$ **do**
  **if** $A[i]$ is empty **then**
    $i \leftarrow i - 1$
    continue to next iteration
  **end if**
  Remove first element 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*

## References


[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.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[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