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 framework 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. Honest nodes output mutually independent blocks, which are uniformly random by default and, more generally, may have high min-entropy. Corrupted nodes output arbitrary functions of their causal views (all predecessors in G). We tightly characterize the conditions under which zero-error and negligible-error randomness extraction from SHEDAG sources is possible.

Zero-error extraction: We show that perfect extraction from SHEDAG sources with t corruptions is possible if and only if G contains an *unrelated set* (an antichain under reachability) of size at least $t+1$. In particular, if every unrelated set has size at most t, no function can extract a perfectly uniform bit. We also give a polynomial-time algorithm for finding a maximum unrelated set, thereby efficiently determining the largest corruption threshold t for which perfect extraction is possible.

Negligible-error extraction: We introduce a graph parameter, called the *resilience* of a DAG G, that characterizes the feasibility of randomness extraction with *negligible* error in the block length. We define two variants of this parameter, $\mathrm{res}_{\mathrm{uni}}(G)$ and $\mathrm{res}_{\mathrm{ent}}(G)$, corresponding to uniform and entropic honest nodes, respectively. We show that when honest nodes are uniform, negligible-error extraction is impossible when the number of corrupted nodes $t > \mathrm{res}_{\mathrm{uni}}(G)$. When honest nodes are entropic, we show that negligible-error extraction is impossible when $t > \mathrm{res}_{\mathrm{ent}}(G)$. Complementing these impossibility results, we establish the following:

(i) There exists an explicit extractor for SHEDAG sources with $t \leq \mathrm{res}_{\mathrm{uni}}(G)$ corrupted nodes and honest blocks that are uniform over $\{0,1\}^n$, achieving output length $\Omega_N(n)$ and error $2^{-\Omega_N(n)}$. Here N is number of vertices of the DAG G.

(ii) There exists a non-explicit extractor for SHEDAG sources with $t \leq \mathrm{res}_{\mathrm{ent}}(G)$ corrupted nodes and honest blocks of min-entropy $k \geq \mathrm{poly}\log n$, achieving output length $k^{\Omega(1)}$ and error $2^{-k^{\Omega(1)}}$.

The SHEDAG source model generalizes the SHELA online source model studied by Aggarwal, Obremski, Ribeiro, Siniscalchi, and Visconti (EUROCRYPT 2020) and by Chattopadhyay,

*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

Gurumukhani, and Ringach (FOCS 2024), Chattopadhyay, Gurumukhani, Ringach, and Servedio (CCC 2026). These prior works consider the special case in which the DAG G is a path, whereas we study correlation models parameterized by arbitrary fixed DAGs and determine the precise graph-dependent corruption thresholds for perfect and negligible-error extraction.

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 most 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, CGRS26].

1.1 The SHEDAG source model

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 a 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 blocks output by a corrupted node are arbitrary functions 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

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

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 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* view(v)*, is the set of all vertices u for which there is a path from u to v in G. A vertex v is not in its own view.*

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 independent randomness, possibly shared among all the vertices $j \in S$) such that, for $\{j_1, \ldots, j_\ell\} =$* view$(j)$*,*

$$\mathbf{X}_j = f_j(\mathbf{X}_{j_1}, \ldots, \mathbf{X}_{j_\ell}).$$

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.

1. *Throughout the paper, whenever we discuss the construction of an extractor or the impossibility of constructing one for a class of SHEDAG sources, we consider that class to be parameterized by a fixed directed acyclic graph G with N vertices. Accordingly, in our asymptotic analysis, we treat the DAG G which parameterizes the class of sources (and hence the number of vertices N) as fixed and the length n of the block output by each vertex as the only growing parameter.*
2. *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].*

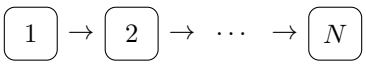

Figure 1: Base graph of a SHELA source from [AOR+20].

SHELA source is the hardest instance to be able to extract, since every corrupted block can depend on all of its predecessor and we have impossibility results on both extracting and condensing in [AOR+20, CGR24, CGRS26]. That's why we study this generalized model of correlation which is parameterized by arbitrary DAG and give tight corruption threshold-as a function of the DAG-below which extraction is possible.

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

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

Remark 2. *Note that the base graph* G *is known to the extractor while 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:

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 the next section we answer to this question.

1.2 Our contributions

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 that zero-error extraction is impossible even when G is the empty graph (and so all blocks are independent (n, k)-sources).[3] 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", a concept 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 family of* (n, G, t)*-SHEDAG sources is possible if and only if* G *has an unrelated set of size at least* $t + 1$*.*

For a formal statement and proof please look at Sections 4.1 and 4.2.

[3] We refer to Lemma 4 for the formal statement and proof.

Randomness extraction with negligible error. If we allow for a small extraction error ε, 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 the corruption pattern can leave some subset of vertices to be uncorrupted only if they are in the *view* of the same *head vertex*. If there exists a subgraph in which the adversary's corruption budget is insufficient to corrupt all vertices outside the view of a single head vertex, then a low-error extractor exists for (n, G, t)-SHEDAG sources.

This idea can be extended to give a tight corruption threshold to *entropic* SHEDAG source to be able to perform negligible-error extraction from it. Formal discussion can be found in Section 5.3.

We begin by defining the *resilience* of a DAG G, which captures the scenario mentioned in the previous paragraph. We need two variants of this parameter, which we denote by $\text{res}_{\text{uni}}(G)$ and $\text{res}_{\text{ent}}(G)$. The first one will be relevant in the context of extraction when honest nodes are uniform and the second one is relevant in when honest nodes are entropic.

Definition 6 (Resilience)**.** *Given a DAG $G = (V, E)$, and a non-empty subset $S \subseteq V$ define $d(S) := \max_{s \in S} |\,\text{view}(s) \cap S| + 1$. Now we define resilience of the set S as,*

$$\text{res}_{\text{uni}}(S) := |S| - d(S) \qquad \textit{and} \qquad \text{res}_{\text{ent}}(S) := |S| - \max\{2, d(S)\}\ .$$

We define the resilience *of G as,*

(i) $\text{res}_{\text{uni}}(G) := \max_{\emptyset \neq S \subseteq V} \text{res}_{\text{uni}}(S)$,

(ii) $\text{res}_{\text{ent}}(G) := \max_{\emptyset \neq S \subseteq V} \text{res}_{\text{ent}}(S)$.

Remark 3. *Observe that for every non-empty subset $S \subseteq V$ we have*

$$\text{res}_{\text{ent}}(S) = \begin{cases} \text{res}_{\text{uni}}(S) - 1, & \textit{if } d(S) = 1, \\ \text{res}_{\text{uni}}(S), & \textit{otherwise.} \end{cases}$$

In particular, $\text{res}_{\text{uni}}(S) - 1 \leq \text{res}_{\text{ent}}(S) \leq \text{res}_{\text{uni}}(S)$. Taking maxima over all non-empty subsets of V and using the integrality of these parameters, we obtain

$$\text{res}_{\text{ent}}(G) \in \{\text{res}_{\text{uni}}(G) - 1, \text{res}_{\text{uni}}(G)\}\ .$$

Moreover, $d(S) = 1$ if and only if no vertex in S is reachable from another distinct vertex in S, or, equivalently, S is an unrelated set (see Definition 4). Consequently,

$$\text{res}_{\text{ent}}(G) = \begin{cases} \text{res}_{\text{uni}}(G) - 1, & \textit{if every maximizer of } \text{res}_{\text{uni}} \textit{ is an unrelated set,} \\ \text{res}_{\text{uni}}(G), & \textit{otherwise.} \end{cases}$$

Indeed, if some maximizer S_0 of res_{uni} is not an unrelated set, then $\text{res}_{\text{ent}}(S_0) = \text{res}_{\text{uni}}(S_0) = \text{res}_{\text{uni}}(G)$ and hence maximizing over all subsets we get $\text{res}_{\text{uni}}(G) = \text{res}_{\text{ent}}(G)$. Otherwise, every maximizer S_0 of res_{uni} satisfies $\text{res}_{\text{ent}}(S_0) = \text{res}_{\text{uni}}(G) - 1$, while every other non-empty subset $T \subseteq V$ satisfies $\text{res}_{\text{ent}}(T) \leq \text{res}_{\text{uni}}(T) \leq \text{res}_{\text{uni}}(G) - 1$.

In Remark 8 we give an outline of the proof of $\mathrm{res}_{\mathrm{uni}}(G), \mathrm{res}_{\mathrm{ent}}(G) \geq 0$ 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, which roughly speaking show that if the number of corrupted nodes is at most $\mathrm{res}_{\mathrm{uni}}(G)$ and the honest nodes are independent and uniformly random then there exists an explicit extractor for SHEDAG sources.

Theorem 2. *Let $G = ([N], E)$ be any fixed DAG. Then there exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ so that for every (n, G, t) SHEDAG source $\mathbf{X}$ with $t \leq \mathrm{res}_{\mathrm{uni}}(G)$ we have*

$$\Delta(\mathsf{Ext}(\mathbf{X}) \,;\, \mathcal{U}_m) \leq \varepsilon \,,$$

where $m = \Omega_N(n)$ and $\varepsilon \leq 2^{-\Omega_N(n)}$. Here $\Omega_N(\cdot)$ hides constants depending on N.

The proof can be found in Section 5.2. Next, we show that if number of corrupted nodes are at most $\mathrm{res}_{\mathrm{ent}}(G)$ and honest nodes are independent and have poly-logarithmic entropy then there exists a (non-explicit) extractor for SHEDAG sources.

Theorem 3. *Fix an integer $N \geq 1$ and an N-vertex DAG G. Then, for any $t \leq \mathrm{res}_{\mathrm{ent}}(G)$ 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.*

A proof is provided in Section 5.3. Our proof of Theorem 3 can be seen as a reduction of randomness extraction from SHEDAG sources to randomness extraction in the *adversarial source model* of Chattopadhyay, Goodman, Goyal, and Li [CGGL20], and then black-box applying a (non-explicit) extractor for adversarial sources. At a high level, an adversarial source is parameterized by a number of blocks N, a number of good blocks K, and a locality parameter d. The good blocks are independent weak sources, while every bad block can depend on the values of at most d good blocks. Our proof of Theorem 3 shows that randomness extraction from SHEDAG sources reduces to randomness extraction from adversarial sources with $K = \max\{2, d(S)\}$ (where S is the set that maximizes $\mathrm{res}_{\mathrm{ent}}$ and $d(S) = \max_{s \in S} |\,\mathrm{view}(s) \cap S| + 1$) good blocks and locality $d = K - 1$, for which non-explicit extractors are known [CGGL20, Theorem 7.2].

We complement the explicit construction in Theorems 2 and 3 with an impossibility criterion. Formally we have the following theorem (combining Corollary 1 and Theorem 10).

Theorem 4 (Impossibility of negligible-error extraction)**.** *Fix an integer $N \geq 1$ and an N-vertex DAG G. Then, the following holds:*

1. *There is a constant $c > 0$ such that for every function $f : (\{0,1\}^n)^N \to \{0,1\}$ there exists an $(n, G, t = \mathrm{res}_{\mathrm{uni}}(G) + 1)$-SHEDAG source $\mathbf{X}$ so that*

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

2. *For all $k \leq n-1$, there is a constant $c' > 0$ such that for every function $g : (\{0,1\}^n)^N \to \{0,1\}$ there exists an $(n, k, G, t = \mathrm{res}_{\mathrm{ent}}(G) + 1)$-SHEDAG source $\mathbf{X}$ so that*

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

1.3 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 survey of previous models and results, and compare our SHEDAG model to other existing models.

Online source 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, CGRS26], 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 of t blocks may be corrupted, while the remaining blocks are independent and high-entropic. In contrast, in Chor–Goldreich-type sources all blocks may be (somewhat) dishonest. Second, dishonest blocks in SHELA sources may depend arbitrarily on previous blocks, including being fixed to a "worst-case" value depending on the view of each corrupted block. 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 prior models. Because of this, prior work has focused on relaxations of this task , such as extracting *somewhere-random* sources or condensing.

SHELA/oNOSF sources are a special case of SHEDAG sources corresponding to the DAG G being a path. As we show in this work, low-error (and sometimes even zero-error) randomness extraction is possible for various interesting types of DAGs (and we completely characterize all such DAGs).

Non-online source models with somewhat-dependent sources. Some source models studied in the literature consider multiple "somewhat-dependent" blocks but do not fit into the general notion of online sources discussed above. As already discussed above, the adversarial source model of [CGGL20] considers multiple blocks, some of which are good (and so are independent and entropic), while the remaining bad blocks can depend on any *bounded* number of good blocks. Measures of dependence beyond locality have also been studied [BGM22].

1.4 Future Work

We highlight the problem of constructing *explicit* negligible-error randomness extractors for low-entropy SHEDAG sources as an interesting direction for future work.

2 Technical Overview

2.1 Zero-error extraction

We begin by giving an overview of the proof of Theorem 1. Fix a DAG G. Recall that 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.

First, it is easy to see that if G contains an unrelated set of at least $t+1$, then we can extract perfectly from the family of (n, G, t)-SHEDAG sources. Indeed, it suffices to take the XOR of the nodes in this unrelated set, since we are guaranteed that at least one of these nodes is honest. Formal proof is deferred to Section 4.1.

It turns out that this is sharp, in the sense that if G has no unrelated set of size larger than t then zero-error extraction from (n, G, t)-SHEDAG sources is impossible. To see why this holds, we start with the observation that randomness extraction from $(n, G, t = |G|)$-SHEDAG sources (i.e., when all of the nodes are *corrupted*) is clearly not possible. Then, we inductively reduce the number of corruptions while maintaining positive bias: 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 with a path from u to v. We show that we can "uncorrupt" one of those two nodes, say u (i.e., set it to be uniformly random instead), and the extractor will remain biased by an appropriate adversarial choice of the still-corrupted v (although the bias might get smaller, it is still positive). There is a caveat, though. Simply "uncorrupting" u might not work, as the extractor might not depend on the node v at all. In that scenario we have to "uncorrupt" v. Nevertheless, corrupting both 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. Formal proof is deferred to Section 4.2.

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 weak k-source. We show that even if the DAG G has no edges, i.e., all nodes are independent, and there are no corruptions, zero-error randomness extraction is impossible. Our argument proceeds by induction on the number of sources. For a single weak source it is a well known fact that zero-error randomness extraction is not possible (simply bias the output distribution of X_1 slightly towards $\mathsf{Ext}^{-1}(i)$ for $i = 0$ or $i = 1$). Then we proceed with the inductive step: given sources $X_1, ..., X_{t+1}$, assume there exists an extractor 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, as otherwise we would have a perfect extractor for t weak sources. Consider the 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 4 we give a more formal statement and proof.

2.2 Negligible-error extraction

We now give an overview of the approach behind our results on negligible-error randomness extraction from SHEDAG sources. Recall that we call vertex v a *head* if there is no other vertex that has v in its view, i.e., if out-deg$(v) = 0$. Given any subset of nodes S we also consider the subgraph G^S induced by the vertices in S. The head vertices of G with respect to S are the head vertices of G^S.

Let's 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. However, the extractor may not depend on v at all, or it may depend on v very weakly only. We capture this through the notion of influence. The usual definition of the influence of the i-th coordinate on a function defined over n-bit strings corresponds to the fraction of inputs on which the value of the function changes if the i-th bit is flipped. Here we need to define the influence of a *vertex*, which corresponds to a block instead of a bit. We define the influence of a node v as follows. First, sample every node but v uniformly at random. Then, we sample the value of v uniformly at

random twice, yielding samples x_0 and x_1. The influence is then given by (for a formal definition see Definition 16)

$$\mathrm{Inf}_v^G(f) = \Pr[f(x_0, \vec{x}_{\neg v}) \neq f(x_1, \vec{x}_{\neg v})], \tag{1}$$

where $x_{\neg v}$ denotes the uniformly random values of all nodes other than 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 begin by discussing the impossibility result.

2.2.1 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 views all other nodes and it has non-negligible influence over the extractor (in the sense of Equation (1)), then we are done by setting $X_v = x_0$ or $X_v = x_1$ depending on which way we want to bias the output of the extractor f.

Notice that if the DAG has more than one head vertices with influence, simply corrupting one of them does not guarantee bias. The issue is that even if v has a full power to flip the output of the extractor, it has to know which way he is biasing the output. Consider, for example, a DAG without any edges and an extractor that just XORs all the nodes. Then, 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 a candidate extractor f, consider its *influence set* V^f, consisting of nodes with non-negligible influence (in n). For now, let's assume that all other nodes have influence 0. Consider the induced subgraph G^{V^f}, as defined before. If there is a head node v in graph G^{V^f} that has all nodes from V^f in its view then we are done: simply corrupt v. Since by definition v has non-negligible influence and it sees all inputs, it has the power to bias the output of f 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 $A \to B, C \to D$ (i.e. B can depend on A, and D can depend on C). Let B_1 and D_1 be the first bits of B and D, respectively, and consider the function $f(A, B, C, D) = \langle A, C\rangle \oplus B_1 \oplus D_1$, where $\langle \cdot, \cdot \rangle$ stands for inner product. In this example all nodes have influence and the graph has heads B and D. However, it is not sufficient to corrupt them, as B does not carry enough information about A, and the whole scenario boils down to the leakage resilience of inner product. As long as A and C carry enough entropy, the output $f(A, B, C, D)$ will be uniform even if B and D are arbitrary functions of A and C, respectively. The above example illustrates the issue that 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 view, then corrupt v and all nodes outside its view that still have influence (i.e., the set $R := V^f \setminus (\mathrm{view}(v) \cup \{v\})$). The idea is simple: v has influence and knows the values of all other nodes with influence[4], so it can bias the output of f. We have to be slightly careful here as the influence of v is defined over the uniform distribution of all other nodes. This means that the values of the corrupted nodes in R have to be set to uniformly random values, but this is fine because the only purpose of corrupting those nodes is to *know* their value.

[4]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.

There are two issues to resolve. First, the corruption strategy seems to be function-specific. Second, we assumed that nodes outside V^f had zero influence.

We address the first issue by working with the *resilience* of G, which we recall is defined as

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

($\mathrm{res}_{\mathrm{uni}}$ in the definition Definition 6) where V is a set of all nodes in graph G. The attack we discussed above required exactly

$$|V^f| - \max_{v \in V^f} |V^f \cap \mathrm{view}(v)| \leq \mathrm{res}(G) + 1$$

corruptions. Therefore, if number of corruptions is greater than $\mathrm{res}(G)$, then we can bias any candidate extractor f.

We now address the second issue where nodes outside V^f may have negligible but non-zero influence. At first sight the solution seems quite obvious: execute the attack as before, and since the influence of each node outside of V^f is negligible, they can only impact the bias of the output distribution by a negligible amount. Therefore, when corrupting v we can predict the output of the function with $1 - \mathsf{negl}(n)$ probability.

A bit more precisely (but still informally), consider grouping the inputs of f into four input classes: the value at node v, the values of nodes in $\mathrm{view}(v)$, the values of corrupted nodes in $R = V^f \setminus (\mathrm{view}(v) \cup \{v\})$, and the values of the remaining nodes with negligible influence. Then, we would be tempted to write that by independently sampling the values of this last group of nodes twice, yielding samples U and U' we would get

$$\Pr[f(v, \mathrm{view}(v), R, U) = f(v, \mathrm{view}(v), R, U')] > 1 - \mathsf{negl}(n).$$

and so we would like to conclude that the output of f is basically known to v, even though v does not know the exact values outside of V^f.

However, there is a caveat here, as the influence of the nodes outside of V^f is defined with respect to the uniform distribution of all nodes. This is precisely the reason why we pick values in $V^f \setminus (\mathrm{view}(v) \cup \{v\})$ uniformly at random, and why even the value of node v is picked as choice between two independent uniformly random samples (one can think of picking a value at random, and we have a choice to reset it once). The distribution of the value of v is not quite uniform, but we show that such a "single-reset" sampling does not impact the influence of nodes outside V^f too much. To be more precise, we show that if the influence of the node measured with respect to the uniform distribution is ε, then the influence of that node with respect to such a "single-reset" distribution is at most 2ε. Consequently, we can still conclude that the output of the function is basically already determined from the perspective of v, i.e., $\Pr[f(v, \mathrm{view}, R, U) = f(v, \mathrm{view}, R, U')] > 1 - \mathsf{negl}(n)$, even if v, view, R are sampled from this not-quite-uniform distribution. This concludes that the strategy of corrupting $|V^f| - \max_{v \in V^f} |V^f \cap \mathrm{view}(v)|$ nodes works. A formal proof can be found in Section 5.1.

Remark 4. *As shown above, for any function f, corrupting*

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

vertices suffices to introduce a non-negligible bias in the output of f when all honest nodes are uniform. When the honest nodes are independent and have entropy $k \leq n - 1$, the same argument applies, except that fewer corruptions suffice in the following special case.

Let f be any function, and let V^f denote its set of influence vertices. If G^{V^f} is an edgeless graph, it suffices to corrupt all but one vertex in V^f, rather than all of them. Indeed, write $V^f = \{u_1, \ldots, u_\ell\}$ for some $\ell \leq N$. Recall the definition of influence Equation (1). From this definition we know that there is some y_0 so that fixing $(X_{u_2}, \ldots, X_{u_\ell})$ will still keep the joint influence of vertices outside V^f negligible. So, we can fix $(X_{u_2}, \ldots, X_{u_r}) = y_0$, set X_{u_1} some honest k entropic source (might depend on y_0 but y_0 is deterministic) and set give of the vertex uniform distribution. Using almost similar idea as before it can be shown that this distribution will give non-negligible bias to the output of f.

Thus, it suffices to corrupt $|V^f| - 1$ vertices. If G^{V^f} is not edgeless, then we will corrupt $|V^f| - \max_{s \in V^f} |\operatorname{view}(s) \cap V^f| - 1$ vertices and follow exact similar argument as before. So in this case corrupting $|V^f| - \max\{2, d_f\}$ many vertices suffices, where,

$$d_f = \max_{u \in V^f} |\operatorname{view}(u) \cap V^f| + 1 \ .$$

More details can be found in Section 5.1.1.

Formally, we can show that, for every function f, there exists an (n, k, G, t)-SHEDAG source $\mathbf{X}$ with $k \leq n - 1$ and $t = \mathrm{res}_{\mathrm{ent}}(G) + 1$ such that $f(\mathbf{X})$ has non-negligible statistical distance from uniform, where

$$\mathrm{res}_{\mathrm{ent}}(G) = \max_{S \subseteq V}[|S| - \max\{2, d(S)\}] \qquad \textit{where } d(S) = \max_{s \in S} |\operatorname{view}(s) \cap S| + 1 \ .$$

In sum, the above shows that corrupting $\mathrm{res}_{\mathrm{uni}}(G) + 1$ ($\mathrm{res}_{\mathrm{ent}}(G) + 1$) nodes allows us to bias any candidate extractor by a non-negligible amount when nodes nodes are uniform (when honest nodes are entropic resp.). We now discuss our approach towards constructing a low-error extractor tolerating resilience many corruptions.

2.2.2 Extraction up to $\mathrm{res}_{\mathrm{ent}}(G)$ corruptions when honest nodes have polylogarithmic entropic.

The construction of our extractor will be based on a set S that maximizes resilience, i.e., which satisfies

$$|S| - \max\{2, d(S)\} = \mathrm{res}_{\mathrm{ent}}(G) \ ,$$

where $d(S) = \max_{s \in S} |\operatorname{view}(s) \cap S| + 1$.

We will discuss the separate algorithmic task of efficiently finding this set later on (see Section 5.4.1). Roughly speaking, S should be the set of high-influence nodes, while nodes outside S should have no influence on the output of the extractor. We will define our extractor to depend only on nodes in S.

Recall that G^S denotes the subgraph induced by vertices in S. From now on we will restrict our view only to G^S instead of G. We start by noticing that if there are $t \leq \mathrm{res}_{\mathrm{ent}}(G)$ corruptions, then number of honest vertices is at least

$$N - |S| + \max\left\{2, \max_{v \in S} |S \cap \operatorname{view}(v)| + 1\right\} \geq \max\left\{2, \max_{v \in S} |S \cap \operatorname{view}(v)| + 1\right\}.$$

Case I: $\max_{v \in S} |S \cap \operatorname{view}(v)| + 1 \geq 2$. In this case since number of honest vertex is $\max_{v \in S} |S \cap \operatorname{view}(v)| + 1$, we conclude that there does not exist any vertex v whose view in G^S contains all honest vertices In turn, from the definition of a SHEDAG source we know that the value of a vertex $v \in G^S$ can depend on the value of $u \in G^S$ only if $u \in \operatorname{view}(v)$. Hence, we obtain the following observation.

Observation 1 (Informal)**.** *Let X_v denote the output distribution of the vertex v. Then, there is no node $v \in G^S$ such that X_v depends on the value of all honest vertices.*

Corrupted nodes introduce many correlations, and a natural tool to break them are *two-source non-malleable extractors*. Roughly speaking, as long as two sources X, Y are independent and have high entropy, the output of a two-source non-malleable extractor $\mathsf{2nmExt}(X, Y)$ remains indistinguishable from uniform even given the outputs of the extractor on correlated/tampered inputs. More precisely, let $X^{(1)}, X^{(2)}, ..., X^{(t)}$ be arbitrarily correlated with X but not equal to X, symmetrically define $Y^{(1)}, ..., Y^{(t)}$, importantly $X^{(1)}, ..., X^{(t)}$ do not depend on Y, and $Y^{(1)}, ..., Y^{(t)}$ do not depend on X, then

$$\mathsf{2nmExt}(X, Y), (\mathsf{2nmExt}(X^{(i)}, Y^{(i)}))_{i\in[t]} \approx U, (\mathsf{2nmExt}(X^{(i)}, Y^{(i)}))_{i\in[t]},$$

where U is uniformly random and independent of $(\mathsf{2nmExt}(X^{(i)}, Y^{(i)}))_{i\in[t]}$. A naive idea would then be to run the non-malleable extractor over all pairs of nodes in the set S and just XOR the outputs. However, this idea will not work. First, if $u \in \text{view}(v)$ then $\mathsf{2nmExt}(X_u, X_v)$ does not help much, as a single corruption of v gives full control over this pair. Similarly, if $u, w \in \text{view}(v)$ then X_v depends on both X_u, X_w, and so $\mathsf{2nmExt}(X_u, X_w) \oplus \mathsf{2nmExt}(X_u, X_v) \oplus \mathsf{2nmExt}(X_v, X_w)$ is fully controlled by corrupting v.

Hence, to make progress it seems that instead of XOR-ing over all pairs we could just XOR over pairs (u, w) such that $u \neq w$, $u \notin \text{view}(w)$, and $w \notin \text{view}(u)$, 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 being that the many different pairs 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. For example, consider the DAG in Figure 2. It is easy to check that resilience of the graph in Figure 2 is 3 and that the maximum resilience is attained by the full vertex set, that is, by taking $S = \{1, 2, 3, 4, 5, 6\}$. Since the number of corrupted vertices is at most $\text{res}_{\text{ent}}(G)$, the adversary is allowed to corrupt at most 3 vertices and it chooses to corrupt vertices 1, 2, and 3. Then, for every vertex pair (i, j) with $i, j \in \{4, 5, 6\}$ there exists $v \in \{1, 2, 3\}$ so that $i, j \in \text{view}(v)$. Using Observation 1, it can only be guaranteed that all the honest vertices cannot be *controlled* by a single corrupted vertex (which indeed is true, as none of the corrupted vertices 1, 2, and 3 sees all three remaining honest nodes).

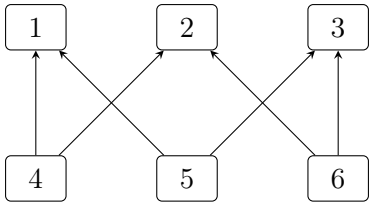

Figure 2: DAG with each pair of vertices with a common head

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 $\alpha = |S| - \text{res}_{\text{ent}}(G^S)$. We know that when the number of corruption is at most $\text{res}_{\text{ent}}(G)$, then G^S has at least α honest vertices and each corrupted vertex can depend only on at most $\alpha - 1$ honest vertices. Based on the discussion above, to break this correlation we need a more general non-malleable extractor that takes α independent and entropic sources $(X_1, \ldots, X_\alpha)$ as input and whose output is close to uniformly distributed even when its output on ℓ many corrupted

α-tuple of sources $\{(X_1^{(i)}, \ldots, X_\alpha^{(i)}) : i = 1, \ldots, \ell\}$ is given, where for all $1 \le i \le \ell$ and $1 \le j \le \alpha$ the random variable $X_j^{(i)}$ depends arbitrarily on at most $\alpha - 1$ sources of $(X_1, \ldots, X_\alpha)$. More formally, we want a function α-nmExt so that for every tuple of α independent entropic sources $X_1, \ldots, X_\alpha$ we have

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

Chattopadhyay, Goodman, Goyal, and Li [CGGL20] gave a non-explicit construction of such a generalized multi-source non-malleable extractor for all α, ℓ (for formal statements, see Definition 12 and Proposition 1). Then, we define our SHEDAG extractor as follows: Take an α-size subset of S, call it $B = \{i_1 < \cdots < i_\alpha\}$. Then, take α-$\mathsf{nmExt}(X_{i_1}, \ldots, X_{i_\alpha})$, and finally compute the XOR over all such α-size subsets. In our case it is enough to choose $\ell = \binom{|S|}{\alpha}$. Our technique is similar to [CGGL20, Theorem 7.2].

To see why this works, note that (1) there exists an α-sized subset $B_0 \subseteq S$ so that $(X_i)_{i \in B_0}$ is a tuple of independent and entropic sources, and (2) for all α-sized subsets $B \neq B_0$ and $j \in B$ we know that X_j depends on at most $\alpha - 1$ honest sources by Observation 1. As a result, the properties required for the α-nmExt to work are satisfied. To handle a small technicality (avoiding fixed points), we must actually append "IDs" to each source in the application of the α-nmExt, leading to the construction in this case,

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

Case II: $\max_{v \in S} |S \cap \mathsf{view}(v)| + 1 = 1$**.** This can only happen when G^S (graph induces by vertex set S) is empty (edge-less) graph. But in this case also there is at least 2 honest vertices in G^S since we corrupt at most $|S| - 2$ vertices. The similar idea as above works again and our final extractor will be

$$\mathsf{Ext}(X) = \bigoplus_{\substack{B \subseteq S \\ |B| = 2 \text{ and } B = \{i < j\}}} 2\text{-}\mathsf{nmExt}(X_i \| i, X_j \| j) \; .$$

Combining Case I, II our final extractor will be,

$$\mathsf{Ext}(X) = \bigoplus_{\substack{B \subseteq S \\ |B| = \beta \text{ and } B = \{i_1 < \cdots < i_\beta\}}} \beta\text{-}\mathsf{nmExt}(X_{i_1} \| i_1, \ldots, X_{i_\beta} \| i_\beta) \; ,$$

where $\beta = \max\{2, \alpha\}$ where $\alpha = \max_{s \in S} |\mathsf{view}(s) \cap S| + 1$. For more details, see Section 5.3.

Explicit extractor when honest blocks are uniform. We now explain our explicit extractor for SHEDAG sources with $t = \mathsf{res}_{\mathsf{uni}}(G)$ corruptions and whose honest blocks are uniformly random. As above, let $S \subseteq G$ be the set with maximum resilience. We denote the set of honest vertices in S by H. Notice that $|H| \ge \alpha$ where $\alpha = \max_{s \in S} |\mathsf{view}(s) \cap S| + 1$. From Observation 1 we know that there cannot exist any vertex v such that X_v depends on all the honest vertices.

Our initial approach will follow the recipe below:

(i) Look at the the view-preserving subgraph G^S on S.

(ii) Find an explicit function $f : (\{0,1\}^n)^\alpha \to \{0,1\}^m$, where α is number of honest vertices in G^S, with certain useful properties.

(iii) Consider the set $\mathcal{B}$ of all α-sized subset of vertices from S.

(iv) Apply the function f on each subset of vertices from $\mathcal{B}$ and XOR the corresponding outputs.

To make this approach work we need the function f to be resilient against the following leakage: $\{f(\{X_i\}_{i\in B})\}_{B\neq H}$. More precisely, we need that

$$f(\{X_i\}_{i\in H}), \{f(\{X_i\}_{i\in B})\}_{B\neq H} \approx U, \{f(\{X_i\}_{i\in B})\}_{B\neq H} ,$$

where U is uniform and independent of $\{f(\{X_i\}_{i\in B})\}_{B\neq H}$. However, we do not know any explicit function that satisfies such property. Hence, rather than revealing the f's output on $\{X_i\}_{i\in B}$ for every $B \neq H$, we may instead consider revealing the values of the X_i's themselves, for all the vertex i so that $i \in B$ for some $B \neq H$. This effectively amounts to fixing the outputs of f on all the *not entirely honest* subset B. At first glance, however, this approach does not work, since $\cup_{B\neq H}B$ may contain all the honest nodes, and we clearly cannot reveal them. But the key advantage of this perspective is that the information leaked by any X_i corresponding to a dishonest node $i \in \cup_{B\neq H}B$ cannot depend on all the honest vertices simultaneously, and we can exploit this structural limitation. This is precisely where the Number-on-Forehead protocol comes into the picture.

Number-on-Forehead (NOF) protocols are a multiparty communication model in which each party is assigned an input. Each party has access to all inputs *except its own*, as though its own input was written on its forehead. The parties communicate by writing messages on a shared blackboard, with the goal of jointly evaluating a function of their inputs. Each new bit added to the blackboard depends only on the inputs visible to the party writing it and on the preceding bits written on the blackboard. Formally, $\Pi : (\{0,1\}^n)^\alpha \to \{0,1\}^\mu$ is said to be computed by an NOF protocol if the input is distributed among α parties and, for each $i \in [\mu]$, there exists $S_i \in \{1,\ldots,\alpha\}$ such that the i-th output bit is a function of the preceding $i-1$ output bits and all inputs except that of the S_i-th party (see Definition 13). A result of Ford and Gal [FG13] (formalized by Chattopadhyay, Goodman, Kumar, Li, Meka, and Zuckerman [CGG+20]) gives an explicit leakage-resilient extractor NOFExt against NOF protocols producing $\mu \leq c \cdot n/2^\alpha$ output bits, for some constant $c > 0$, provided that the parties' inputs are independent and uniformly distributed over $\{0,1\}^n$. See Proposition 2 for the precise statement.

Motivated by the definition of NOF protocols and the existence of explicit leakage-resilient extractors against them, a natural approach is to apply this leakage-resilient extractor NOFExt against NOF protocols to every α-tuple of vertices and XOR the resulting outputs. To analyze this construction, we would like to reveal the blocks output by all corrupted nodes. Since each corrupted node does not depend on all the honest ones, we interpret them as NOF leakage. This strategy, however, faces an immediate obstacle: revealing even one single corrupted block incurs leakage of length n, whereas NOFExt tolerates leakage of length at most $cn/2^\alpha$, for some $0 < c < 1$.

To address this issue, we consider revealing only a short prefix of each corrupted source, keeping the total leakage at most $cn/2^\alpha$. Our goal is to fix $\mathsf{NOFExt}(\{X_i\}_{i\in B})$ for all *not-entirely-honest* sets B. However, fixing the corrupted prefixes alone may not suffice, since a set B may contain both honest and corrupted vertices. We therefore seek additional fixings that will also be an NOF leakage and, together with the corrupted prefixes, determine these extractor outputs without fully revealing all the honest blocks. To achieve this, we design our extractor so that, for every $B \neq H$, the output of NOFExt on $\{X_i\}_{i\in B}$ depends only on a short prefix of each X_i. Accordingly, our final

extractor is

$$\bigoplus_{B\subseteq S:|B|=\alpha} \mathsf{NOFExt}(\{\pi_B(X_i)\}_{i\in B}) \ ,$$

where π_B is a projection determined by the tuple B.

In this construction, rather than revealing the full values of the corrupted blocks, we reveal only the prefixes used in computing NOFExt. Our goal is therefore to keep

$$\underbrace{\sum_{i:\ i \text{ is corrupted}} \ \sum_{B:\ i\in B} |\pi_B(X_i)|}_{\substack{\text{Leakage from revealing the corrupted prefixes} \\ \text{used in the extractor computation}}} + L$$

small, where L denotes the length of the additional fixing needed to determine $\{\mathsf{NOFExt}(\{X_i\}_{i\in B})\}_{B\neq H}$. To formalize this argument and specify the additional fixing, we first fix an ordering of all α-subsets of S. For the j-th subset B_j in this ordering, we take the r_j-bit prefix of each source in $\{X_i\}_{i\in B_j}$, for some carefully chosen r_j, and apply NOFExt to these prefixes.

Suppose we fix an ordering $\prec$ of the α-subsets of S in which the set of honest vertices H appears first. For the j-th subset in this ordering, we apply NOFExt to the first $r_j = \rho^{j-1}n$ bits of each participating source, where $0 < \rho < 1$ is an appropriately chosen constant. Formally, after fixing this ordering of subsets, our extractor is

$$\bigoplus_{j=1}^{\ell} \mathsf{NOFExt}\big(\{\pi_j(X_i)\}_{i\in B_j}\big),$$

with π_j projects to the first $\rho^{j-1}n$ bits. Note that $\ell = \binom{|S|}{\alpha}$ and, by assumption, $B_1 = H$. Furthermore, the ordering of the subsets that we fixed plays a role only in the analysis of the extractor.

Since every subset containing a corrupted vertex occurs after H in the ordering, the fixing procedure described above incurs leakage of at most $\rho n/(1-\rho)$ bits for each corrupted vertex i. We additionally reveal the corresponding prefixes of each honest block that appears in a *not entirely honest* subset B. This additionally incurs leakage of at most $\rho n/(1-\rho)$ bits per honest vertex. Together, these fixings completely determine

$$\bigoplus_{j:B_j\neq H} \mathsf{NOFExt}(\{\pi_j(X_i)\}_{i\in B_j}) \ .$$

Each of these revelations can be viewed as NOF leakage. Finally, we choose ρ so that the total leakage length is at most $c\cdot n/2^{\alpha}$, allowing us to invoke the security guarantee of the NOF extractor.

However, if the honest set appears in the j-th position for some $j > 1$, then every vertex in that set contributes only $\rho^{j-1}n$ bits for the evaluation of NOFExt. Consequently, $\mathsf{NOFExt}(\{\pi_H(X_i)\}_{i\in H})$ is guaranteed to remain close to uniform only against NOF leakage of size at most $c\rho^{j-1}n/2^{\alpha}$. On the other hand, the prefix revealed from any corrupted node that appears in a predecessor set has size $\rho^{j-2}n$, which is strictly larger than the tolerated leakage bound.

To overcome this issue, we define a carefully chosen ordering of the α-subsets so that for every set appearing before the honest set we simply reveal the value of NOFExt on that set and argue that this revelation itself constitutes only a short NOF leakage. Let $\prec$ denote the ordering of the vertices in S induced by topological sorting. We extend this order to the collection of α-subsets as follows. Consider two distinct α-subsets $\{u_1,\ldots,u_\alpha\}$ and $\{v_1,\ldots,v_\alpha\}$, where $u_1 \prec \cdots \prec u_\alpha$ and $v_1 \prec \cdots \prec v_\alpha$. We declare that $\{u_1,\ldots,u_\alpha\} \prec \{v_1,\ldots,v_\alpha\}$ if there exists an index ℓ such that $u_{\ell'} = v_{\ell'}$ for every $\ell' > \ell$, and $u_\ell \prec v_\ell$.

This particular ordering allows us to establish the following key property. Suppose the honest set is $H = \{h_1 \prec \cdots \prec h_\alpha\}$ and it appears in the j-th position of the ordering. Then, for every $j' < j$, if $B_{j'} = \{b_1 \prec \cdots \prec b_\alpha\}$, there exists an index ℓ such that $b_{\ell'} = h_{\ell'}$ for all $\ell' > \ell$, while $b_\ell \prec h_\ell$. Consequently, the vertices $b_1, \ldots, b_\ell$ all appear before h_ℓ in the topological ordering. By the definition of a SHEDAG source, none of the random variables X_{b_s}, for $1 \le s \le \ell$, can depend on X_{h_ℓ}. It follows that $\mathsf{NOFExt}(\{X_i\}_{i \in B_{j'}})$ constitutes NOF leakage. By choosing the output length of NOFExt to be sufficiently small, we ensure that the total leakage contributed by all such terms remains short.

Finally, we define our extractor as

$$\bigoplus_{j=1}^{\ell} \mathsf{NOFExt}\Big(\pi_j\Big(X_{i_1^j}\Big), \ldots, \pi_j\Big(X_{i_\alpha^j}\Big)\Big) \,,$$

where $\ell = \binom{|S|}{\alpha}$, $\{i_1^j \prec \ldots \prec i_\alpha^j\}$ is the j-th set in the ordering of α-sized subsets of S, π_j denotes the projection onto the first $\rho^{j-1}n$ bits for an appropriately chosen constant $\rho < 1$, and NOFExt is a leakage-resilient extractor against NOF protocols when the inputs are uniform. For every α-subset appearing after the honest set in the ordering, we reveal the relevant prefixes of every node in that subset, while for every subset appearing before the honest set, we reveal only the output of NOFExt evaluated on prefix of the nodes of that subset. Altogether, these revelations constitute a short NOF leakage. The formal proof can be found in Section 5.2.

Algorithm to find the most resilient subset of nodes. The extractor described above relies on finding a subset that maximizes the resilience of the graph. We show that this step can be performed in time polynomial in the number of vertices V, while a naive approach requires enumerating over an exponential number of subsets. This algorithm can be used to find the corruption threshold of a given DAG, and it also may be useful for designing more efficient extractors. The key observation behind this algorithm is that there must be a maximally-resilient set S that does not truncate the view of nodes. More precisely, S can be written as the union of the complete views of its heads This significantly narrows down the space we have to search, and we can do it in time linear in the size of the graph. Formal algorithm and analysis can be found in Section 5.4.

2.3 AI disclosure.

GPT 6.0 was used to assist with checking the correctness of the arguments and polishing the manuscript. It identified a corner case we had missed regarding an edgeless graph and proposed an easy fix to that corner case by introducing an additional notion of resilience. It also found an omitted term in Theorem 8 and suggested the proof of Claim 2, relying on the proof of Claim 1, which was developed by the authors, to bound that term appropriately. All other research ideas and technical analysis in this paper came from and were developed by the authors. The authors take full responsibility for the content and correctness of the paper.

3 Preliminaries

3.1 Notation

We use uppercase roman letters such as X and Y to denote random variables. $\mathcal{U}_n$ denotes the uniform distribution over $\{0,1\}^n$. For any $T \subseteq \{0,1\}^n$, we use $\mathcal{U}_T$ to refer to the uniform distribution

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

3.2 Basic probability theory

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

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

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

□

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3.3 Extractors and non-malleable extractors

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

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

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

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

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

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

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

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

Definition 12 (Generalized (s, ℓ) non-malleable extractor)**.** *A function $s\text{-}\mathsf{nmExt} : (\{0,1\}^n)^s \to \{0,1\}^m$ is called generalized (s, ℓ) non-malleable extractor for entropy k, output length m and error ε 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 ℓ 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, ℓ) 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, ℓ) non-malleable extractor for entropy k, output length m and error ε 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 Leakage resilient extractor against Number-on-Forehead protocols

Now we will provide a formal definition of Number-on-Forehead multiparty communication protocol and explicit leakage resilient extractor against such protocols, from [CG25]. This definition and construction will be helpful to construct explicit extractor for SHEDAG sources, when the honest nodes are uniform.

Definition 13 (Number-on-Forehead Protocol)**.** *A function* $\Pi : (\{0,1\}^n)^N \to \{0,1\}^\mu$ *is called a number-on-forehead protocol (NOF protocol) if for every* $i \in \{1,\dots,\mu\}$*, there exist functions* $S_i : \{0,1\}^{i-1} \to [N]$ *and* $g_i : \{0,1\}^{i-1} \times (\{0,1\}^n)^{N-1} \to \{0,1\}$ *so that,*

$$\Pi_i(x) = g_i\big(\Pi_1(x),\dots,\Pi_{i-1}(x), x_{-S_i(\Pi_1(x),\dots,\Pi_{i-1}(x))}\big)\ ,$$

where for $x = (x_1,\dots,x_N) \in (\{0,1\}^n)^N$*, we use the notation* x_{-j} *to denote* $(x_1,\dots,x_{j-1},x_{j+1},\dots,x_N)$*. We call the output of a NOF protocol on some input as "NOF transcript" and length of the transcrupt is length of the output.*

[FG13, CGG+20] gives explicit leakage resilient extractor against NOF protocols when input distribution is uniform over $(\{0,1\}^n)^N$. Later [CG25] provides explicit construction when input distribution $\mathbf{X} = (\mathbf{X}_1,\dots,\mathbf{X}_N)$ where each $\mathbf{X}_i$ is independent entropic source. Below we state the two results formally.

Proposition 2 ([FG13, CGG+20])**.** *There exists a universal constant* $c > 0$ *so that the following holds: Let* $f : (\{0,1\}^n)^N \to \{0,1\}^m$ *be the function* $f(x_1,\dots,x_N) := (x_1 \cdot x_2 \cdot \dots \cdot x_N)_{\le m}$*, where* $x_1 \cdot x_2 \cdot \dots \cdot x_N$ *is the multiplication of the inputs over* $\mathbb{F}_{2^n}$ *and* $(x_1 \cdot \dots \cdot x_N)_{\le m}$ *is the first* m *bits of the product. If* $m \le cn/2^N$ *then for any NOF protocol* $\Pi : (\{0,1\}^n)^N \to \{0,1\}^\mu$ *with* $\mu = \lceil cn/2^N \rceil$ *we have,*

$$f(\mathcal{U}_{nN}),\ \Pi(\mathcal{U}_{nN}) \approx_\varepsilon \mathcal{U}_m,\ \Pi(\mathcal{U}_{nN})\ ,$$

where $\varepsilon = 2^{-cn/2^N}$*.*

Proposition 3 ([CG25])**.** *There exist universal constants* $C, \gamma > 0$ *so that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *so that for any* N *independent* (n,k) *sources* $\mathbf{X}_1,\dots,\mathbf{X}_N$ *where* $k \ge \log^C n$ *and any NOF protocol* $\Pi : (\{0,1\}^n)^N \to \{0,1\}^\mu$ *with* $\mu = \lceil k^\gamma/2^N \rceil$ *we have,*

$$\mathsf{Ext}(\mathbf{X}_1,\dots,\mathbf{X}_N),\ \Pi(\mathbf{X}_1,\dots,\mathbf{X}_N) \approx_\varepsilon \mathcal{U}_m,\ \Pi(\mathbf{X}_1,\dots,\mathbf{X}_N)\ ,$$

where $\varepsilon = 2^{-k^\gamma/2^N}$ *and* $m \le k^\gamma/2^N$*.*

3.5 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 14 (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 15 (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 4. *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,

$$\begin{aligned}
\Pr[f(\mathbf{X}) = 1] &= \Pr[f(\mathbf{X}) = 1 \mid \mathbf{X} = s] \Pr[\mathbf{X} = s] \ + \ \Pr[f(\mathbf{X}) = 1 \mid \mathbf{X} \neq s] \Pr[\mathbf{X} \neq s] \\
&= 2^{-k} + \frac{2^{n-1} - 1}{2^n - 1} \cdot (1 - 2^{-k}) && \text{(since, } |f^{-1}(1)| = 2^{n-1}\text{)} \\
&= \left(1 - \frac{2^{n-1} - 1}{2^n - 1}\right) 2^{-k} + \frac{2^{n-1} - 1}{2^n - 1} \\
&> \left(1 - \frac{2^{n-1} - 1}{2^n - 1}\right) 2^{-n} + \frac{2^{n-1} - 1}{2^n - 1} && \text{(since, } k < n\text{)} \\
&= \frac{2^{n-1}}{2^n - 1} \cdot 2^{-n} + \frac{2^{n-1} - 1}{2^n - 1} = \frac{1}{2}
\end{aligned}$$

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

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

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,

$$\Pr[f(\mathbf{Y}_1, \ldots, \mathbf{Y}_\ell, \mathbf{Z}) = 1]$$

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

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

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

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 (2) 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 5. *For all $n \in \mathbb{N}$, fixed $N \in \mathbb{N}$ and $t \leq N-1$ the following holds: Let $G=(V,E)$ be any directed acyclic graph with $V=[N]$ and G has 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 5. *From Theorem 5 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 5. *For all $n \in \mathbb{N}$, fixed $N \in \mathbb{N}$ and $t < \ell \leq N$ the following holds: Let $G = (V, E)$ be any directed acyclic graph with $V = [N]$ and 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, ℓ)-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 an (n, G, ℓ)-SHEDAG source for which $\Delta(f(\mathbf{X}) \,;\, \mathcal{U}_1) > 0$, i.e., $\Pr[f(\mathbf{X}) = 1] \neq 1/2$. 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 $|V_A| \leq \ell$.

Notice that from definition of SHEDAG source, $\{\mathbf{X}_j\}_{j \in V_{\mathcal{A}}}$ shares fresh independent randomness and $f(\mathbf{X})$ is biased. Hence, by averaging argument, there exist fixed shared coins for the adversary that biases the output of f on $\mathbf{X}$. So, we can assume adversary is deterministic.

Assume $|V_{\mathcal{A}}| > t$. Since every unrelated set of G has size at most t, the set $V_{\mathcal{A}}$ is not unrelated, so there exist two distinct vertices in $V_{\mathcal{A}}$ so that one of them contains the other in its view. We fix $i \in V_{\mathcal{A}}$ in the following way:

- There does not exist $u \in V_{\mathcal{A}}$ so that $i \in \mathrm{view}(u)$,
- $V_{\mathcal{A}} \cap \mathrm{view}(i) \neq \emptyset$.

Since G is a DAG and $|V_{\mathcal{A}}| > t$, such i will always exist. Let $H := \mathrm{view}(i)$ and $R := V \setminus (H \cup \{i\})$.

Notice that from the choice of i, all nodes that can see i are honest. Hence the generation of $\mathbf{X}_R$ does not depend on $\mathbf{X}_i$, and generation of $\mathbf{X}_i$ does not use $\mathbf{X}_R$ as well since $R \cap \mathrm{view}(i) = \emptyset$. Generation of $\mathbf{X}_H$ also does not depend on $\mathbf{X}_i$ since $H = \mathrm{view}(i)$. And finally since for all $j \in$

view(i), we have view$(j) \subseteq$ view(i) and generation of $\mathbf{X}_j$ can only depend on $\mathbf{X}_{\text{view}(j)}$, we also have the following conclusion: Generation of $\mathbf{X}_H$ does not depend on $\mathbf{X}_R$. Equivalently, we can generate the randomness of $\mathbf{X}$ in the following order:

$$\mathbf{X}_H \longrightarrow \mathbf{X}_R \longrightarrow \mathbf{X}_i \ .$$

Notice that this we can do because of our choice of vertex i and assuming adversary is deterministic.

For $h \in (\{0,1\}^n)^{|H|}$, we use the notation $\mathbf{X}_R^h$ to denote $\mathbf{X}_R \mid \mathbf{X}_H = h$[5]. For $y \in \{0,1\}^n$ and $h \in (\{0,1\}^n)^{|H|}$ define the following quantity,

$$P(y,h) := \Pr\left[f(\mathbf{X}_i = y,\, \mathbf{X}_H = h,\, \mathbf{X}_R^h) = 1\right],$$

and further define $\bar{P}(h) = \mathbb{E}_{y\sim\mathcal{U}_n} P(y,h)$. Let us consider two cases:

Case 1: There exist $y_0, y_1 \in \{0,1\}^n$ with $y_0 \neq y_1$ and $h \in \mathsf{supp}(\mathbf{X}_H)$ such that[6]

$$P(y_0,h) \neq P(y_1,h).$$

Since $\bar{P}(h)$ is the average of $P(\cdot,h)$, at least one of $P(y_0,h), P(y_1,h)$ is different from $\bar{P}(h)$. Without loss of generality $P(y_0,h) \neq \bar{P}(h)$. Now consider the (n,G,ℓ')-SHEDAG source $\mathbf{X}'$ in the following way: $\mathbf{X}'_H := (\mathcal{U}_n)^{|H|}$, $\mathbf{X}'_i := \mathcal{U}_n$, and keep the exactly same honest nodes and adversarial rules at corrupted nodes as $\mathbf{X}_R$. Notice that $\mathbf{X}'$ is a valid SHEDAG source with number of corruptions $\ell' < \ell$.

$$\begin{aligned}\Pr[f(\mathbf{X}') = 1] &= \mathbb{E}_{h'\sim(\mathcal{U}_n)^{|H|}} \mathbb{E}_{y\sim\mathcal{U}_n} P(y,h') \\ &= \mathbb{E}_{h'\sim(\mathcal{U}_n)^{|H|}} \bar{P}(h') \ .\end{aligned}$$

If $\Pr[f(\mathbf{X}') = 1] \neq 1/2$, then we are done. Otherwise, assume $\mathbb{E}_{h'\sim(\mathcal{U}_n)^{|H|}} \bar{P}(h') = 1/2$, and define a new source $\tilde{\mathbf{X}}$ as follows: $\tilde{\mathbf{X}}_H := (\mathcal{U}_n)^{|H|}$, $\tilde{\mathbf{X}}_R := \mathbf{X}'_R$ and

$$\tilde{\mathbf{X}}_i = \begin{cases} y_0 & \text{If } \tilde{\mathbf{X}}_H = h \\ \mathcal{U}_n & \text{Otherwise.} \end{cases}$$

Notice $\tilde{\mathbf{X}}$ is a valid $(n,G,\tilde{\ell})$ SHEDAG source with $\tilde{\ell} < \ell$. Moreover,

$$\begin{aligned}\Pr[f(\tilde{\mathbf{X}}) = 1] &= \mathbb{E}_{h'\sim\tilde{\mathbf{X}}_H} \mathbb{E}_{y'\sim\tilde{\mathbf{X}}_i|\tilde{\mathbf{X}}_H=h'} P(y',h') \\ &= 2^{-n|H|} P(y_0,h) + 2^{-n|H|} \sum_{h'\neq h} \mathbb{E}_{y'\sim\mathcal{U}_n} P(y',h') \\ &= 2^{-n|H|} P(y_0,h) + 2^{-n|H|} \sum_{h'\neq h} \bar{P}(h') \\ &= 2^{-n|H|} P(y_0,h) + (1/2 - 2^{-n|H|}\bar{P}(h)) \\ &\neq 1/2.\end{aligned}$$

Above the second equality holds from the way we defined $\tilde{\mathbf{X}}_H$ and $\tilde{\mathbf{X}}_i$ respectively. The fourth equality holds because of our assumption $\mathbb{E}_{h'\sim(\mathcal{U}_n)^{|H|}} \bar{P}(h') = 1/2$ and last inequality holds since $P(y_0,h) \neq \bar{P}(h)$. Therefore $\Delta(f(\tilde{\mathbf{X}})\,;\mathcal{U}_1) > 0$ and $\tilde{\mathbf{X}}$ has at least one less corruption.

[5]It is possible that $\mathsf{supp}(\mathbf{X}_H) \neq (\{0,1\}^n)^{|H|}$, but we can always extend the adversarial strategy arbitrarily when it sees some new value in its view which is not in $\mathsf{supp}(\mathbf{X}_H)$, and this does not change the proof.

[6]Note that such a tuple might not exist; the crucial observation is that P is defined with respect to the conditional distribution $\mathbf{X}_R^h$, and not with respect to the marginal distribution of $\mathbf{X}_R$.

Case 2: For all $h \in \mathsf{supp}(\mathbf{X}_H)$ and all $y_0, y_1 \in \{0,1\}^n$ we have

$$P(y_0, h) = P(y_1, h).$$

This implies that $P(y,h) = \bar{P}(h)$ for all y, i.e., its value depends only on h. Now define the $(n, G, \hat{\ell})$-SHEDAG source $\hat{\mathbf{X}}$ with $\hat{\ell} < \ell$ as follows: $(\hat{\mathbf{X}}_H, \hat{\mathbf{X}}_R) = (\mathbf{X}_H, \mathbf{X}_R)$ and $\hat{\mathbf{X}}_i = \mathcal{U}_n$. Let $\mathbf{X}_i^h$ be the distribution $\mathbf{X}_i \mid \mathbf{X}_H = h$. Then we have

$$\begin{aligned}
\Pr[f(\hat{\mathbf{X}}) = 1] &= \mathbb{E}_{h' \sim \mathbf{X}_H} \mathbb{E}_{y \sim \mathcal{U}_n} P(y, h') \\
&= \mathbb{E}_{h' \sim \mathbf{X}_H} \bar{P}(h') \\
&= \mathbb{E}_{h' \sim \mathbf{X}_H} \mathbb{E}_{y \sim \mathbf{X}_i^{h'}} P(y, h') \\
&= \Pr[f(\mathbf{X}) = 1] \neq \frac{1}{2}.
\end{aligned}$$

We can apply the above process repeatedly, each time reducing the number of corruptions by at least one, until $|V_A| \leq t$. This concludes our proof.

□

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

Theorem 6. *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 5 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 7 (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 7 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 5, it is possible to extract from the sources at vertices in $\mathcal{C}$ to obtain perfect randomness.

5 Negligible-error randomness extraction from SHEDAG sources

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 16 (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:*

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

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 17 (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 are 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 6. *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 18 (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.*

Remark 7. *Note that, the DAG is fixed and number of vertex N is constant and only the growing parameter is n, where n is the length of the string outputted by each vertex. That's why, when we say negligible, we mean negligible as a function of n and we study $\mathsf{negl}(n)$ error extraction from SHEDAG source.*

Now recall the definition of resilience of a directed acyclic graph G.

Definition 19 (Resilience)**.** *Given a DAG $G = (V, E)$, with $|V| \geq 2$, and a non-empty subset $S \subseteq V$ define $d(S) := \max_{s \in S} |\operatorname{view}(s) \cap S| + 1$. Now we define resilience of the set S as,*

$$\operatorname{res}_{\text{uni}}(S) := |S| - d(S) \qquad \text{and} \qquad \operatorname{res}_{\text{ent}}(S) := |S| - \max\{2, d(S)\} .$$

We define the resilience *of G as,*

(i) $\operatorname{res}_{\text{uni}}(G) := \max_{\emptyset \neq S \subseteq V} \operatorname{res}_{\text{uni}}(S)$,

(ii) $\operatorname{res}_{\text{ent}}(G) := \max_{\emptyset \neq S \subseteq V} \operatorname{res}_{\text{ent}}(S)$.

Remark 8. *Note that for every $S \subseteq V$, number of head vertices in G^S is at least 1 and*

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

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

Take any arbitrary set $T = \{u, v\} \subseteq V$. Then clearly $d(T) \leq 2$ which implies $\max\{2, d(T)\} = 2$. Hence $\operatorname{res}_{\text{ent}}(T) \geq 0$ and taking maximum over all non-empty subsets of V we have $\operatorname{res}_{\text{ent}}(G) \geq 0$.

We are now ready to state our results on the impossibility of extraction from SHEDAG sources. At first we consider the case where honest nodes are uniform. In this case, the next theorem shows that, for a given directed acyclic graph G and a function f, if some vertex has non-negligible influence, then f cannot extract from SHEDAG sources with $\operatorname{res}_{\text{uni}}(G) + 1$ 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 $\operatorname{Inf}_v^G(f) \notin \mathsf{negl}(n)$. 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 17. For every vertex $u \in V^f$ we define $\operatorname{view}^f(u) = \operatorname{view}(u) \cap V^f$.

Theorem 8 (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 $t = \operatorname{res}_{\text{uni}}(G) + 1$ 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 $u_0 \in \operatorname{Head}(G^f)$ so that for all $u \in \operatorname{Head}(G^f)$ 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:

$\mathbf{X}_R = (\mathcal{U}_n)^{|R|}$, $\mathbf{X}_{\text{view}^f(u_0)} = (\mathcal{U}_n)^{|\text{view}^f(u_0)|}$, then sample x, x' uniformly and independently from $\{0,1\}^n$, when $\mathbf{X}_R = y_1$ and $\mathbf{X}_{\text{view}^f(u_0)} = y_2$ define $\mathbf{X}_{u_0}$ as:

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

Here, $E(x, y_1, y_2)$ is the event that: number of $z' \in (\{0,1\}^n)^m$ so that $f(X_{u_0} = x, X_{\text{view}(u_0)} = y_2, X_R = y_1, X_{V \setminus V^f} = z') = 0$ 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 $t = \text{res}_{\text{uni}}(G) + 1$. As, $|\text{view}^f(u_0)|$ is at least $|\text{view}^f(u)|$ for every $u \in G^f$ and every vertex of G^f is in view of some head vertex, we have

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

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

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

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

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 8 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 16 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

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

Assume y_1, y_2 are uniformly sampled and fixed. Then if for all x we had the guarantee that both $E(x, y_1, y_2)$ and $E_0(x, y_1, y_2)$ occurs or none of these two occurs (if $E_0(x, y_1, y_2)$ does not occur we set value of u_0 by x') then the above expression is same as,

$$\Pr_{x,y_1,y_2}[E_0(x, y_1, y_2) \mid z_0 \in \text{Good}] + \Pr_{x,x',y_1,y_2}[E_0(x', y_1, y_2) \wedge E_1(x, y_1, y_2) \mid z_0 \in \text{Good}] .$$

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

But the only way this can be different when one of $E(x, y_1, y_2)$ and $E_0(x, y_1, y_2)$ occurs and the other does not. Hence,

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

Define the following event

$$F(x, y_1, y_2) := [(E(x, y_1, y_2) \wedge \neg E_0(x, y_1, y_2)) \vee (\neg E(x, y_1, y_2) \wedge E_0(x, y_1, y_2))] .$$

Claim 2. $\Pr_{x,y_1,y_2}[F(x, y_1, y_2) \mid z_0 \in \text{ Good }] \leq \delta'''(n)$ *for some* $\delta''' \in \mathsf{negl}(n)$.

Proof of Claim 2 can be found after the proof of Claim 1. As, $\Pr_{x,y_1,y_2}[E_0(x, y_1, y_2) \mid z_0 \in \text{Good}] \geq 1/2 - \delta'(n)$ combining with Claim 2 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 ε is non-negligible function and $\delta'(n) \in \mathsf{negl}(n)$ we have $\varepsilon(n)/2 - \delta'(n)$ is non-negligible. Also, $\tilde{\delta}(n)$ is a negligible function. Hence $\Pr[f(\mathbf{X}) = 0] \geq 1/2 + \varepsilon_0(n)$ for some $\varepsilon_0(n) \notin \mathsf{negl}(n)$ and from the definition of statistical distance our proof follows.

□

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

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

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

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,

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

is at least $1 - 2\sqrt{m\delta(n)}$. From the definition of influence (see Definition 16) and again by union bound we have,

$$\Pr_{x_0,x_1,y}[f_{z_0}(X_{u_0} = x_0, X_{S^{u_0}} = y) \neq f_{z_0}(X_{u_0} = x_1, X_{S^{u_0}} = y) \mid z_0 \in \text{Great}] \tag{7}$$

is at least $\varepsilon_{u_0}(n) - 2\sqrt{m\delta(n)}$. 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 $\text{Inf}^G_{u_0}(f_{z_0}) \geq \varepsilon_{u_0}(n) - 2\sqrt{m\delta(n)}$.

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.

□

Proof of Claim 2. Recall the definition of the set $\text{Great} \subseteq (\{0,1\}^n)^m$ from Claim 1, where $z' \in (\{0,1\}^n)^m$ we have $z' \in \text{Great}$ if

$$\Pr_{x,y,z}[f_{z'}(x,y) = f(x,y,z)] > 1 - \sqrt{m\delta(n)}\,.$$

Here $\delta(n)$ is such a function so that influence of each vertex outside V^f is bounded by $\delta(n)$ for large enough n. Since we are calculating the probability under the condition $z_0 \in \text{Good}$, from here we immidiately have that $z_0 \in \text{Great}$.

Further recall that $E(x, y_1, y_2)$ is the event that $f(x, y_1, y_2, X_S)$ is biased towards 0, where $S = V \setminus V^f$ and $E_0(x, y_1, y_2)$ is the event that $f(x, y_1, y_2, z_0) = 0$. Now for some x, y_1, y_2 if we have $E(x, y_1, y_2) \wedge \neg E_0(x, y_1, y_2)$ that implies $f(x, y_1, y_2, z_0) = 1$ but for at least half of the elements $z \in (\{0,1\}^n)^m$ we have $f(x, y_1, y_2, z) = 0$. And $\neg E(x, y_1, y_2) \wedge E_0(x, y_1, y_2)$ implies $f(x, y_1, y_2, z_0) = 0$ but for at least half of the elements $z \in (\{0,1\}^n)^m$ we have $f(x, y_1, y_2, z) = 1$. Therefore,

$$\begin{aligned}\frac{1}{2}\Pr_{x,y_1,y_2}[F(x, y_1, y_2) \mid z_0 \in \text{Great}] &\leq \Pr_{x,y_1,y_2,z}[f_{z'}(x, y_1, y_2) \neq f(x, y_1, y_2, z)] \\ &\leq \sqrt{m\delta(n)}\end{aligned}$$

Take $\delta'''(n) = 2\sqrt{m\delta(n)}$ and it completes our proof. □

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

Theorem 9. *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 $u \in V$, $\text{Inf}^G_u(f) \in \mathsf{negl}(n)$. 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 16, 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,

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

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 8 and Theorem 9 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 $t = \mathrm{res}_{\mathrm{uni}}(G) + 1$ so that $\Delta(f(\mathbf{X}) \,;\, \mathcal{U}_1) \geq n^{-c}$ for some constant c.*

5.1.1 Impossibility proof sketch when honest nodes are entropic.

Using an argument similar to the one above, we show that, when honest nodes are entropic, corrupting $\mathrm{res}_{\mathrm{ent}}(G)+1$ vertices (see Definition 19) suffices to introduce a non-negligible bias in the extractor's output. For any function g, if $|V^g| \neq \emptyset$ and G^{V^g} is an edgeless graph, then $\mathrm{res}_{\mathrm{ent}}(V^g) = |V^g| - 2$ by the definition of $\mathrm{res}_{\mathrm{ent}}(G)$. Let $V^g = \{u_1, \ldots, u_\ell\}$ and denote $R = V \setminus V^g$. Denote the set $\{u_2, \ldots, u_\ell\}$ by S. We can show that only corrupting the vertices in S we can produce a SHEDAG source that can produce non-negligible bias on the output of g.

Now from the definition of influence and by a hybrid argument,

$$\Pr_{x,y,z,z'}[g(X_{u_1} = x, X_S = y, X_R = z) \neq g(X_{u_1} = x, X_S = y, X_R = z')] \leq N\delta(n)\ , \tag{9}$$

for some negligible function δ, when x, y, z, z' are independently and uniformly sampled. That means there exists y_0 so that,

$$\Pr_{x,z,z'}[g(X_{u_1} = x, X_S = y_0, X_R = z) \neq g(X_{u_1} = x, X_S = y_0, X_R = z')] \leq N\delta(n)\ . \tag{10}$$

Without loss of generality, assume that the function $g_{y_0}(X_{u_1}, X_R) := g(X_{u_1}, X_S = y_0, X_R)$ is biased towards 0. Define the following function :

$$p(x) = \Pr_{r \leftarrow (\{0,1\}^n)^{|R|}}[g_{y_0}(X_{u_1} = x, X_R = r) = 0]\ .$$

Let A is the set of 2^{n-1} inputs with largest values of $p(x)$. Define the following source $\mathbf{Y} = \mathcal{U}_A$. Now we are ready to define our SHEDAG source $\mathbf{X}$: $\mathbf{X}_S = y_0$, $\mathbf{X}_{u_1} = \mathbf{Y}$ and finally $\mathbf{X}_R = (\mathcal{U}_n)^{|R|}$. Clearly $\mathbf{X}_{u_1}$ has min-entropy $n-1$ and it is not corrupted since y_0 is fixed and known to it.

Claim 3. $\Delta(g(\mathbf{X})\,;\mathcal{U}_1) \geq 1/2 - \delta'(n)$ *for some negligible function* δ'.

Proof. By the choice of y_0 and the definition of function p, $\mathbb{E}_{x\sim\mathcal{U}_n}[p(x)] \geq 1/2$. Now, from Equation (10) we have,

$$\begin{aligned}
N\delta(n) &\geq \Pr_{x,z,z'}[g_{y_0}(X_{u_1} = x, X_R = z) \neq g_{y_0}(X_{u_1} = x, X_R = z')] \\
&= \Pr_{x,z,z'}[g_{y_0}(X_{u_1} = x, X_R = z) = 0 \wedge g_{y_0}(X_{u_1} = x, X_R = z') = 1]] \\
&+ g_{y_0}(X_{u_1} = x, X_R = z) = 1 \wedge g_{y_0}(X_{u_1} = x, X_R = z') = 0] \\
&= \mathbb{E}_x[2p(x)(1-p(x))] \\
&\geq \mathbb{E}_x[\min\{p(x), 1-p(x)\}]
\end{aligned}$$

Now if $p(x) \geq 1/2$ for all $x \in A$ then clearly for all $x \in A$, we have $\min\{p(x), 1-p(x)\} = 1-p(x)$. In this case,

$$\begin{aligned}
\mathbb{E}_{x\leftarrow\mathcal{U}_A}[1-p(x)] &= 2\mathbb{E}_{x\leftarrow\mathcal{U}_n}[\mathbf{1}_A(x)\min\{p(x), 1-p(x)\}] \\
&\leq 2\mathbb{E}_{x\leftarrow\mathcal{U}_n}[\min\{p(x), 1-p(x)\}] \\
&\leq 2N\delta(n)
\end{aligned}$$

Above, $\mathbf{1}_A$ is the characteristic function of the set A, that is $\mathbf{1}_A(x) = 1$ if $x \in A$, otherwise 0.

Hence from above we have $\mathbb{E}_{x\leftarrow\mathcal{U}_A}[p(x)] \geq 1-2N\delta(n)$. If there is some $x \in A$ so that $p(x) < 1/2$ then from the choice of A we have for all $x \in \{0,1\}^n \setminus A$, $p(x) < 1/2$. Using the fact that $\mathbb{E}_x[p(x)] \geq 1/2$, in this case also we can show that $\mathbb{E}_{x\leftarrow\mathcal{U}_A}[p(x)] \geq 1-2N\delta(n)$. Take $\delta'(n) = 2N\delta(n)$. Since N is fixed and δ is negligible, δ' is also negligible. Therefore finally we have,

$$\Delta(g(\mathbf{X})\,;\mathcal{U}_1) \geq 1/2 - \delta'(n)\,.$$

□

On the other hand, when G^{V^g} contains an edge, we have $\mathrm{res}_{\mathrm{ent}}(V^g) = |V^g| - \max_{u\in V^g} |\,\mathrm{view}(u) \cap V^g| - 1$. In this case, exactly similar argument as in Theorem 8 applies. We therefore obtain the following analogue of the theorem.

Theorem 10. *For all large enough* $n, k \in \mathbb{N}$ *with* $k \leq n-1$ *and fixed* $N \in \mathbb{N}$ *the following holds: Let* $G = (V, E)$ *be a directed acyclic graph with* $V = [N]$ *and* $N \geq 2$. *Then for every function* $g : (\{0,1\}^n)^N \to \{0,1\}$ *there exists a* (n, k, G, t)*-SHEDAG source with* $k \leq n-1$ *and* $t = \mathrm{res}_{\mathrm{ent}}(G) + 1$ *so that* $\Delta(g(\mathbf{X})\,;\,\mathcal{U}_1) \geq n^{-c'}$ *for some constant* c'.

5.2 Explicit extractor when honest nodes are uniform

Now we will give an explicit construction of SHEDAG sources when honest nodes are uniform. We will start by recalling the definition of resilience of a directed acyclic graph.

Definition 20 (Resilience of a graph in context of uniform honest vertices)**.** *Given a directed acyclic graph* $G = (V, E)$ *with* $|V| \geq 2$ *and a non-empty subset* $S \subseteq V$ *we defined resilience of a set* S *as*

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

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

Before staring the proof, we need the following definition.

In this section, for any DAG G with vertex set $[N]$ and any two vertex $u, v \in [N]$ we will use the notation $u \prec v$ to denote that u comes before v in topological sorting of vertices of G, that is u has a path to v in G.

Before stating the main theorem of this section we need to define a ordering between q-tuples of vertices of a DAG G, for some q, which will be useful.

Definition 21 (Colex ordering)**.** *Let $G = ([N], E)$ be a DAG and $1 \prec 2 \prec \cdots \prec N$. For any $q > 1$ and two distinct q tuples $(u_1, \ldots, u_q), (v_1, \ldots, v_q) \in [N]^q$ such that $u_1 \prec \ldots \prec u_q$ and $v_1 \prec \ldots \prec v_q$, we say $(u_1, \ldots, u_q) \prec (v_1, \ldots, v_q)$ if the following holds: If $j \in [q]$ be the highest index such that $u_j \neq v_j$, then $u_j \prec v_j$.*

Now we are ready to state our main theorem.

Theorem 11. *Let $G = ([N], E)$ be any fixed DAG with $N \geq 2$. Then there exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ so that for every (n, G, t) SHEDAG source $\mathbf{X}$ with $t \leq \mathrm{res}_{\mathrm{uni}}(G)$,*

$$\mathsf{Ext}(\mathbf{X}) \approx_\varepsilon \mathcal{U}_m \ ,$$

where $m = \Omega_N(n)$ and $\varepsilon \leq 2^{-\Omega_N(n)}$. Here $\Omega_N(.)$ hides constants depending on N.

Proof. Recall the definition of resilience of subset $S \subseteq [N]$,

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

and resilience of the DAG G is, $\mathrm{res}_{\mathrm{uni}}(G) = \max_{\emptyset \neq S \subseteq [N]} \mathrm{res}_{\mathrm{uni}}(S)$. Let $S \subseteq [N]$ be the subset of vertices with maximum resilience. We can find such S efficiently (see Section 5.4). Let, $S = \{s_1, \ldots, s_p\}$. Notice that we can have a ordering $\prec$ restricted to the set S and assume $s_1 \prec s_2 \prec \ldots \prec s_p$.
Let $\alpha := \max_{s \in S} |\operatorname{view}(s) \cap S| + 1$. Since number of corruptions is at most $\mathrm{res}(G)$, we have number of honest blocks at least $N - |S| + \alpha$, which is at least α.

Observation 3. *From the definition of α we have, for every vertex v in S, $|\operatorname{view}(v) \cap S| < \alpha$.*

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

$$B := \{(v_1, \ldots, v_\alpha) : v_i \in S \text{ and } v_1 \prec \ldots \prec v_\alpha\} \ .$$

Note that we can further order elements of B by the colex ordering for tuples as defined in Definition 21. Let $B = \{b_1 \prec b_2 \prec \ldots \prec b_L\}$ where $L = \binom{|S|}{\alpha}$. Now we define our extractor as:

$$\mathsf{Ext}(\mathbf{X}_1, \ldots, \mathbf{X}_N) := \bigoplus_{\substack{b_i \in B \\ b_i = (v_1, \ldots, v_\alpha)}} T_{r_i, m}(\pi_{r_i}(\mathbf{X}_{v_1}), \ldots, \pi_{r_i}(\mathbf{X}_{v_\alpha})) \ .$$

Here for $1 \leq i \leq L$, $r_i = \rho^{i-1} n$ and $m = c \cdot \beta r_L$ where $\rho = c/(c + N2^{N+2})$ and $\beta = 1/(2^{N+2} N^N)$ where c being the constant from Proposition 2 and

(i) $\pi_{r_i} : \{0,1\}^n \to \{0,1\}^{r_i}$ is the projection map on first r_i bits,

(ii) $T_{r_i, m} : (\{0,1\}^{r_i})^\alpha \to \{0,1\}^m$ is the explicit leakage resilient extractor against NOF protocols with NOF transcript length $\mu = c r_i / 2^\alpha$, from Proposition 2.

Notice that, without loss of generality we can assume, number of good vertices is exactly α since if it is more than α we can fix all the other honest vertices while retaining only α many.
Let for some $1 \leq i_0 \leq L$, $b_{i_0} = (u_1, \ldots, u_\alpha) \in B$ so that for all i, u_i is honest, that is each of $\mathbf{X}_{u_1}, \ldots, \mathbf{X}_{u_\alpha}$ are independent and each of them is uniform over $\{0,1\}^n$. Then each of $\pi_{r_{i_0}}(\mathbf{X}_{u_1}), \ldots, \pi_{r_{i_0}}(\mathbf{X}_{u_\alpha})$ is also uniform over $\{0,1\}^{r_{i_0}}$ and they are independent.

We start by fixing all the vertices that does not hold any of u_i in its view. Next we do some further fixing which we analyze in following two cases:

(i) For any arbitrary $i < i_0$ consider ith tuple b_i which comes before b_{i_0} in colex ordering (see Definition 21), that is $b_i \prec b_{i_0}$. Assume $b_i = (w_1, \ldots, w_\alpha)$. Then by the definition of colex ordering of tuples (see Definition 21) we have some index $j \in [\alpha]$ so that $w_j \prec u_j$ and for all $j' > j$ we have $w_{j'} = u_{j'}$. Therefore we can rewrite b_i as $(w_1, \ldots, w_j, u_{j+1}, \ldots, u_\alpha)$ where

$$w_1 \prec w_2 \prec \ldots \prec w_j \prec u_j \ .$$

Since in topological sorting, u_j appears after all of $w_1, \ldots, w_j$, from there it follows that there is no path from w_ℓ to u_j in G^S (for definition of G^S see Definition 17) for $\ell = 1, \ldots, j$. Hence, $\mathbf{X}_{w_1}, \ldots, \mathbf{X}_{w_j}$ cannot depend on $\mathbf{X}_{u_j}$. So, $T_{r_i,m}(\pi_{r_i}(\mathbf{X}_{w_1}), \ldots, \pi_{r_i}(\mathbf{X}_{w_\alpha}))$ is a function of $(\mathbf{X}_{u_1}, \ldots, \mathbf{X}_{u_{j-1}}, \mathbf{X}_{u_{j+1}}, \ldots \mathbf{X}_{u_\alpha})$.

Since the above argument holds for every i less than i_0, we have the following conclusion: For all $i < i_0$, output of $T_{r_i,m}$ on $(\pi_{r_i}(\mathbf{X}_{a_1^i}), \ldots, \pi_{r_i}(\mathbf{X}_{a_\alpha^i}))$ where $b_i = (a_1^i, \ldots, a_\alpha^i)$, cannot depend on all the honest nodes. Hence, we reveal

$$\left(T_{r_i,m}(\pi(\mathbf{X}_{a_1^i}), \ldots, \pi(\mathbf{X}_{a_\alpha^i}))\right)_{i=1}^{i_0-1}, \quad \text{where } b_i = (a_1^i, \ldots, a_\alpha^i) \ ,$$

and from the above discussion we know that, what we reveal is a NOF transcript (see Definition 13) of length $(i_0 - 1)m$.

(ii) Next, we consider any arbitrarily chosen $e > i_0$ and we have $b_{i_0} \prec b_e$. Assume $b_e = (y_1, \ldots, y_\alpha)$. Therefore, the input tuple of distributions to $T_{r_e,m}$ will be $(\pi_{r_e}(\mathbf{X}_{y_1}), \ldots, \pi_{r_e}(\mathbf{X}_{y_\alpha}))$. Notice that from Observation 3 we have,

$$\text{There does not exist any } y_j \text{ for } 1 \leq j \leq \alpha \text{ so that } \mathbf{X}_{y_j} \text{ depends on all of } \mathbf{X}_{u_1}, \ldots, \mathbf{X}_{u_\alpha} \ ,$$

since all size of view of all the vertices is strictly less than α. So in this case we reveal $(\pi_{r_e}(\mathbf{X}_{y_1}), \ldots, \pi_{r_e}(\mathbf{X}_{y_\alpha}))$. Since none of the $\pi_{r_e}(\mathbf{X}_{y_j})$ can depend on all the honest vertices, it is also a NOF transcript of length $\alpha \cdot r_e$. We do this fixing for all $e > i_0$ which implies that in total we reveal NOF transcript of length $\sum_{e=i_0+1}^{L} \alpha \cdot r_e$.

Combining case (i) and (ii) we see that we reveal NOF transcript of length $(i_0-1)m + \sum_{e=i_0+1}^{L} \alpha \cdot r_e$ which fixes

$$\bigoplus_{\substack{b_i : i \neq i_0 \\ b_i = (v_1, \ldots, v_\alpha)}} T_{r_i,m}(\pi_{r_i}(\mathbf{X}_{v_1}), \ldots, \pi_{r_i}(\mathbf{X}_{v_\alpha})) \ .$$

Recall that, for $1 \leq i \leq L$, $r_i = \rho^{i-1} n$ and $m = c \cdot \beta r_L$ where $\rho = c/(c + N2^{N+2})$ and $\beta = 1/(2^{N+2} N^N)$ for c being the constant from Proposition 2.

Claim 4. *For any $i_0 \in [L]$ where $L = \binom{|S|}{\alpha}$ we have $(i_0 - 1)m + \sum_{e=i_0+1}^{L} \alpha \cdot r_e \leq c r_{i_0}/2^N$.*

From Claim 4 we have total transcript length is at most $cr_{i_0}/2^N$. Since, $T_{r_{i_0},m}$ is a leakage resilient extractor against NOF protocols of transcript size at most $cr_{i_0}/2^N$ and error $\varepsilon \leq 2^{-cr_{i_0}/2^N}$, XORing the outputs of all $T_{r_i,m}$, by data processing inequality (Lemma 3) we get the desired result. □

Proof of Claim 4. Notice that $L \leq N^N$ since $|S|, \alpha \leq N$. Then clearly $i_0 \leq N^N$. Hence,

$$(i_0 - 1)m \leq cN^N \beta r_L \leq cr_{i_0}/2^{N+2} .$$

Above the second inequality holds because of the fact that $r_L \leq r_{i_0}$ and $\beta = 1/(2^{N+2}N^N)$. To bound the remaining term in the summand, notice that,

$$\sum_{e=i_0+1}^{L} \alpha r_e \leq N \sum_{i=i_0}^{L-1} \rho^i n \leq N(\rho^{i_0-1} n)\frac{\rho}{1-\rho} \leq Ncr_{i_0}(1/N2^{N+2}) = cr_{i_0}/2^{N+2}$$

Here the first inequality holds since $\alpha \leq N$ and the second inequality holds from infinite GP sum, since since $0 < \rho < 1$. Finally plugging the value of ρ we obtain the above bound. □

5.3 Non-explicit extractors when honest nodes have poly-logarithmic entropy

In the previous section we gave an explicit construction SHEDAG extractor when honest nodes are uniform and number of corruption is at most $\mathrm{res}_{\mathrm{uni}}(G)$ and honest nodes are uniform. Now we will give a non-explicit construction when honest nodes have polylogarithmic entropy. Before starting let us recall the definition of resilience in context of entropic honest nodes.

Definition 22 (Resilience of a graph in context of entropic honest vertices)**.** *Given a directed acyclic graph $G = (V, E)$ with $|V| \geq 2$ and a non-empty subset $S \subseteq V$ we defined resilience of a set S as*

$$\mathrm{res}_{\mathrm{ent}}(S) := |S| - \max\{2, d(S)\} ,$$

where $d(S) = \max_{s \in S} |\,\mathrm{view}(s) \cap S| + 1$. Further we define resilience of G as $\mathrm{res}_{\mathrm{ent}}(G) := \max_{\emptyset \neq S \subseteq V} \mathrm{res}_{\mathrm{ent}}(S)$.

Theorem 12. *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, with $N \geq 2$. 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 $t \leq \mathrm{res}_{\mathrm{ent}}(G)$.*

Proof of Theorem 12. Without loss of generality we can assume that $\mathrm{res}_{\mathrm{ent}}(G) \geq 1$ because otherwise number of corrupted blocks is 0 and we can apply standard independent source extractor in this case. By definition of resilience, this implies there exists non-empty $S \subseteq V$ such that $\mathrm{res}_{\mathrm{ent}}(S) \geq \mathrm{res}_{\mathrm{ent}}(G) \geq 1$. In Section 5.4.1 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 17. Now let, $\mathbf{X} \sim \{0,1\}^{nN}$ be any (n, k, G, t)-SHEDAG source with $t \leq \mathrm{res}_{\mathrm{ent}}(G)$. Notice that $\mathrm{res}_{\mathrm{ent}}(G) = \mathrm{res}_{\mathrm{ent}}(S)$. We define the set

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

Say, $\alpha := \max_{s \in S} |\,\mathrm{view}(s) \cap S| + 1$. Now there are two cases:

(i) **Case I: $\alpha \geq 2$.** Observe that from definition of resilience (Definition 22), in this case we have $\mathrm{res}_{\mathrm{ent}}(S) = |S| - \alpha$. Since at most $\mathrm{res}_{\mathrm{ent}}(S)$ many vertices are corrupted, from here it follows that at least $N - |S| - \alpha$ many vertices in G^S are honest, which further can be lower bounded by α.

(ii) **Case II:** $\alpha < 2$. In this case we have $\mathrm{res}_{\mathrm{ent}}(S) = |S| - 2$. Therefore we know in this case at least two vertices in G^S are honest.

Define $\beta := \max\{2, \alpha\}$. From the above discussion we have the following observation.

Observation 4. $|A_{\mathbf{X}}| \geq \beta$.

Before stating the extractor construction we need the following claim.

Claim 5. *There does not exist any $v \in S$ so that $A_{\mathbf{X}} \subseteq \mathrm{view}(v)$.*

Proof. We have $|A_{\mathbf{X}}| \geq \beta$ where $\beta = \max\{2, \alpha\}$. When $\alpha = 1$, we have $|\mathrm{view}(s) \cap S| = 0$ for all $s \in S$. That means G^S is empty (edgeless) graph. In this case $\beta = 2$ and since G^S is empty, there is no path between any two vertices in S. Hence in this case our claim is true.

When $\alpha \geq 2$ that implies $\beta = \max_{s \in S} |\mathrm{view}(s) \cap S| + 1$ and we know $|A_{\mathbf{X}}| \geq \beta$. Since number of honest nodes in S is strictly more than $|\mathrm{view}(s) \cap S|$ for all $s \in S$, in this case also our claim is true. □

We define our extractor in the following way,

$$\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) := \bigoplus_{\substack{B \subseteq S: \\ |B| = \beta \\ B = \{i_1 < \cdots < i_\beta\}}} \beta\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\beta} \| \mathsf{bit}(i_\beta)) \,, \tag{11}$$

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

Recall that $A_{\mathbf{X}} = \{i \in S : \mathbf{X}_i \text{ is honest}\}$ is the set of honest vertices. From Observation 4 we have $|A_{\mathbf{X}}| \geq \beta$. We can assume $|A_{\mathbf{X}}| = \beta$ since if it is more than β we can fix all the other honest vertices while retaining only β many. From Claim 5 it follows that, $\mathbf{X}_j$ depends on at most $\beta - 1$ many honest blocks if $j \in S \setminus A_{\mathbf{X}}$. Now Equation (11) looks like, $\mathsf{shedagExt}(\mathbf{X}_1, \ldots, \mathbf{X}_N) = \alpha\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1^0} \| \mathsf{bit}(i_1^0), \ldots, \mathbf{X}_{i_\beta^0} \| \mathsf{bit}(i_\beta^0)) \oplus \mathbf{X}''$ where

$$\mathbf{X}'' = \bigoplus_{\substack{\tilde{B} \subseteq S \\ |\tilde{B}| = \beta \\ \tilde{B} \neq A_{\mathbf{X}} \\ \tilde{B} = \{i_1 < \cdots < i_\beta\}}} \beta\text{-}\mathsf{nmExt}(\mathbf{X}_{i_1} \| \mathsf{bit}(i_1), \ldots, \mathbf{X}_{i_\beta} \| \mathsf{bit}(i_\beta))$$

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

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

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

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

when $k > f(n, \beta, \ell, m, \varepsilon)$, 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 β, ℓ 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. □

5.4 Algorithm for locating subset with optimal resilience

Let $G = (V, E)$ be a DAG where $|V| = N$ and $N \geq 2$. At first we focus on finding a non-empty set $S \subseteq V$ such that $\mathrm{res}_{\mathrm{uni}}(S) = \mathrm{res}_{\mathrm{uni}}(G)$.

Definition 23 (Resilience)**.** *Given a graph $G = (V, E)$ with $|V| \geq 2$, we say that a subset of vertices $S \subseteq V$ has resilience ($\mathrm{res}_{\mathrm{uni}}$) 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.

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

where $G^S = H(S)$ is the graph induced by the set S defined in Definition 17. Since for all $v \in S$, if there exists $u \in \mathrm{parents}(v) \cap S$, then $\operatorname{view}(v) \cap S \subset \operatorname{view}(u) \cap S$.

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

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

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

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

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

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

Therefore, we can get the following bound:

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

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

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

Lemma 7. *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: $S_0 := \{h \in S : |\text{view}(h) \cap S| = \max_{p \in S}(|\text{view}(p) \cap S|)\}$. Then either S maximizes resilience, or $H \cap S_0 = \emptyset$.*

Proof. If S maximizes resilience, lemma is proven. Otherwise, assume $\exists v \in H \cap S_0$. Then by the Definition 24, 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 6 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 6 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.

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 6.

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 7, 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

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*

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.

5.4.1 Algorithm to find a non-empty subset $S \subseteq V$ that maximizes $\mathrm{res}_{\mathrm{ent}}$.

Let us at first recall the definition of $\mathrm{res}_{\mathrm{ent}}$.

Definition 25 (Resilience of a graph in context of entropic honest vertices)**.** *Given a directed acyclic graph $G = (V, E)$ with $|V| \geq 2$ and a non-empty subset $S \subseteq V$ we defined resilience of a set S as*

$$\mathrm{res}_{\mathrm{ent}}(S) := |S| - \max\{2, d(S)\}\ ,$$

where $d(S) = \max_{s \in S} |\operatorname{view}(s) \cap S| + 1$. Further we define resilience of G as $\mathrm{res}_{\mathrm{ent}}(G) := \max_{\emptyset \neq S \subseteq V} \mathrm{res}_{\mathrm{ent}}(S)$.

Next we will show that by the similar algorithm as finding a set that maximizes $\mathrm{res}_{\mathrm{uni}}$, with a slight modification we can find a set that gives a set with optimal $\mathrm{res}_{\mathrm{ent}}$.

For each vertex $v \in V$, define the weight of the vertex as,

$$w(v) := \max\{2, |\operatorname{view}(v)| + 1\}.$$

Then for every non-empty subset S we have $\mathrm{res}_{\mathrm{ent}}(S) = |S| - \max_{s \in S} w(s)$. Let $W := \{w(v) : v \in V\}$ be the set of weights of the vertices. For each $q \in W$, define the threshold set

$$S_q := \{v \in V : w(v) \leq q\}.$$

Since $q \in W$, there is some vertex in S_q has weight exactly q. Consequently,

$$\mathrm{res}_{\mathrm{ent}}(S_q) = |S_q| - q \ .$$

Now we prove the following claim.

Lemma 8. *For every nonempty set $S \subseteq V$ that achieves $\mathrm{res}_{\mathrm{ent}}(S) = \mathrm{res}_{\mathrm{ent}}(G)$, we have $S = S_q$ for some $q \in W$.*

Proof sketch. Let $S \subseteq V$ be nonempty, and let $q_0 := \max_{s \in S} w(s)$. Then $q_0 \in W$ and $S \subseteq S_{q_0}$. Hence

$$\mathrm{res}_{\mathrm{ent}}(S) = |S| - q_0 \leq |S_{q_0}| - q_0 = \mathrm{res}_{\mathrm{ent}}(S_{q_0}).$$

Moreover, this inequality is strict whenever $S \subsetneq S_{q_0}$. Therefore, if S maximizes res′, then $S = S_{q_0}$. □

From the above lemma, follows that

$$\max_{\varnothing \neq S \subseteq V} \mathrm{res}_{\mathrm{ent}}(S) = \max_{q \in W} \big(|S_q| - q\big)$$

Now we follow the following algorithm First, compute $\mathrm{view}(v)$ for each vertex v and assign the weight $w(v) = \max\{2, |\mathrm{view}(v)| + 1\}$. Group vertices by their weights and process the distinct weights in increasing order. For each threshold q, maintain the number of vertices in $S_q = \{v \in V : w(v) \leq q\}$. Evaluate $\mathrm{res}_{\mathrm{ent}}(S_q) = |S_q| - q$ and record a threshold q^* maximizing this value. Return S_{q^*}, which is optimal by Lemma 8. The algorithm runs in $O(n(n+m))$ time, where $n = |V|$ and $m = |E|$.

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.

[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.

[CG25] Eshan Chattopadhyay and Jesse Goodman. Leakage-resilient extractors against number-on-forehead protocols. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 604–614, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718272.

[CGG+20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, Ashutosh Kumar, Xin Li, Raghu Meka, and David Zuckerman. Extractors and secret sharing against bounded collusion protocols. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1226–1242, 2020. doi:10.1109/FOCS46700.2020.00117.

[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.

[CGRS26] Eshan Chattopadhyay, Mohit Gurumukhani, Noam Ringach, and Rocco A. Servedio. Condensing and Extracting Against Online Adversaries. In Dana Moshkovitz, editor, *41st Computational Complexity Conference (CCC 2026)*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:22, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CCC.2026.5, doi:10.4230/LIPIcs.CCC.2026.5.

[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.

[FG13] Jeff Ford and Anna Gál. Hadamard tensors and lower bounds on multiparty communication complexity. 22(3):595–622, September 2013. doi:10.1007/s00037-012-0052-6.

[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.