Cryptanalysis of Goldreich's PRGs with MAJ–XOR Predicates

Tianning Wang[1], Haoyang Wang[1], Fukang Liu[2] and Willi Meier[3]

[1] School of Computer Science, Shanghai Jiao Tong University, Shanghai, China
[2] Institute of Science Tokyo, Tokyo, Japan
[3] University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland

Abstract

Goldreich's pseudorandom generators (PRGs) expand a secret n-bit seed into m output bits by applying the same local predicate to each selected subset of seed bits. We develop a pooled recovery framework for MAJ–XOR predicates. The method guesses that a set of seed bits has the same value and pools outputs whose majority inputs sufficiently overlap this set. Under a correct guess, Bias-Amplified Pooled Recovery (BAPR) exploits the amplified bias of the resulting noisy XOR equations. When the guessed bits force the selected majority outputs to a constant, its noiseless special case, Deterministic Pooled Recovery (DPR), uses exact equations. For $m = n^s$ with fixed $s > 1$, majority locality $a \in \omega(1) \cap O(\log n)$, and fixed XOR locality $b \geq 3$, we establish a worst-case $2^{o(n)}$-time recovery bound for BAPR when the input sets are sampled uniformly and independently across outputs. Under the stated cost models, our concrete recovery estimates for balanced seeds fall below the claimed security levels for the $\mathrm{MAJ}_7 \oplus \mathrm{XOR}_4$ challenge and selected $\mathrm{THR}_{64,32} \oplus \mathrm{XOR}_{10}$ instances collected by Boura et al. (ToSC 2025), as well as selected challenge instances proposed by Fu et al. (EUROCRYPT 2026). We then generalize DPR to fixed predicates satisfying explicit affine-restriction conditions and prove subexponential recovery without an additional rank assumption. Our analysis of alternative predicates highlights the potential use of exact or approximate affine relations in attacks, suggesting that such structure should also be considered in predicate design.

Keywords: Goldreich's PRGs, random local functions, MAJ–XOR predicates, seed recovery

1 Introduction

Pseudorandom generators (PRGs) are a fundamental cryptographic primitive that expands a short random seed into a longer string indistinguishable from uniform. Goldreich proposed a family of candidate one-way functions in which each output bit is obtained by applying a fixed predicate to a small, publicly specified subset of the input bits [Gol00]. When more output bits than seed bits are generated, this framework yields highly parallel candidate PRGs. Their security depends delicately on the choice of predicate, its locality, and the stretch.

Goldreich's construction also has a natural interpretation as a weak pseudorandom function (PRF). Concretely, the seed can be viewed as a weak PRF key, with each random tuple of input positions serving as an input and the corresponding generator bit as the output. Applebaum and Raykov connected expander-based local functions with weak

Emails: tianning@sjtu.edu.cn (Tianning Wang), haoyang.wang@sjtu.edu.cn (Haoyang Wang), liufukangs@gmail.com (Fukang Liu), willimeier48@gmail.com (Willi Meier)

Date of this document: 2026-11-11.

PRFs of very low circuit complexity [AR16]. As highlighted by Boura et al., high-stretch constructions based on $\mathrm{MAJ}_a \oplus \mathrm{XOR}_b$ with growing a form an important class of shallow weak PRF candidates [BCPR25]. Their low evaluation complexity is attractive in secure computation, including applications to oblivious PRFs, pseudorandom correlation functions for two-party protocols, and zero-knowledge proofs [BCPR25]. Concrete uses include silent OT and QuietOT instantiated with weak PRFs based on Goldreich's construction [BCM+24, CDD+24], as well as low-bandwidth mixed arithmetic in VOLE-based zero-knowledge proofs using a Goldreich-style $\mathrm{MAJ}_7 \oplus \mathrm{XOR}_4$ generator [ABBS25].

The MAJ–XOR family combines a linear XOR component with a nonlinear majority component and can attain high resiliency and algebraic immunity [BCPR25, Méa21, Méa22, DMR23]. Closely related structures appear in the FLIP and FiLIP stream ciphers, which were designed for efficient homomorphic evaluation [MJSC16, MCJS19]. Consequently, understanding the concrete security of these predicates is relevant both to Goldreich's construction and to the protocols and symmetric primitives built around the same low-complexity structure.

The security of Goldreich's construction has consequently been studied from both asymptotic and concrete perspectives. Early work showed that locality three is vulnerable to linear tests at sufficient stretch and that no predicate of locality at most four yields a local small-bias generator with arbitrary polynomial stretch [CM01, MST06]. Subsequent work obtained longer stretch from random local one-way functions and clarified the predicate-dependent behavior of local small-bias generators [App12, ABR12]. General attacks exploit correlations with a few input variables, low rational degree, or bounded degree and locality [BQ12, AL18, Üna23]. Couteau et al. [CDM+18] developed guess-and-determine attacks, including the constant-value guessing method of Section 5.3 that underlies our DPR analysis for MAJ–XOR. Yang et al. [YGJL22] substantially improved this line of attack, introduced a guess-and-decode method for sparse predicates, and also considered extensions to MAJ–XOR predicates.

More recently, several attacks have been developed specifically for MAJ–XOR predicates. Fu et al. [FLLL26] introduced the group-and-solve (GAS) attack. It obtains noisy linear systems by grouping outputs around a common subset and gives strong seed-recovery attacks in several parameter regimes. D'Antona et al. [DMÜ26] introduced the common-bias attack and obtained strong average-seed inversion results. GAS loses equations as the common subset grows, whereas the common-bias attack is especially effective for large majority locality or high stretch. D'Antona et al. leave open whether $\mathrm{MAJ}_a \oplus \mathrm{XOR}_b$ PRGs can retain exponential security in the intermediate regime $a \in \omega(1) \cap O(\log n)$. In parallel and independently, Lyu et al. [LLMF26] develop the Guess-Filter-Solve (GFS) attack using the same overlap-based pooling principle and introduce the Filter-Guess-Propagate (FGP) attack based on belief propagation (BP). Our analysis establishes subexponential recovery guarantees in the intermediate-locality regime and examines the rank and error behavior of the selected systems.

Our contributions. We analyze pooled recovery within this guess-and-determine framework. Our main result concerns *Bias-Amplified Pooled Recovery* (BAPR), which guesses that a set of seed bits has the same value and pools outputs whose majority inputs sufficiently overlap this set. Under a correct guess, this yields noisy XOR equations with amplified bias. For $m = n^s$ with fixed $s > 1$, majority locality $a \in \omega(1) \cap O(\log n)$, and fixed XOR locality $b \geq 3$, we give a worst-case $2^{o(n)}$-time recovery algorithm. Recovery succeeds with high probability for every fixed seed under the independent uniform input model. This rules out exponential recovery security in the stated regime, answering the question above for fixed XOR locality and polynomial stretch under the stated input model. We explicitly analyze the rank of the selected systems: the XOR equations are sparse and selection changes their coefficient distribution, so having enough equations does not ensure

full rank. We derive this distribution for both BAPR and its noiseless special case, using it to bound rank deficiency and the number of seed candidates.

In the noiseless case, *Deterministic Pooled Recovery* (DPR) uses constant-value guesses to obtain exact affine equations. For a fixed predicate, suppose that, for each constant $c \in \{0,1\}$, fixing r_{aff} suitable local inputs to c leaves a nonconstant affine function. For fixed $1 < s < r_{\text{aff}} + 1$, we prove worst-case subexponential-time recovery by Gaussian elimination without an additional rank assumption. Recovery succeeds with high probability for every fixed seed in the same random input model. Motivated by Goldreich's original expander-graph proposal, we also extend the recovery guarantee for fixed MAJ–XOR predicates to a random layered matching model that allows dependencies between the input sets of different outputs.

Our fixed-weight analysis shows how unbalanced seeds can dominate the average common bias. Under the common-bias inversion-cost model, recovery estimates can be much higher at balanced seed weights. This motivates evaluating recovery at balanced or nearest-balanced weights. Under the stated cost models with linear-algebra exponent $\omega = 3$, BAPR yields estimated recovery costs of $2^{65.4}$ and $2^{105.2}$ for $\text{THR}_{64,32} \oplus \text{XOR}_{10}$ at $(n, s) = (512, 4.5)$ and $(894, 3.4)$, respectively. DPR gives an estimated cost of $2^{114.51}$ for $\text{MAJ}_7 \oplus \text{XOR}_4$ at $(n, s) = (1024, 2)$. These results illustrate the effectiveness of pooled recovery on concrete challenge instances.

To assess alternative predicates, we apply DPR to selected instances of the constructions in [Sar25]. We establish a direct-recovery guarantee for these instances. Under an explicit recovery-rank condition, we also compare ordinary DPR with a quadratic variant. For the predicate family $f_{d,\epsilon,a} \oplus \text{XOR}_b$ proposed by D'Antona et al. [DMÜ26], we derive conditional recovery-cost estimates from an affine approximation. These applications suggest evaluating the cost of exposing and solving affine or low-degree relations alongside the usual predicate criteria.

Organization. Section 2 introduces the notation and background on Goldreich's construction. Section 3 revisits common bias at fixed seed weights, providing the motivation for our recovery approach. Building on this analysis, Section 4 develops the pooled recovery framework and studies its concrete costs and asymptotic guarantees. Section 5 explores the broader applicability of deterministic pooled recovery to other predicates and structured input models. Section 6 examines an affine approximation for $f_{d,\epsilon,a} \oplus \text{XOR}_b$ and its implications for predicate design. Section 7 concludes this paper.

2 Preliminaries

2.1 Notation

For a positive integer n, let $[n] = \{1, \ldots, n\}$. If $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ and $S \subseteq [n]$, then $x_S = (x_i)_{i \in S}$ denotes the restriction of x to the coordinates indexed by S, in increasing order. If σ is an ordered tuple of distinct elements of $[n]$, then x_σ denotes the corresponding coordinates of x in the order specified by σ. The Hamming weight of x is $\text{wt}(x) = |\{i : x_i = 1\}|$. For an integer r, $\binom{[n]}{r}$ denotes the set of r-subsets of $[n]$. We use the convention $\binom{r}{u} = 0$ whenever $u < 0$ or $u > r$. For a positive integer r, write $\mathbf{1}_r$ for the all-one vector of length r. We write $\oplus$ for XOR and $\bigoplus$ for XOR over a collection of bits.

We use $O(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$, $o(\cdot)$, and $\omega(\cdot)$ in their standard asymptotic senses. The standalone symbol ω denotes the exponent of linear algebra. log and $\log_2$ denote natural and base-two logarithms, respectively, and exp denotes the natural exponential function.

2.2 Probabilistic and Combinatorial Preliminaries

We recall the probability distributions and bounds used below.

Binomial distribution. We write $X \sim \mathrm{Bin}(M, p)$ if X has the binomial distribution with M trials and success probability p. For $0 \le j \le M$,

$$\Pr[X = j] = \binom{M}{j} p^j (1-p)^{M-j}, \qquad \mathbb{E}[X] = Mp.$$

Hypergeometric distribution. We write $X \sim \mathrm{Hypergeom}(N, w, a)$ if X has the hypergeometric distribution with population size N, w marked elements, and sample size a, drawn without replacement. For $0 \le j \le a$,

$$\Pr[X = j] = \frac{\binom{w}{j}\binom{N-w}{a-j}}{\binom{N}{a}}, \qquad \mathbb{E}[X] = \frac{aw}{N}.$$

Markov's inequality [MU17]. This inequality bounds the probability that a nonnegative random variable is at least a given threshold in terms of its expectation. For the variable X with finite expectation and any $u > 0$,

$$\Pr[X \ge u] \le \frac{\mathbb{E}[X]}{u}. \tag{1}$$

The same bound holds under a conditional distribution, using its conditional expectation.

Chernoff lower-tail bound [Che52, MU17]. This bound controls the probability that the number of successes in independent trials falls substantially below its expectation. Let X be a sum of independent $\{0,1\}$-valued random variables with mean $\mu > 0$. For $0 < \delta < 1$,

$$\Pr[X \le (1-\delta)\mu] \le \exp\left(-\frac{\delta^2 \mu}{2}\right). \tag{2}$$

In particular, $\Pr[X \le \mu/2] \le \exp(-\mu/8)$.

Hypergeometric tail bounds [Hoe63]. These bounds control deviations from the expected number of marked elements in sampling without replacement. For $X \sim \mathrm{Hypergeom}(N, w, a)$ with $a \ge 1$, let $\mu = aw/N$. For every $u \ge 0$,

$$\Pr[X \le \mu - u] \le \exp\left(-\frac{2u^2}{a}\right), \qquad \Pr[X \ge \mu + u] \le \exp\left(-\frac{2u^2}{a}\right). \tag{3}$$

Binomial coefficient bounds [Ver26]. These bounds estimate the number of subsets of a given size or of size at most a prescribed value. For integers $1 \le r \le N$,

$$\left(\frac{N}{r}\right)^r \le \binom{N}{r} \le \sum_{j=0}^{r} \binom{N}{j} \le \left(\frac{\exp(1)N}{r}\right)^r. \tag{4}$$

If $r = o(N)$, the upper bound is $2^{o(N)}$, since $r \log_2(\exp(1)N/r) = o(N)$.

2.3 Weak pseudorandom functions

A PRF is a keyed family of functions that is computationally hard to distinguish from a uniformly random function. A weak PRF asks for the same indistinguishability property only on random inputs. Thus the difference between a weak PRF and a PRF lies in the query model: for a weak PRF, the adversary receives evaluations at uniformly sampled inputs, whereas for a standard PRF, the adversary may choose its inputs adaptively.

Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function. For a key $\mathsf{sk} \in \mathcal{K}$, write $f_{\mathsf{sk}}(x) = f(\mathsf{sk}, x)$. The weak PRF advantage of an adversary $\mathcal{A}$ receiving Q random input–output samples is

$$\mathrm{Adv}^{\mathrm{wPRF}}_{f,\mathcal{A}} = \left|\Pr\left[\mathcal{A}\big((\xi_i, f_{\mathsf{sk}}(\xi_i))_{i=1}^{Q}\big) = 1\right] - \Pr\left[\mathcal{A}\big((\xi_i, g(\xi_i))_{i=1}^{Q}\big) = 1\right]\right|,$$

where $\mathsf{sk} \leftarrow \mathcal{K}$ and $\xi_1, \ldots, \xi_Q \leftarrow \mathcal{X}$ are sampled uniformly and independently, $g : \mathcal{X} \to \mathcal{Y}$ is a uniformly random function, and the probabilities also include the internal randomness of $\mathcal{A}$.

2.4 Goldreich's PRGs and Local Predicates

Goldreich's PRGs apply the same local predicate to the seed bits indexed by each tuple in a public sequence. Fix positive integers n, m, a Boolean predicate P, and ordered tuples $\sigma_1, \ldots, \sigma_m$, each consisting of distinct elements of $[n]$ and having the arity of P. The resulting generator is

$$G_{\Gamma,P}(x) = \big(P(x_{\sigma_1}), P(x_{\sigma_2}), \ldots, P(x_{\sigma_m})\big),$$

where $\Gamma = (\sigma_1, \ldots, \sigma_m)$ is public. Unless stated otherwise, for each output, the seed positions are sampled uniformly without replacement in the order of the predicate inputs. Different outputs are sampled independently. In the polynomial-stretch regime, we write $m = n^s$, up to rounding, where $s > 1$ is the stretch exponent.

The same construction also admits a natural weak PRF interpretation: the seed x serves as the secret key, while a random ordered tuple of input positions serves as the input. Namely, for an ordered tuple σ of the appropriate length, define

$$f_x(\sigma) = P(x_\sigma).$$

Thus, independently sampled random queries to f_x have the same distribution as the coordinates of $G_{\Gamma,P}(x)$ when the tuples in Γ are sampled independently. This gives a single-output weak PRF formulation of the same local construction. Applebaum and Raykov [AR16] study a closely related batched construction in which each weak PRF input encodes a hypergraph and the output contains one predicate evaluation for each of its hyperedges. They establish weak PRF security under suitable expander-based one-wayness assumptions and parameter settings.

The choice of predicate is the main security-relevant ingredient in Goldreich's PRGs. Its arity controls both evaluation efficiency and resistance to known attacks.

The main class considered in this work is the MAJ–XOR family. We describe it using threshold functions, with majority as a special case. For $z \in \{0,1\}^a$, define the threshold function

$$\mathrm{THR}_{a,t}(z) = \begin{cases} 1, & \mathrm{wt}(z) \geq t, \\ 0, & \mathrm{wt}(z) < t. \end{cases}$$

The majority function uses a threshold of half the input length, rounded up:

$$\mathrm{MAJ}_a(z) = \mathrm{THR}_{a,\lceil a/2 \rceil}(z).$$

For odd a, this threshold is $(a+1)/2$. For $u \in \{0,1\}^b$, define

$$\mathrm{XOR}_b(u) = u_1 \oplus \cdots \oplus u_b.$$

We consider predicates obtained by XORing a threshold function and a parity function on disjoint input sets:

$$P_{a,t,b}(x_A, x_B) = \mathrm{THR}_{a,t}(x_A) \oplus \mathrm{XOR}_b(x_B), \qquad |A| = a,\ |B| = b,\ A \cap B = \varnothing.$$

The MAJ–XOR predicate is the special case $t = \lceil a/2 \rceil$.

For each output position i, we sample A_i uniformly from $\binom{[n]}{a}$. We then sample B_i uniformly from $\binom{[n]\setminus A_i}{b}$. The pairs (A_i, B_i) are sampled independently across outputs. The observed output bit is

$$y_i = \mathrm{THR}_{a,t}(x_{A_i}) \oplus \mathrm{XOR}_b(x_{B_i}). \tag{5}$$

These predicates have very low evaluation complexity and can be chosen to satisfy classical design criteria such as resiliency and algebraic immunity [Méa21, Méa22, DMR23]. The concrete values of these properties depend on the parameters.

3 A Fixed-Weight Refinement of Common Bias

D'Antona et al. [DMÜ26] introduced the common-bias inversion attack, which exploits agreement among threshold outputs in Goldreich generators with MAJ–XOR predicates. Their average-seed estimates, however, do not by themselves show how performance varies with seed weight. We examine this dependence explicitly by evaluating the attack at fixed seed weights under the same concrete cost model.

Let $x \in \{0,1\}^n$ be the seed. The concrete inverter uses the first n output equations and guesses that their threshold values all equal a constant $c \in \{0,1\}$. When this guess is correct,

$$y_i \oplus c = \bigoplus_{j \in B_i} x_j, \qquad i \in [n].$$

Writing M for the XOR coefficient matrix, the algorithm solves $Mx = (y_1, \ldots, y_n)^{\mathsf{T}} \oplus c\mathbf{1}_n$ and returns a uniformly chosen solution, or fails if the system is inconsistent. Under a correct threshold guess, the true seed belongs to the solution space.

For a uniformly sampled seed, define the probability of agreement with the better fixed constant by

$$\text{c-bias}^n_{\text{const}}(\mathrm{THR}_{a,t}) = \max_{c \in \{0,1\}} \Pr[\mathrm{THR}_{a,t}(x_{A_1}) = \cdots = \mathrm{THR}_{a,t}(x_{A_n}) = c]\,. \tag{6}$$

The probability is over the seed and the independent uniform threshold input sets. Based on rank experiments, the concrete model of [DMÜ26] uses a conservative factor of $1/4$ for the event $\mathrm{rank}(M) \geq n - 1$. When the threshold guess is correct and this rank condition holds, there are at most two solutions, so choosing one uniformly recovers the true seed with probability at least $1/2$. The estimated success probability for one instance is therefore

$$\mathrm{Adv} = \text{c-bias}^n_{\text{const}} \cdot \frac{1}{4} \cdot \frac{1}{2} = \frac{\text{c-bias}^n_{\text{const}}}{8}.$$

With Gaussian-elimination cost $T = n^3$ per instance, the reported $T/\,\mathrm{Adv}$ estimates the cost per successful inversion in the multi-instance setting.

To expose the seed-weight dependence, for $0 \leq h \leq n$ define

$$q_{a,t}(n,h) = \sum_{u=t}^{a} \frac{\binom{h}{u}\binom{n-h}{a-u}}{\binom{n}{a}}. \tag{7}$$

This is the probability that one threshold output is 1 for a seed of Hamming weight h. The average common bias is obtained by averaging the agreement probability over seed

weights for each fixed constant and then taking the larger of the two values. Although unbalanced seeds are less frequent, their much higher threshold-agreement probabilities can dominate the average.

We now condition on $\mathrm{wt}(x) = w$. For each weight, we evaluate the better of the two constant guesses, whose agreement probability is

$$C_w = \max\left\{q_{a,t}(n,w)^n,\ (1-q_{a,t}(n,w))^n\right\}.$$

Assuming that the same empirical rank factor applies at each seed weight, the estimated cost per successful inversion is

$$T_{\mathrm{CB}}(w) = \frac{8n^3}{C_w}. \tag{8}$$

Table 1 compares the reported average-seed estimates with the fixed-weight estimates for $\mathrm{THR}_{64,32} \oplus \mathrm{XOR}_{10}$. The weight ranges consist of those w for which $T_{\mathrm{CB}}(w) < 2^{128}$, and the seed fraction is their total probability under a uniformly sampled n-bit seed.

Table 1: Common-bias inversion estimates using n output equations. The two cost columns give base-two logarithms. The average-seed values are from [DMÜ26, Table 1].

n	average cost	balanced cost	weights with cost below 2^{128}	seed fraction
256	41.94	242.90	$w \le 116$ or $w \ge 136$	$2^{-2.00}$
512	62.17	467.45	$w \le 217$ or $w \ge 287$	$2^{-8.04}$
894	90.22	799.85	$w \le 362$ or $w \ge 519$	$2^{-20.20}$
2048	171.21	1799.88	$w \le 776$ or $w \ge 1241$	$2^{-71.14}$

In fact, for $n = 256$ and $\mathrm{THR}_{64,32}$, seeds with weights $152 \le h \le 168$ account for only about 0.16% of all seeds but contribute about 99% of the average common bias for $c = 1$. Under this model, common-bias inversion has a substantially higher estimated cost for balanced seeds. The average-seed results therefore do not imply comparable performance across all seed weights. This motivates recovery methods that exploit biased equations without requiring all threshold outputs to agree.

Remark 1. Appendix D of D'Antona et al. [DMÜ26] extends the common-bias inversion attack to uniformly sampled balanced seeds. It selects outputs whose threshold inputs lie in a fixed subset of $n' < n$ coordinates and averages over the hypergeometric seed weight within this subset. Restricting the inputs to a smaller subset lowers the selection probability, so sufficiently many outputs are needed to obtain n selected equations. The applicability of this variant is therefore constrained by the available output budget.

4 Pooled Recovery

Motivated by the fixed-weight analysis, we study pooled recovery by guessing that a set of seed bits has the same value and selecting outputs whose threshold inputs sufficiently overlap this set. A correct guess gives exact or biased affine relations, allowing recovery without requiring all selected threshold outputs to agree.

4.1 The Pooled Recovery Method

We first describe how to construct the pool of equations used for recovery. Choose a set $K \subseteq [n]$ of size k, independently of the public input sets, and guess $x_K = \mathbf{1}_k$. This leaves $N = n - k$ unguessed seed bits. For an integer $0 \le q \le \min(a, k)$, retain the output indices

$$I_{K,q} = \{i \in [m] : |A_i \cap K| \ge q\}, \qquad e = |I_{K,q}|,$$

where e is the number of retained equations. This pools every output meeting the overlap threshold, regardless of which q positions of K it contains.

Lemma 1 (Distribution of the selected-pool size)**.** *For any fixed $K \subseteq [n]$ with $|K| = k$, assume that the threshold input sets $A_1, \ldots, A_m$ are chosen independently and uniformly from $\binom{[n]}{a}$. Define the selection probability*

$$p_{\mathrm{sel}} = \sum_{j=q}^{\min(a,k)} \frac{\binom{k}{j}\binom{N}{a-j}}{\binom{n}{a}}. \tag{9}$$

Then

$$e \sim \mathrm{Bin}(m, p_{\mathrm{sel}}), \qquad \mu := \mathbb{E}[e] = m p_{\mathrm{sel}}. \tag{10}$$

Proof. For each i, counting according to $|A_i \cap K|$ shows that $\Pr[|A_i \cap K| \geq q] = p_{\mathrm{sel}}$. Since the sets A_i are sampled independently, the corresponding selection indicators are independent. Their sum is e, which gives the stated distribution and expectation. □

For a seed of weight w and $k \leq w$, a uniform guessed set is all-one with probability $\binom{w}{k}/\binom{n}{k}$. The expected number of independently sampled sets until such a guess is therefore

$$S_1(n, w, k) = \frac{\binom{n}{k}}{\binom{w}{k}}.$$

Even when $q < t$, a correct guess can bias the selected threshold outputs toward 1. We exploit this bias for seed recovery and call this approach *Bias-Amplified Pooled Recovery* (BAPR). We quantify this bias for general q, conditioning on a correct all-one guess. For fixed seed weight w, conditioning on $|A_i \cap K| = j$ leaves $a - j$ threshold positions outside K, among which $w - k$ of the N available seed coordinates are ones. Using eq. (7), the probability that the threshold output equals 1 is

$$\beta_j(w) = \begin{cases} q_{a-j,t-j}(N, w-k), & j < t, \\ 1, & j \geq t. \end{cases}$$

Conditioned on selection, the possible values of j have weights proportional to $\binom{k}{j}\binom{N}{a-j}$. Thus the effective holding probability is

$$\beta_{k,q}(w) = \frac{\sum_{j=q}^{\min(a,k)} \binom{k}{j}\binom{N}{a-j}\,\beta_j(w)}{\sum_{j=q}^{\min(a,k)} \binom{k}{j}\binom{N}{a-j}}.$$

When $\beta_{k,q}(w) > 1/2$, predicting the selected threshold values as 1 gives useful biased XOR equations.

For a correct guess $x_K = \mathbf{1}_k$, substituting the guessed XOR variables and predicting the selected threshold value as 1 gives, for each $i \in I_{K,q}$, the noisy equation

$$\bigoplus_{v \in B_i \setminus K} x_v = y_i \oplus 1 \oplus (|B_i \cap K| \bmod 2).$$

It is correct exactly when $\mathrm{THR}_{a,t}(x_{A_i}) = 1$, and hence with probability $\beta_{k,q}(w)$ conditioned on selection and on $\mathrm{wt}(x) = w$.

Algorithm 1 Deterministic Pooled Recovery for $\mathrm{THR}_{a,t} \oplus \mathrm{XOR}_b$.

Require: Public sets $(A_i, B_i)_{i=1}^m$, output $y \in \{0,1\}^m$, and parameters k, q with $q \geq t$, and a rank-deficiency bound δ.
Ensure: A seed candidate $x \in \{0,1\}^n$, or failure.
1: Choose a candidate set $K \subseteq [n]$ with $|K| = k$.
2: Guess $x_K = \mathbf{1}_k$.
3: Set $I_{K,q} = \{i : |A_i \cap K| \geq q\}$ and $e = |I_{K,q}|$.
4: **if** $e < N$ **then**
5: **return** failure.
6: **end if**
7: **for** each $i \in I_{K,q}$ **do**
8: Form the linear equation implied by the guess

$$\bigoplus_{v \in B_i \setminus K} x_v = y_i \oplus 1 \oplus (|B_i \cap K| \bmod 2).$$

9: **end for**
10: Apply Gaussian elimination to the resulting system over $\mathbb{F}_2$.
11: **if** the system is inconsistent or has rank less than $N - \delta$ **then**
12: **return** failure.
13: **end if**
14: Enumerate its at most 2^δ solutions and verify them against the full PRG output.
15: **return** the first candidate passing verification, or failure.

Deterministic Pooled Recovery. When $q \geq t$, a correct all-one guess forces every selected threshold output to 1, giving $\beta_{k,q}(w) = 1$. The resulting XOR equations are therefore noiseless. We call this special case *Deterministic Pooled Recovery* (DPR). Its direct implementation solves the entire selected system and verifies the resulting seed candidates, as summarized in Algorithm 1.

The all-zero branch instead guesses $x_K = \mathbf{0}_k$, predicts the selected threshold values as zero, and has search factor $\binom{n}{k}/\binom{n-w}{k}$. Conditioned on $|A_i \cap K| = j$, its holding probability is $1 - q_{a-j,t}(N, w)$; averaging over j uses the same selection weights as above. For balanced seeds, trying both values saves at most one bit in the guessing cost before accounting for the additional solving work. For simplicity, we use the all-one branch in the concrete estimates.

4.2 Rank Model

For the selected XOR subsystems, related work such as [LLMF26] estimates the solving cost under a full-rank assumption. In pooled recovery, however, the XOR equations are sparse, and selection changes their coefficient distribution after substituting the guessed bits. Some unknown coordinates may not appear in the selected equations, so having enough equations does not by itself guarantee full rank. We therefore analyze the selected matrix distribution and quantify its effect on the recovery cost.

Let M_K be the $e \times N$ coefficient matrix whose row for $i \in I_{K,q}$ is the incidence vector of $B_i \setminus K$ on the $N = n - k \geq 1$ unguessed coordinates. Write

$$d_K = N - \mathrm{rank}(M_K).$$

On a correct DPR branch the system is consistent and has exactly 2^{d_K} solutions. The coefficient distribution depends on the selection rule, irrespective of whether the equations are noisy. This rank analysis applies to both BAPR and DPR for all allowed values of q.

Selected-row distribution. Selecting rows with $|A_i \cap K| \geq q$ affects the XOR coefficients because B_i is sampled from $[n] \setminus A_i$. Given $|A_i \cap K| = j$, this set contains $k - j$ guessed coordinates and $N - (a - j)$ unknown coordinates, so

$$\Pr[|B_i \setminus K| = \ell \mid |A_i \cap K| = j] = \frac{\binom{N-(a-j)}{\ell}\binom{k-j}{b-\ell}}{\binom{n-a}{b}}, \qquad 0 \leq \ell \leq \min\{b, N\}. \tag{11}$$

Conditional on row weight ℓ, the unknown support is uniform among the ℓ-subsets of the N remaining coordinates, by symmetry. Let ν_ℓ be the probability that a retained equation contains exactly ℓ unknown XOR variables after substituting the guessed values, i.e., that its coefficient row has weight ℓ. Averaging over the overlap size with probabilities η_j gives

$$\eta_j := \Pr[|A_i \cap K| = j \mid i \in I_{K,q}] = \frac{\binom{k}{j}\binom{N}{a-j}}{\sum_{h=q}^{\min(a,k)} \binom{k}{h}\binom{N}{a-h}},$$
$$\nu_\ell := \Pr[|B_i \setminus K| = \ell \mid i \in I_{K,q}] = \sum_{j=q}^{\min(a,k)} \eta_j \frac{\binom{N-(a-j)}{\ell}\binom{k-j}{b-\ell}}{\binom{n-a}{b}}.$$

Conditional on their number $e = |I_{K,q}|$, the selected rows remain independent: the public pairs (A_i, B_i) are independent and selection tests each A_i separately.

If $bk/(n-a)$ is small, then a selected row has weight b with probability close to one, since $\Pr[B_i \cap K \neq \varnothing \mid i \in I_{K,q}] \leq bk/(n-a)$. In particular, this holds for fixed a, b and $k = o(n)$, making the fixed-row-weight model a useful reference. For fixed $b \geq 3$, Cooper–Frieze–Pegden [CFP19, Theorem 3] show that $e = \lceil C_b N \ln N \rceil$ uniformly sampled distinct weight-b rows have rank N for odd b and $N-1$ for even b, with probability $1 - o(1)$, for a sufficiently large constant C_b. Their theorem is stated for columns, so we apply it after transposing the matrix. The same rank guarantee holds for independent rows sampled with replacement. Among these e rows, the probability of any repeated support is at most $\binom{e}{2}/\binom{N}{b}$, which tends to zero for fixed $b \geq 3$. Conditional on no repetition, the supports are uniformly sampled distinct sets.

This asymptotic result does not quantify the enumeration cost at concrete parameters. We therefore analyze the full selected-row distribution to bound the number of candidate seeds and the resulting enumeration cost.

For an upper bound on d_K, let ϕ_z be the probability that one selected row has zero inner product over $\mathbb{F}_2$ with a fixed vector of weight z. This requires an even intersection of their supports, hence

$$\phi_z = \sum_{\ell=0}^{\min\{b,N\}} \nu_\ell \frac{\sum_{\substack{0 \leq h \leq \ell \\ h \text{ even}}} \binom{z}{h}\binom{N-z}{\ell-h}}{\binom{N}{\ell}}.$$

Lemma 2 (Bounding the rank deficiency)**.** *Let* $\Psi_e = \sum_{z=0}^{N} \binom{N}{z}\phi_z^e$. *Conditional on* $|I_{K,q}| = e$, *for every integer* $D \geq 0$,

$$\mathbb{E}[2^{d_K} \mid |I_{K,q}| = e] = \Psi_e, \qquad \Pr[d_K \geq D \mid |I_{K,q}| = e] \leq 2^{-D}\Psi_e.$$

Proof. Rank–nullity gives $|\ker(M_K)| = 2^{d_K}$. Conditional on the row count e, independence of the selected rows implies that each fixed vector of weight z lies in the kernel with probability ϕ_z^e. Summing these probabilities over the $\binom{N}{z}$ vectors of each weight gives $\mathbb{E}[2^{d_K} \mid |I_{K,q}| = e] = \Psi_e$. Markov's inequality in eq. (1), applied to 2^{d_K} at threshold 2^D, gives the tail bound. □

Lower bounds on rank deficiency. A zero column contains only zeros and corresponds to an unknown seed bit that does not appear in any selected equation. Such a bit remains undetermined by the selected system. If Z is the number of zero columns, their coordinate vectors are linearly independent elements of $\ker(M_K)$, so $d_K \geq Z$. By uniformity of the support conditional on its weight, a selected row contains each unknown coordinate with probability $\rho = N^{-1} \sum_{\ell=0}^{\min\{b,N\}} \ell \nu_\ell$. Independence of rows, followed by averaging over $e \sim \mathrm{Bin}(m, p_{\mathrm{sel}})$, gives

$$\mathbb{E}[Z \mid |I_{K,q}| = e] = N(1-\rho)^e, \tag{12}$$

$$\mathbb{E}[Z] = N(1-p_{\mathrm{sel}}\rho)^m. \tag{13}$$

Also, $\mathrm{rank}(M_K) \leq e$ implies $d_K \geq (N-e)_+$, where $(u)_+ = \max\{u, 0\}$. Thus

$$\mathbb{E}[d_K] \geq \max\{\mathbb{E}[(N-e)_+],\, N(1-p_{\mathrm{sel}}\rho)^m\}\,. \tag{14}$$

These bounds concern the full selected pool and apply regardless of whether its equations are noisy.

Random subsystems for BAPR. DPR solves the full selected system. For noisy BAPR equations, we use the Gauss and SubGauss solvers introduced by Fu et al. [FLLL26]. At the row-count thresholds used in our concrete estimates, both solvers sample N distinct equations uniformly from the selected pool. Write R for the sampled indices, $M_{K,R}$ for the submatrix of M_K formed by those rows, and $d_{K,R} = N - \mathrm{rank}(M_{K,R})$. An inconsistent system is discarded; otherwise, its affine solution space contains exactly $2^{d_{K,R}}$ candidates, which are enumerated and checked against the full output.

For each pool size $e \geq N$, averaging over the sampled pairs (A_i, B_i) and the random choice of R, the sampled equations have the distribution of N independent selected rows. Lemma 2 therefore gives

$$\mathbb{E}[2^{d_{K,R}} \mid e] = \Psi_N.$$

Since inconsistent subsystems yield no candidates, Ψ_N bounds the expected candidate count per sampled subsystem, including those formed under incorrect guesses. For these N-row subsystems, the zero-column lower bound is $\mathbb{E}[d_{K,R} \mid e] \geq N(1-\rho)^N$.

4.3 Recovery Cost Estimates

We discuss the direct solution of the noiseless DPR pool and the repeated subsystem trials for noisy BAPR equations separately.

Complexity analysis for DPR. For a correct guess $x_K = \mathbf{1}_k$ with $q \geq t$, the selected system is consistent and its affine solution space contains exactly 2^{d_K} candidates. Hence, if $d_K \leq \delta$, enumerating this space requires at most 2^δ candidate checks against the complete PRG output. This recovers the original seed whenever it is the unique candidate passing verification.

We next account for trials in which the guess on K is incorrect. Let C_{sel} denote the cost of locating the selected rows for one candidate K, let V denote the cost of verifying one seed candidate against the full PRG output, and let ω be the linear-algebra exponent. A direct scan of the m threshold input sets gives $C_{\mathrm{sel}} = O(ma)$, while elimination over the N remaining unknowns is charged $O(N^\omega)$ operations in the concrete estimates.

Incorrect guesses. For an incorrect guess, the assumed threshold values and the substituted guessed bits induce an error in the right-hand side. Consistency imposes $e - \mathrm{rank}(M_K)$ independent parity checks on this error. The following estimate uses a random-syndrome model for incorrect guesses: conditional on M_K, we model the outcomes

of these checks as uniformly distributed in $\mathbb{F}_2^{e-\mathrm{rank}(M_K)}$. Under this model, a false system is consistent with probability

$$2^{-(e-\mathrm{rank}(M_K))}.$$

Conditional on consistency, its affine solution space contains $2^{N-\mathrm{rank}(M_K)}$ vectors. Hence the expected number of candidates generated by a false guess, conditional on e and M_K, is

$$2^{-(e-\mathrm{rank}(M_K))}2^{N-\mathrm{rank}(M_K)} = 2^{N-e}.$$

In particular, this estimate is independent of the rank of M_K. We discard trials with $e < N$ before candidate enumeration. For every remaining false trial, $e \geq N$, so $2^{N-e} \leq 1$. Thus, within the random-syndrome model, the expected verification cost of a false trial is at most $O(V)$. The probability that a correct guess is discarded is included in the retry estimate below.

Not every correct guess produces a usable system. For an integer $\delta \geq 0$, write

$$p_{\mathrm{use}}(\delta) = \Pr[e \geq N,\ d_K \leq \delta \mid x_K = \mathbf{1}_k]$$

for the probability that a correct guess yields enough rows and rank deficiency at most δ. Unusable systems are discarded. Using an independent-trial approximation for the retry count, the estimated number of trials before obtaining a usable correct system is

$$\frac{S_1(n,w,k)}{p_{\mathrm{use}}(\delta)}.$$

This gives the total-cost estimate

$$O\left(\frac{S_1(n,w,k)}{p_{\mathrm{use}}(\delta)}\big(C_{\mathrm{sel}} + N^\omega + V\big) + 2^\delta V\right).$$

Here $V = O(m(a+b))$ for full verification. The corresponding bit-complexity exponent is

$$\max\big\{\log_2 S_1(n,w,k) - \log_2 p_{\mathrm{use}}(\delta) + \log_2\big(C_{\mathrm{sel}} + N^\omega + V\big), \delta + \log_2 V\big\} + O(1). \tag{15}$$

Within this model, verification in false trials is charged at its expected cost $O(V)$, while the successful trial contributes at most $2^\delta V$.

Complexity analysis for BAPR. We estimate the complexity of BAPR using the Gauss and SubGauss solvers under the cost assumptions of Fu et al. [FLLL26]. Both solvers provisionally treat the selected equations as correct, solve them by Gaussian elimination, enumerate any remaining free variables, and verify the resulting candidates against the full PRG output.

Consider a pool of e noisy equations in N unknowns, each with holding probability $\beta > 1/2$. Gauss makes one attempt using $\min\{e, N\}$ equations and moves to a new pool if recovery fails. SubGauss instead repeatedly samples subsystems from the same pool, each containing $h = \min\{\lfloor e\beta \rfloor, N\}$ equations, where $e\beta$ is the expected number of correct equations in the pool. Reusing the same pool reduces the cost of selecting equations. Under the full-row-rank assumption used in the underlying solver model, the Gauss estimate is

$$T_{\mathrm{G}}(N,e,\beta;m) = \beta^{-\min(e,N)} \begin{cases} m + N^\omega, & e \geq N, \\ m + 2^{N-e}e^\omega, & e < N. \end{cases}$$

The SubGauss estimate is

$$T_{\mathrm{SG}}(N,e,\beta;m) = \frac{1}{1-\exp(-1)} \begin{cases} m + \beta^{-N}N^\omega, & e \geq N/\beta, \\ m + 2^{N-e\beta}\beta^{-e\beta}(e\beta)^\omega, & e < N/\beta. \end{cases}$$

Algorithm 2 Bias-Amplified Pooled Recovery for $\mathrm{THR}_{a,t} \oplus \mathrm{XOR}_b$.

Require: Public sets $(A_i, B_i)_{i=1}^m$, output $y \in \{0,1\}^m$, parameters k, q, a holding estimate $\beta > 1/2$, and a choice of Gauss or SubGauss.
Ensure: A seed candidate $x \in \{0,1\}^n$, or failure.
Choose or enumerate a candidate set $K \subseteq [n]$ with $|K| = k$.
Guess $x_K = \mathbf{1}_k$.
Select the outputs indexed by $I_{K,q}$ and compute the row count e.
Set $e_0 \leftarrow N$ for Gauss or $e_0 \leftarrow \lceil N/\beta \rceil$ for SubGauss.
if $e < e_0$ **then**
    return failure.
end if
for each $i \in I_{K,q}$ **do**
    Form the noisy XOR equation

$$\bigoplus_{v \in B_i \setminus K} x_v = y_i \oplus 1 \oplus (|B_i \cap K| \bmod 2).$$

end for
for each subsystem trial prescribed by the chosen solver **do**
    Sample R uniformly among the N-subsets of $I_{K,q}$.
    Apply Gaussian elimination to this subsystem.
    if the subsystem is consistent **then**
        Enumerate all solutions of the subsystem and verify the completed seed candidates against the full PRG output.
        return immediately if a candidate passes verification.
    end if
end for
return failure.

For a fixed (k, q), set $\beta = \beta_{k,q}(w)$. We discard pools with $e < N$ for Gauss or $e < \lceil N/\beta \rceil$ for SubGauss. Both solvers therefore use N-row subsystems. The full-row-rank estimates above do not account for additional free variables caused by dependent equations. Section 4.2 bounds the expected number of candidate solutions by Ψ_N. This bound also applies when the guessed seed bits are incorrect, accounting for the candidate-enumeration cost of unsuccessful guesses. Under the stated solver-cost assumptions, we use the following conservative rank-adjusted estimate

$$T_{\mathrm{BAPR}}(w; k, q) = S_1(n, w, k) \min\left\{ \frac{\Psi_N T_{\mathrm{G}}(N, N, \beta; m)}{\Pr[e \geq N]}, \frac{\Psi_N T_{\mathrm{SG}}(N, \lceil N/\beta \rceil, \beta; m)}{\Pr[e \geq \lceil N/\beta \rceil]} \right\}.$$

For simplicity, the rank and pool-survival corrections are applied to the entire cost expression. As in the underlying solver model, the polynomial cost of verifying each candidate is omitted. Optimizing the estimate over valid parameters gives

$$T_{\mathrm{BAPR}}(w) = \min_{\substack{1 \leq k \leq w,\ 0 \leq q \leq \min(a,k),\ q<t \\ \beta_{k,q}(w) > 1/2}} T_{\mathrm{BAPR}}(w; k, q).$$

Algorithm 2 summarizes the BAPR procedure used for these estimates.

4.4 Applications to Concrete Instances

For consistency with the cubic-elimination cost model in [DMÜ26], the concrete estimates in this section use the linear-algebra exponent $\omega = 3$.

A challenge predicate. We first consider $\mathrm{THR}_{64,32} \oplus \mathrm{XOR}_{10}$, a predicate used in challenge instances targeting 128-bit security [BCPR25]. Its threshold is $t = 32$. Deterministic pooling would therefore require $q \geq 32$, which is too expensive at the stretches considered here. BAPR instead uses $q < 32$, producing biased noisy XOR equations.

Table 2 compares BAPR with the balanced-seed selection variant of common-bias inversion. We reproduce its reported $T/\,\mathrm{Adv}$ estimates, including output-selection work; the BAPR entries use the recovery-cost model above.

Table 2: Estimates for $\mathrm{THR}_{64,32} \oplus \mathrm{XOR}_{10}$ at balanced seed weight. The $\mathrm{CB}_{\mathrm{bal}}$ estimates use selection and are from [DMÜ26, Table 10], with $m' = n$. Both methods use cubic elimination under their respective cost models.

n	s	method	parameter	μ	β	$\log_2(\mathrm{cost})$
256	5.0	$\mathrm{CB}_{\mathrm{bal}}$	–	–	–	77.86
256	5.0	BAPR	$k = 13,\ q = 13$	6044.8	0.9629	**55.9**
512	4.5	$\mathrm{CB}_{\mathrm{bal}}$	–	–	–	112.59
512	4.5	BAPR	$k = 22,\ q = 15$	789.8	0.9823	**65.4**
894	3.4	$\mathrm{CB}_{\mathrm{bal}}$	–	–	–	187.33
894	3.4	BAPR	$k = 45,\ q = 15$	921.7	0.9778	**105.2**
2048	3.0	$\mathrm{CB}_{\mathrm{bal}}$	–	–	–	398.16
2048	3.0	BAPR	$k = 117,\ q = 16$	1932.6	0.9826	**206.9**

Under the stated cost models, BAPR has substantially lower estimated recovery costs than the balanced-seed common-bias inversion variant. For $n = 256$, 512, and 894, these estimates remain below the proposed 128-bit target even at balanced seed weight.

More instances. We further evaluate BAPR on the parameters proposed by Fu et al. [FLLL26], taking $w = \lfloor n/2 \rfloor$. Table 3 compares BAPR with the common-bias inverter, evaluated both on average over seeds and at the same fixed weight.

Table 3: Estimates for the challenge parameters of Fu et al. [FLLL26, Table 2]. The three cost columns report base-two logarithms. $\mathrm{CB}_{\mathrm{avg}}$ gives the average-seed costs reported in [DMÜ26], while $\mathrm{CB}_{\mathrm{bal}}$ evaluates the inverter at $w = \lfloor n/2 \rfloor$. BAPR uses the same fixed seed weight.

κ	n	m	(a, b)	$\mathrm{CB}_{\mathrm{avg}}$	$\mathrm{CB}_{\mathrm{bal}}$	BAPR	(k, q)
80	91	281	$(47, 7)$	28.59	102.67	**72.99**	$(25, 15)$
128	141	485	$(47, 7)$	35.16	155.97	**98.02**	$(40, 16)$
256	285	1170	$(47, 7)$	51.38	303.89	**166.02**	$(72, 15)$
80	105	1075	$(57, 8)$	29.03	115.84	**68.02**	$(23, 16)$
128	155	1929	$(57, 8)$	34.82	169.16	**85.61**	$(32, 16)$
256	303	5274	$(57, 8)$	49.55	321.17	**135.10**	$(58, 16)$
80	146	21316	$(75, 10)$	31.36	170.57	**64.72**	$(25, 19)$
128	205	42025	$(75, 10)$	36.77	218.78	**80.62**	$(37, 21)$
256	365	133225	$(75, 10)$	49.76	382.45	**113.68**	$(55, 20)$

The low $\mathrm{CB}_{\mathrm{avg}}$ costs reflect the contribution of unbalanced seeds. These seeds are more likely to produce agreeing threshold outputs and can dominate the average success

probability. We therefore also evaluate recovery at balanced or nearest-balanced seed weights. Under the stated cost models, BAPR achieves estimated recovery costs below the claimed security levels for all these challenge instances even in this more conservative setting.

Application of DPR. DPR, the noiseless special case of BAPR, is particularly useful for small majority locality a, since forcing MAJ_a requires at least $(a+1)/2$ equal guessed inputs. For the $\mathrm{MAJ}_7 \oplus \mathrm{XOR}_4$ challenge of Boura et al. [BCPR25], take $n = 1024$, $m = n^2$, and balanced seed weight $w = 512$. Choosing $q = 4$ and $k = 80$ leaves $N = 944$ unknowns and gives $\mathbb{E}[e] \approx 1060.22$ selected equations. Averaging Lemma 2 over $e \geq N$, the smallest cutoff for which this bound gives $\Pr[d_K > \delta \mid e \geq N] \leq 1.31 \cdot 10^{-4}$ is $\delta = 39$; together with the row-count bound, this gives $p_{\mathrm{use}}(39) > 0.99976$. For $a = 7$ and $b = 4$, the stated work model gives $C_{\mathrm{sel}} = 7m$ for equation selection and $V = 11m$ for full candidate verification. With $\omega = 3$, substituting $\log_2(S_1/p_{\mathrm{use}}) \approx 84.83$ and $\log_2(C_{\mathrm{sel}} + N^\omega + V) \approx 29.68$ into eq. (15) yields

$$\log_2 T_{\mathrm{rec}} \approx \max\{84.83 + 29.68,\ 39 + 23.46\} = 114.51.$$

Under the stated cost-model assumptions, this estimate falls below the proposed 128-bit security target even for balanced seeds.

Experiments. Using $\mathrm{MAJ}_7 \oplus \mathrm{XOR}_4$ with $n = 1024$ and $q = 4$ as an example, we experimentally examine the rank behavior of the selected systems. For each pair (s, k), we sampled e using Lemma 1 and generated the rows from eq. (11), with 2000 independent matrices per pair.

Write d_K^{th} for the lower bound on $\mathbb{E}[d_K]$ in eq. (14). Its row-shortage and zero-column terms are evaluated using Lemma 1 and eq. (13), respectively. We also report a conservative threshold δ, defined as the smallest value for which Lemma 2, combined with the binomial distribution of e, gives $\Pr[d_K > \delta] \leq 1.31 \cdot 10^{-4}$. This probability includes the randomness of both the row count and the selected rows.

Table 4: Rank experiment for DPR with $\mathrm{MAJ}_7 \oplus \mathrm{XOR}_4$, $n = 1024$, and $q = 4$. The last two columns report the mean and maximum deficiency among 2000 matrices.

s	k	$\mathbb{E}[e]$	N	d_K^{th}	δ	mean d_K	max d_K
2.0	80	1060.22	944	14.78	40	16.60	36
2.5	32	867.50	992	124.50	231	123.27	209
2.5	38	1755.17	986	1.01	14	1.02	6
2.5	42	2635.85	982	0.032	12	0.034	1
2.7	28	1994.81	996	0.399	13	0.42	3
2.7	32	3470.00	992	0.0012	12	0.00	0

The observed mean deficiencies are close to the lower-bound estimates in Table 4. This indicates that row shortages and uncovered coordinates account for much of the observed rank loss. For $(s, k) = (2, 80)$, the mean deficiency is 16.60, compared with the zero-column lower bound 14.78, and the maximum is 36, below the unconditional cutoff 40. All tested maxima lie below their respective cutoffs. The upper bounds are conservative and these experiments illustrate the rank behavior predicted by the analysis.

4.5 The Intermediate-Locality Regime

D'Antona et al. [DMÜ26] leave open whether $\mathrm{MAJ}_a \oplus \mathrm{XOR}_b$ can retain exponential security when $a \in \omega(1) \cap O(\log n)$. We now analyze BAPR asymptotically to determine whether bias-amplified pooling remains effective in this intermediate-locality regime.

Theorem 1 (Subexponential seed recovery with BAPR)**.** *Let $a = a(n)$ satisfy $a \in \omega(1) \cap O(\log n)$, set $t = \lceil a/2 \rceil$, and let $m = n^s$ for a fixed $s > 1$. Let the XOR locality $b \geq 3$ be fixed, and choose*

$$q = \left\lceil \sqrt{a \log a} \right\rceil, \qquad k = \left\lceil 2\frac{q}{a} n \left(n^{1-s} \log n \right)^{1/q} \right\rceil.$$

BAPR admits an implementation with worst-case running time $2^{o(n)}$ and seed-recovery probability $1 - o(1)$ over independently sampled pairs (A_i, B_i), uniformly over all seeds.

Proof. Write $N = n - k$. The choice of q gives $q = o(a)$, $q \to \infty$, and $q \log q = o(\log n)$. Substituting into the definition of k gives

$$k = n^{1-o(1)}, \qquad \frac{ak}{n} \sim 2q \left(n^{1-s} \log n \right)^{1/q} = o(1).$$

In particular, $k = o(n)$, $q \leq k$, and $q < t$ for large n.

Fix a seed x of weight w. Take $c = 1$ if $w \geq n/2$ and $c = 0$ otherwise, and choose a set K of size k with $x_K = c\mathbf{1}_k$, depending only on x and independently of the sampled input pairs. Such a set exists for large n, since $k = o(n)$. We analyze this correct branch; the algorithm will try all k-subsets and both constant guesses. For $J = |A_i \cap K|$, the hypergeometric formula gives

$$\Pr[J = q] = (1 + o(1)) \binom{a}{q} \left(\frac{k}{n} \right)^q,$$

since $q^2/k + ak/n + a^2/n = o(1)$. Since $\Pr[B_i \cap K \neq \varnothing \mid A_i] \leq bk/(n-a) = o(1)$, the choice of k and the binomial coefficient lower bound $\binom{a}{q} \geq (a/q)^q$ show that the expected number of selected rows with $B_i \cap K = \varnothing$ is at least

$$(1 - o(1)) m \binom{a}{q} \left(\frac{k}{n} \right)^q \geq (1 - o(1)) 2^q n \log n.$$

This is much larger than $N \log N$, so the Chernoff lower-tail bound supplies at least $\lceil C_b N \ln N \rceil$ such rows with probability $1 - o(1)$. Conditional on this event, the first $\lceil C_b N \ln N \rceil$ qualifying supports are independent uniform b-subsets of $[n] \setminus K$, by symmetry and independence across outputs. The fixed-row-weight rank bound in Section 4.2 gives rank at least $N - 1$ for this submatrix. Adding the other selected rows cannot decrease rank, so $\operatorname{rank}(M_K) \geq N - 1$ with probability $1 - o(1)$.

To obtain a worst-case running-time bound, we enumerate low-weight error patterns in the chosen linearly independent equations and solve each corrected system by Gaussian elimination. Set

$$\varepsilon_n = \exp\left(-\frac{2q^2}{9a} \right) = o(1), \qquad h = \lceil N \sqrt{\varepsilon_n} \rceil = o(n).$$

For each guessed set and value $c \in \{0, 1\}$, form the selected XOR equations with right-hand sides $y_i \oplus c \oplus c(|B_i \cap K| \bmod 2)$. Discard the branch if $\operatorname{rank}(M_K) < N - 1$. Otherwise choose a basis of original selected rows in a fixed order, indexed by R. Enumerate all ways to flip at most h right-hand-side bits in these basis equations. For each choice, solve the corrected system for the unguessed bits and verify its at most two completed seeds against the full output. Return the seed that passes, or failure after all branches have been exhausted.

It remains to bound errors in the chosen basis on the fixed correct branch. For a selected row, condition on its XOR support B and on $J = |A \cap K| = j \geq q$. Put $r = |B \setminus K| \leq b$

and $v = \mathrm{wt}(x_{B\setminus K}) \le r$. The set $A \setminus K$ is uniform among the $(a-j)$-subsets outside $K \cup B$. Let U count its ones. On the all-one branch, $w \ge n/2$ and

$$U \mid (B, J = j) \sim \mathrm{Hypergeom}(N - r, w - k - v, a - j).$$

For $q \le j < t$, using $w \ge n/2$ and $t \le (a+1)/2$ gives

$$\mathbb{E}[U \mid B, J = j] - (t - j) \ge \frac{j-1}{2} - \frac{(a-j)(k+b)}{2(N-b)} \ge \frac{q}{3}$$

for large n, using $ak/n = o(1)$ and $ab/n = o(1)$. The hypergeometric lower-tail bound therefore gives error probability at most $\exp(-2q^2/(9(a-j))) \le \varepsilon_n$. For $j \ge t$, majority is forced to one.

For the all-zero branch, $w < n/2$ and $t \ge a/2$ similarly give $t - \mathbb{E}[U \mid B, J = j] \ge q/3$ for $q \le j \le a - t$ and sufficiently large n. Applying the hypergeometric upper-tail bound to the error event $U \ge t$ gives the same bound ε_n. For $j \ge a - t + 1$, majority is forced to zero. Thus both branches satisfy, after averaging over $j \ge q$ while still conditioning on B,

$$\Pr[\mathrm{MAJ}_a(x_A) \ne c \mid B,\ |A \cap K| \ge q] \le \varepsilon_n. \tag{16}$$

Let $\mathcal{F}$ record all XOR supports B_i and selection indicators $\mathbf{1}_{\{|A_i \cap K| \ge q\}}$. The indices R of the chosen linearly independent rows are determined by $\mathcal{F}$. Independence across outputs extends eq. (16) to conditioning on all of $\mathcal{F}$. For discarded pools put $R = \varnothing$. Conditional on $\mathcal{F}$, linearity of expectation bounds the expected number of errors in R by $N\varepsilon_n$. Markov's inequality therefore bounds the conditional probability of more than h errors by $N\varepsilon_n/h \le \sqrt{\varepsilon_n} = o(1)$. Together with the rank bound, this shows that the fixed correct branch includes the true error pattern and seed with probability $1 - o(1)$. The algorithm reaches this branch unless it has already returned a seed passing full verification. All bounds are uniform over fixed seeds.

Finally, the algorithm tries $\binom{n}{k}$ sets K and both constant guesses. Each basis has at most N rows, so the number of error patterns of weight at most h is bounded by $\sum_{j=0}^{h} \binom{N}{j}$. The rank check leaves at most two seed candidates per error pattern on every branch, including incorrect guesses. Selecting equations, checking rank, solving each system, and verifying its candidates all take polynomial time. Thus

$$T_{\mathrm{BAPR}}^{\mathrm{enum}} \le \binom{n}{k}\ \mathrm{poly}(n, m, a) \sum_{j=0}^{h} \binom{N}{j}.$$

The binomial coefficient bounds, together with $m = n^s$ for fixed s and $a = O(\log n)$, give

$$\log_2 T_{\mathrm{BAPR}}^{\mathrm{enum}} \le k \log_2 \frac{\exp(1)n}{k} + h \log_2 \frac{\exp(1)N}{h} + O(\log n) = o(n).$$

The last equality follows from $k = o(n)$, $h = o(N)$, and $N \sim n$, proving the worst-case bound $T_{\mathrm{BAPR}}^{\mathrm{enum}} = 2^{o(n)}$. □

Connection with concrete parameters. The instances in Table 3 provide concrete illustrations of the parameter regime studied in Theorem 1. Across the table, the ratio $a/\log_2 n$ lies between 5.76 and 10.43, placing these majority localities on the logarithmic scale considered in the theorem. For $a = 47, 57, 75$, the asymptotic choice $q = \lceil \sqrt{a \log a} \rceil$ gives $14, 16, 18$, respectively, while the optimized values in the table are 15–16, 16, and 19–21. This agreement suggests that the asymptotic rule captures the selection-threshold scale at these concrete parameters. The table illustrates the parameter regime, while its concrete cost estimates rely on the stated Gauss/SubGauss assumptions.

5 Recovery from Affine Restrictions

Pooled recovery is a general approach to seed recovery that exploits exact or biased affine relations exposed by partial assignments. Although related guessing and linearization ideas have appeared in previous work [DLR16, CDM+18, YGJL22], we give an explicit formulation of the method for a general class of predicates, analyze the selected systems in detail, and prove recovery without an additional rank assumption under the conditions stated below. Since recovery from biased affine relations requires a noise analysis specific to the predicate, we focus here on deterministic pooled recovery.

5.1 General Recovery Guarantees

Consider a fixed predicate P with d inputs, and let $1 \le r_{\mathrm{aff}} < d$ be fixed. Suppose that, for each of the values zero and one, fixing r_{aff} suitable inputs to that value makes P a nonconstant affine function of the remaining inputs. That is, the remaining expression is an XOR of at least one unfixed input, possibly with an added constant bit. The inputs chosen for the two values may differ.

Choose a set K of k seed positions and try both the all-zero and all-one guesses. For each guess, retain outputs whose inputs to be fixed come from K and whose other inputs come from outside K. A correct guess gives exact affine equations in the unguessed seed bits, which can be solved efficiently. Candidate seeds are then verified against the full output.

Theorem 2 (Direct recovery from affine restrictions)**.** *Let P satisfy the affine-restriction condition above, and let $m = n^s$, with input positions sampled according to the random model of Section 2. For fixed $1 < s < r_{\mathrm{aff}} + 1$, there is a Deterministic Pooled Recovery algorithm with guessed-set size*

$$k = \left\lceil C n^{1-(s-1)/r_{\mathrm{aff}}} (\ln n)^{1/r_{\mathrm{aff}}} \right\rceil$$

for a sufficiently large constant C depending only on the fixed predicate and parameters. The algorithm has worst-case subexponential running time $2^{O(k)}\,\mathrm{poly}(n) = 2^{o(n)}$. Recovery succeeds with probability $1 - o(1)$, uniformly over all seeds fixed independently of the input positions. No additional rank assumption is required.

Proof idea. For each guessed value, the selected equations contain a fixed number of unknown bits, with positions chosen uniformly and independently across outputs. The stated size of K supplies enough equations to apply the fixed-row-weight rank bounds, leaving at most one free variable with high probability. Discarding inconsistent systems and those with more free variables leaves at most two seed candidates per branch, including incorrect guesses. Randomly sampling sufficiently many guessed sets gives a correct guess with high probability; capping their number gives the stated worst-case time. The complete proof appears in Appendix A.1. □

For fixed odd $a \ge 3$ and fixed $b \ge 1$, the theorem applies to $\mathrm{MAJ}_a \oplus \mathrm{XOR}_b$ with $r_{\mathrm{aff}} = (a+1)/2$, since fixing that many majority inputs to the same value makes the majority part constant and leaves the XOR part.

Extension to layered matching wiring. Goldreich's original proposal used expander graphs to specify the local input sets [Gol00]. The preceding analysis assumes independently sampled output tuples. Motivated by this graph-based viewpoint, we also ask whether DPR applies to structured random connections with dependencies between tuples. We establish such an extension for the layered matching model below.

For fixed odd $a \geq 3$, fixed $b \geq 1$, and $b \mid n$, consider the following wiring for $\mathrm{MAJ}_a \oplus \mathrm{XOR}_b$. Each of L independent layers is a uniform partition of $[n]$ into blocks of size b. Every block is the XOR support of one output. Conditional on all partitions, its ordered a majority inputs are sampled uniformly without replacement from the block's complement, independently across blocks and layers. Thus $m = Ln/b$, and each seed position appears in exactly L XOR supports. This model allows dependence between output tuples within a layer.

Theorem 3 (DPR under layered matching wiring)**.** *In this model, fix $s > 1$, set $r = (a+1)/2$, and take $L = \lceil bn^{s-1} \rceil$, so $m = (1+o(1))n^s$. For $1 < s < r+1$, a randomized DPR variant with guessed-set size*

$$k = \left\lceil Cn^{1-(s-1)/r} (\ln n)^{1/r} \right\rceil$$

for a sufficiently large constant C has worst-case running time $2^{O(k)}\,\mathrm{poly}(n) = 2^{o(n)}$. It returns a valid seed with probability $1 - o(1)$, uniformly over seeds fixed independently of the wiring. Probability is over the partitions, majority inputs, and private guesses. No additional rank assumption is required. For fixed $s \geq r+1$, a polynomial-time variant has the same success guarantee.

Proof idea. The proof uses a fixed-weight subpool of the usual DPR equations. An exposure argument for the matching blocks and their selection outcomes rules out nonzero kernel vectors, apart from the all-one vector when b is even, with high probability. This replaces the independent-row rank argument; the capped guessing and verification procedure then gives the same recovery exponent. Appendix A.2 gives the proof. □

We view this result as a step towards analyzing pooled recovery on expander-based input structures. The present guarantee is limited to the layered matching model. We leave open the problem of obtaining comparable recovery guarantees for more general graphs under explicit conditions on expansion and the MAJ and XOR input sets.

5.2 Pooled Recovery for Alternative Predicates

The constructions of Sarkar [Sar25] achieve provable trade-offs among resiliency, non-linearity, and algebraic immunity. These properties are important criteria for choosing local predicates in Goldreich's construction, making these functions worth considering as alternative predicates. To illustrate the generality of pooled recovery, we apply DPR to selected constructions and assess their suitability in this setting.

We use the construction in Sarkar's Theorem 3 as our main example. The predicate can be written as[4]

$$F_{a,b}(\mathbf{Z}, \mathbf{U}, \mathbf{V}) = \mathrm{XOR}_b(\mathbf{Z}) \oplus \bigoplus_{i=1}^{a} U_i V_i \oplus \mathrm{MAJ}_a(\mathbf{U}). \tag{17}$$

The blocks $\mathbf{U}$ and $\mathbf{V}$ each have a inputs, and $\mathbf{Z}$ has b. Each output samples its $d = 2a+b$ seed positions uniformly without replacement in the order of the predicate inputs, independently across outputs.

[4]In the original paper, the function is denoted by f and written as the XOR of $m+1$ variables and a Maiorana–McFarland (MM) component (ψ, h)-MM_{2k}. The parameters k and $m+1$ correspond to our a and b, respectively. In our notation, this component is $\langle \psi(\mathbf{U}), \mathbf{V} \rangle \oplus h(\mathbf{U})$, with $h = \mathrm{MAJ}_a$. Since ψ only permutes coordinates, relabelling $\mathbf{V}$ turns the inner product into $\bigoplus_{i=1}^{a} U_i V_i$.

Ordinary DPR. For fixed odd $a \geq 3$ and fixed $b \geq 1$, guess $x_K = c\mathbf{1}_k$ for $c = 0, 1$, where $|K| = k$ and $N = n - k$.[5] Set $q = (a+1)/2$. Retain outputs with at least q of their U inputs in K and at least one guessed input in every pair $\{U_i, V_i\}$. These conditions force majority to a constant and make every product affine, without requiring all U_i to be guessed.

Let p_L be the probability that an output is retained. If exactly j of its U inputs lie in K, the remaining $a - j$ must have their V partners among the $k - j$ unused positions of K to ensure a guessed input in every pair. Multiplying each overlap term in eq. (9) by the conditional probability of this event gives

$$p_L = \sum_{j=q}^{\min(a,k)} \frac{\binom{k}{j}\binom{N}{a-j}}{\binom{n}{a}} \frac{\binom{k-j}{a-j}}{\binom{n-a}{a-j}}. \tag{18}$$

For $k \to \infty$ and $k/n \to 0$, each term in the sum is asymptotically equal to $\binom{a}{j}(k/n)^a$. Since a is fixed, eventually $k \geq a$, so the upper summation limit is a. As a is odd and $q = (a+1)/2$, symmetry of the binomial coefficients gives $\sum_{j=q}^{a} \binom{a}{j} = 2^{a-1}$, hence

$$p_L \sim 2^{a-1}(k/n)^a.$$

The pool includes the restrictions $U_i = c$ for every i, whose residuals are $\mathrm{XOR}_b(\mathbf{Z})$ at $c = 0$ and $1 \oplus \mathrm{XOR}_b(\mathbf{Z}) \oplus \mathrm{XOR}_a(\mathbf{V})$ at $c = 1$. These supply the subpools required by Theorem 2; retaining the other affine rows cannot decrease rank.

Quadratic variant of DPR. For odd $a \geq 5$, one can instead retain every output with at least $q = (a+1)/2$ of its U input positions in K, without requiring the remaining product terms to be covered. Majority is then constant and the residual has degree at most two. The selection probability is the overlap probability from eq. (9). For fixed a, q, as $k \to \infty$ and $k/n \to 0$, the leading contribution comes from outputs with exactly q of their U input positions in K, giving

$$p_Q = p_{\mathrm{sel}} \sim \binom{a}{q}(k/n)^q.$$

Treating the quadratic products as additional variables gives a linear system in $N + \binom{N}{2}$ variables. We estimate the solving cost under the assumption that this system uniquely determines the N unknown seed bits. Recovered candidates are verified against the public outputs.

Balancing $mp_L = \Theta(N)$ for ordinary DPR and $mp_Q = \Theta(N^2)$ for quadratic DPR gives, for fixed $2 < s < q + 2$, the equation-counting scales

$$k = \begin{cases} \Theta\left(n^{1-(s-1)/a}\right), & \text{ordinary DPR,} \\ \Theta\left(n^{1-(s-2)/q}\right), & \text{quadratic DPR.} \end{cases} \tag{19}$$

The quadratic guessing power is smaller precisely when $s > 3 + 2/(a-1)$. We take $a = 7$ as a concrete example, matching the majority locality of the $\mathrm{MAJ}_7 \oplus \mathrm{XOR}_4$ challenge. For this choice, the threshold is $s > 10/3$; the two scales are $n^{(8-s)/7}$ and $n^{(6-s)/4}$. This motivates considering quadratic DPR at larger stretches, where the reduction in guessing cost may offset the higher cost of solving the lifted system. We compare the total costs below under the stated recovery conditions.

[5] The method also applies to even $a \geq 4$, with forcing thresholds $a/2$ and $a/2 + 1$ for the all-one and all-zero branches, respectively.

Conditional cost comparison. The basic algebraic method treats each monomial as a separate variable and solves the resulting linear system without guessing seed bits. For example, MAJ_7 has algebraic degree 4. The predicate $F_{7,b}$ contains only linear, quadratic, and quartic terms, so direct linearization uses $n + \binom{n}{2} + \binom{n}{4} = \Theta(n^4)$ variables. A comparable number of output equations corresponds to $s = 4$ asymptotically. Under a sufficient-rank assumption, this gives a basic upper bound on the secure stretch exponent. The linear-algebra model estimates the solving cost as $\left(n + \binom{n}{2} + \binom{n}{4}\right)^{\omega} = \Theta(n^{4\omega})$.

We next estimate the concrete costs of ordinary and quadratic DPR. For both methods, we assume that the selected equations have sufficient rank to determine the remaining seed bits on a correct branch. Let E be the number of equations used per solving step and D the number of variables. We use $(E, D) = (N, N)$ for ordinary DPR and $(E, D) = (N^2, N(N+1)/2)$ for quadratic DPR. Trying both constant guesses on independent uniform k-subsets at balanced seed weight gives the conditional expected-cost model

$$T \approx \frac{\binom{n}{k}}{2\binom{n/2}{k}}\left(m + 2ED^{\omega-1}\right). \tag{20}$$

The binomial ratio is the expected number of guessed sets until one constant assignment is correct. The term m accounts for scanning the outputs once per guessed set, and $2ED^{\omega-1}$ accounts for solving both branches. By the Chernoff lower-tail bound in eq. (2), the probability of retaining fewer than E equations on a fixed correct branch is at most 0.01 if

$$m \geq \frac{E + \ln 100 + \sqrt{(\ln 100)^2 + 2E \ln 100}}{p}, \qquad p = p_L \text{ or } p_Q.$$

For comparison, we evaluate the direct-linearization benchmark using the full binomial expression for the number of variables. Under the stated rank assumption, we take $m = n + \binom{n}{2} + \binom{n}{4}$ output equations. The corresponding stretch exponent $s_0 = \log_n m$ is approximately 3.42 at $n = 256$ and 3.49 at $n = 512$. For each $m = n^s$ and ω, we choose the integer k that minimizes the estimated cost subject to the equation-count condition above. We use $\omega = 2$ as a simplified cost model for sparse linear algebra, motivated by Wiedemann's algorithm [Wie86], and $\omega = 2.38$ as a representative exponent for fast dense linear algebra. Table 5 reports the estimated costs for each (n, ω). We consider both the smallest s for which at least one DPR method has a lower estimated cost than the benchmark, and the benchmark stretch $s = s_0$.

Table 5: Conditional expected-cost estimates for $F_{7,b}$ at balanced seed weight.

			Ordinary DPR		Quadratic DPR		Benchmark
n	s	ω	k	$\log_2 T$	k	$\log_2 T$	$(s_0, \log_2 T)$
256	3.18	2	30	57.22	22	**53.93**	(3.42, 54.76)
	3.42		25	**53.26**	17	**48.50**	
	2.82	2.38	38	**64.39**	36	75.81	(3.42, 65.17)
	3.42		25	**53.26**	17	**54.06**	
512	3.36	2	39	70.50	27	**62.77**	(3.49, 62.80)
	3.49		35	67.20	23	**58.55**	
	3.23	2.38	43	**73.86**	33	75.60	(3.49, 74.73)
	3.49		35	**67.20**	23	**64.90**	

For each (n, ω), consider the interval from the smaller s reported in Table 5 to the exact benchmark stretch s_0. Under the stated cost and rank assumptions, at every s in this interval, at least one of the two DPR methods uses no more outputs and has a lower estimated cost than the direct-linearization benchmark. This suggests that an

assessment based only on direct linearization can overlook more efficient attacks. These results illustrate how pooled recovery can be used to assess alternative predicates for Goldreich's construction. In addition to resiliency, nonlinearity, and algebraic immunity, the cost of exposing and solving affine or quadratic equations through partial guessing should also be considered.

6 Affine Approximation for $f_{d,\epsilon,a} \oplus \mathrm{XOR}_b$

Beyond the DPR analysis above, we examine an alternative predicate family from [DMÜ26] that replaces MAJ within an XOR construction to reduce common bias. We analyze an affine approximation of this family and its implications for seed recovery.

For this replacement of MAJ, a useful affine approximation is available before any seed bits are guessed. For $0 \le d \le a/2$, $\epsilon \in \{0,1\}$, and $z \in \{0,1\}^a$, let $u = \mathrm{wt}(z)$. D'Antona et al. [DMÜ26] define $f_{d,\epsilon,a}$ by

$$f_{d,\epsilon,a}(z) = \begin{cases} 0, & u < d, \\ u + \epsilon \pmod 2, & d \le u \le a - d, \\ 1, & u > a - d. \end{cases}$$

Thus, in the middle range $d \le u \le a - d$, the predicate is the XOR of its input bits with the constant ϵ, so it agrees with an affine function over $\mathbb{F}_2$ throughout this interval. We evaluate this approximation at fixed seed weights under the recovery model below. Whenever $d \le \mathrm{wt}(x_{A_i}) \le a - d$, an output satisfies

$$\begin{aligned} y_i &= f_{d,\epsilon,a}(x_{A_i}) \oplus \mathrm{XOR}_b(x_{B_i}) \\ &= \epsilon \oplus \bigoplus_{v \in A_i \cup B_i} x_v. \end{aligned}$$

The second equality follows from $A_i \cap B_i = \varnothing$. Thus every such output gives a noiseless affine equation in the seed bits. The attack below uses the same affine equation for all outputs and succeeds when the chosen equations fall in Hamming-weight regions on which this approximation is correct.

For a fixed seed of weight w, let $p_{\mathrm{aff}}(n, w)$ be the probability that the affine equation above is correct for a random output. It is given by

$$p_{\mathrm{aff}}(n,w) = \sum_{\substack{0 \le u \le a \\ f_{d,\epsilon,a}(u) = u + \epsilon \pmod 2}} \frac{\binom{w}{u}\binom{n-w}{a-u}}{\binom{n}{a}},$$

where $f_{d,\epsilon,a}(u)$ denotes the value of the symmetric predicate on an input of weight u. The sum includes the middle interval and any extreme weights at which the affine expression happens to agree with the constant part of the predicate.

Recovery model. Assume $1 \le a + b < n$, as in all instances below. The coefficient vector of an affine equation is the incidence vector of $A_i \cup B_i$, and hence has weight $a + b$. The maximum possible number of independent equations is

$$N^\star = \begin{cases} n, & a + b \text{ odd}, \\ n - 1, & a + b \text{ even}. \end{cases}$$

In the even case, every coefficient row has even weight, so the all-one vector belongs to the kernel. We estimate the attack under the model that a maximum-rank set of rows is

Table 6: Conditional affine recovery estimates for selected $f_{d,\epsilon,a} \oplus \mathrm{XOR}_b$ instances at balanced seed weight. The cost includes candidate verification and uses $\omega = 2.38$.

n	s	(d, ϵ, a, b)	d/a	$p_{\mathrm{aff}}(n, n/2)$	$\log_2 T_{\mathrm{aff}}$
256	2.0	$(17, 0, 101, 10)$	0.168	$1 - 2^{-65.8}$	19.2
512	4.5	$(37, 0, 127, 23)$	0.291	$1 - 2^{-27.6}$	40.5
894	3.4	$(29, 0, 71, 17)$	0.408	0.9696	73.2

available, that its affine equations are correct independently with probability $p_{\mathrm{aff}}(n, w)$, and that repeated choices behave as independent trials. These assumptions are not proved here. One solve then succeeds with probability $p_{\mathrm{aff}}(n, w)^{N^\star}$. Including the cost of solving and checking a candidate against the available outputs gives

$$T_{\mathrm{aff}}(w) \approx p_{\mathrm{aff}}(n, w)^{-N^\star} \left(m + (N^\star)^\omega\right).$$

When a and b are both odd, the complete predicate is invariant under complementing the seed. In this case $a + b$ is even and the attack recovers a representative of the equivalent pair $\{x, \mathbf{1} \oplus x\}$.

Proposition 1 (Affine-approximation criterion)**.** *Suppose that n is even, the seed is balanced, and the stretch is polynomial. Under the affine recovery model above, the attack has $T_{\mathrm{aff}} = 2^{o(n)}$ whenever*

$$\frac{a/2 - d}{\sqrt{a}} \longrightarrow \infty.$$

In particular, for every constant $c > 0$, it is sufficient that $d \le (1/2 - c)a$ and a tends to infinity. The attack is polynomial under the stronger condition

$$\frac{(a/2 - d)^2}{a} \ge \left(\frac{1}{2} + c\right) \log n.$$

Proof. For balanced seeds, the two tail bounds in eq. (3) give

$$1 - p_{\mathrm{aff}}(n, n/2) \le 2 \exp\left(-\frac{2(a/2 - d)^2}{a}\right).$$

The first condition makes this error $o(1)$, so $-N^\star \log_2 p_{\mathrm{aff}} = o(n)$ in the stated cost model. Under the stronger condition the error is $O(n^{-1-2c})$, giving a bounded expected number of trials and polynomial cost. □

Thus the relevant quantity is not only the ratio d/a, but also the width of the affine interval measured relative to the natural fluctuation $\sqrt{a}$. Values of d close to $a/2$ leave only a narrow affine region and do not satisfy this criterion.

Finite-parameter estimates. Table 6 reports three illustrative $f_{d,\epsilon,a}$-type instances for which the affine approximation yields low estimated recovery costs under the stated model.[6] The estimates use balanced seed weight $w = n/2$ and $\omega = 2.38$.

These instances illustrate how reducing common bias can leave an accurate affine approximation that supports low recovery estimates under the stated model. Together with the analysis of Sarkar's construction in Section 5.2, these results suggest examining exact and approximate affine structure alongside the usual predicate criteria. Turning such structure into an attack also requires enough usable equations and a recovery analysis.

[6]These instances were proposed as challenge parameters in an earlier version of [DMÜ26]. They were subsequently removed, and footnote 8 of the revised version acknowledges our attacks.

7 Conclusion

We developed a pooled recovery framework for MAJ–XOR Goldreich generators. For $m = n^s$ with fixed $s > 1$, majority locality $a \in \omega(1) \cap O(\log n)$, and fixed XOR locality $b \geq 3$, BAPR achieves worst-case $2^{o(n)}$-time seed recovery with high probability under independent uniform sampling of input positions. This answers the open question of [DMÜ26] negatively in the parameter regime. The proof establishes the required rank bound and controls errors after selecting linearly independent equations. Under the stated cost models, our concrete pooled-recovery estimates also fall below the claimed security levels of selected challenge instances from ToSC 2025 [BCPR25] and EUROCRYPT 2026 [FLLL26], even at balanced or nearest-balanced seed weights.

For the noiseless special case, DPR, we establish a general recovery guarantee for fixed predicates satisfying the stated affine-restriction conditions. Within the specified stretch range, recovery takes subexponential time without an additional rank assumption. We also extend the DPR guarantee to a random layered matching model for fixed MAJ–XOR predicates.

We further examine potentially exploitable affine structure in alternative predicates. These examples suggest that security assessment should consider the cost of exposing and solving affine or low-degree relations alongside resiliency, nonlinearity, and algebraic immunity. An interesting direction for future work is to extend pooled recovery to input structures defined by expander graphs.

References

[ABBS25] Amit Agarwal, Carsten Baum, Lennart Braun, and Peter Scholl. Low-bandwidth mixed arithmetic in VOLE-based ZK from low-degree PRGs. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part IV*, volume 15604 of *Lecture Notes in Computer Science*, pages 396–426. Springer, 2025. doi:10.1007/978-3-031-91134-7_14.

[ABR12] Benny Applebaum, Andrej Bogdanov, and Alon Rosen. A dichotomy for local small-bias generators. In Ronald Cramer, editor, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*, pages 600–617. Springer, 2012. doi:10.1007/978-3-642-28914-9_34.

[AL18] Benny Applebaum and Shachar Lovett. Algebraic attacks against random local functions and their countermeasures. *SIAM J. Comput.*, 47(1):52–79, 2018. doi:10.1137/16M1085942.

[App12] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 805–816. ACM, 2012. doi:10.1145/2213977.2214050.

[AR16] Benny Applebaum and Pavel Raykov. Fast pseudorandom functions based on expander graphs. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 27–56, 2016. doi:10.1007/978-3-662-53641-4_2.

[BCM+24] Dung Bui, Geoffroy Couteau, Pierre Meyer, Alain Passelègue, and Mahshid Riahinia. Fast public-key silent OT and more from constrained Naor-Reingold. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 88–118. Springer, 2024. doi:10.1007/978-3-031-58751-1_4.

[BCPR25] Christina Boura, Geoffroy Couteau, Léo Perrin, and Yann Rotella. SoK: On shallow weak PRFs A common symmetric building block for MPC protocols. *IACR Trans. Symmetric Cryptol.*, 2025(3):289–336, 2025. doi:10.46586/TOSC.V2025.I3.289-336.

[BQ12] Andrej Bogdanov and Youming Qiao. On the security of Goldreich's one-way function. *Comput. Complex.*, 21(1):83–127, 2012. doi:10.1007/S00037-011-0034-0.

[CDD+24] Geoffroy Couteau, Lalita Devadas, Srinivas Devadas, Alexander Koch, and Sacha Servan-Schreiber. Quietot: Lightweight oblivious transfer with a public-key setup. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part II*, volume 15485 of *Lecture Notes in Computer Science*, pages 197–231. Springer, 2024. doi:10.1007/978-981-96-0888-1_7.

[CDM+18] Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi, and Yann Rotella. On the concrete security of Goldreich's pseudorandom generator. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 96–124. Springer, 2018. doi:10.1007/978-3-030-03329-3_4.

[CFP19] Colin Cooper, Alan M. Frieze, and Wesley Pegden. On the rank of a random binary matrix. *Electron. J. Comb.*, 26(4):P4.12, 2019. doi:10.37236/8092.

[Che52] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952. doi:10.1214/aoms/1177729330.

[CM01] Mary Cryan and Peter Bro Miltersen. On pseudorandom generators in NC^0. In *Mathematical Foundations of Computer Science 2001*, volume 2136 of *Lecture Notes in Computer Science*, pages 272–284. Springer, 2001. doi:10.1007/3-540-44683-4_24.

[DLR16] Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP family of stream ciphers. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 457–475. Springer, 2016. URL: https://eprint.iacr.org/2016/271, doi:10.1007/978-3-662-53018-4_17.

[DMR23] Aurélien Dupin, Pierrick Méaux, and Mélissa Rossi. On the algebraic immunity–resiliency trade-off, implications for Goldreich's pseudorandom generator. *Designs, Codes and Cryptography*, 91(9):3035–3079, 2023. doi:10.1007/s10623-023-01220-w.

[DMÜ26] Antonio Giulio D'Antona, Pierrick Méaux, and Akin Ünal. On the common bias of majorities: Poly-time attacks on THR-XOR PRGs. *IACR Cryptol. ePrint Arch.*, 2026:880, 2026. URL: https://eprint.iacr.org/2026/880.

[ER59] Paul Erdős and Alfréd Rényi. On random graphs. I. *Publicationes Mathematicae Debrecen*, 6(3–4):290–297, 1959. URL: https://www.renyi.hu/~p_erdos/1959-11.pdf, doi:10.5486/PMD.1959.6.3-4.12.

[FLLL26] Ximing Fu, Mo Li, Shihan Lyu, and Chuanyi Liu. Attacks on Goldreich's pseudorandom generators by grouping and solving. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology - EUROCRYPT 2026 - 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Rome, Italy, May 10-14, 2026, Proceedings, Part VI*, volume 16546 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2026. doi:10.1007/978-3-032-25333-0_1.

[Gol00] Oded Goldreich. Candidate one-way functions based on expander graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(90), 2000. URL: https://eccc.weizmann.ac.il/eccc-reports/2000/TR00-090/index.html.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi:10.1080/01621459.1963.10500830.

[LLMF26] Shihan Lyu, Mo Li, Chuhan Ma, and Ximing Fu. Improved cryptanalysis of local PRGs. Cryptology ePrint Archive, Paper 2026/2240, 2026. URL: https://eprint.iacr.org/2026/2240.

[MCJS19] Pierrick Méaux, Claude Carlet, Anthony Journault, and François-Xavier Standaert. Improved filter permutators for efficient FHE: better instances and implementations. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *Progress in Cryptology - INDOCRYPT 2019 - 20th International Conference on Cryptology in India, Hyderabad, India, December 15-18, 2019, Proceedings*, volume 11898 of *Lecture Notes in Computer Science*, pages 68–91. Springer, 2019. doi:10.1007/978-3-030-35423-7_4.

[Méa21] Pierrick Méaux. On the fast algebraic immunity of threshold functions. *Cryptography and Communications*, 13(5):741–762, 2021. doi:10.1007/s12095-021-00505-y.

[Méa22] Pierrick Méaux. On the algebraic immunity of direct sum constructions. *Discrete Applied Mathematics*, 320:223–234, 2022. doi:10.1016/j.dam.2022.05.021.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343. Springer, 2016. doi:10.1007/978-3-662-49890-3_13.

[MST06] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On ϵ-biased generators in NC^0. *Random Structures & Algorithms*, 29(1):56–81, 2006. doi:10.1002/rsa.20112.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2 edition, 2017. doi:10.1017/9781316651124.

[Sar25] Palash Sarkar. Constructions of efficiently implementable boolean functions with provable nonlinearity/resiliency/algebraic immunity trade-offs. Cryptology ePrint Archive, Paper 2025/1800, 2025. URL: https://eprint.iacr.org/2025/1800.

[Ünal23] Akin Ünal. Worst-case subexponential attacks on PRGs of constant degree or constant locality. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part I*, volume 14004 of *Lecture Notes in Computer Science*, pages 25–54. Springer, 2023. doi:10.1007/978-3-031-30545-0_2.

[Ver26] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2 edition, 2026. doi:10.1017/9781009490672.

[Wie86] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, 1986. doi:10.1109/TIT.1986.1057137.

[YGJL22] Jing Yang, Qian Guo, Thomas Johansson, and Michael Lentmaier. Revisiting the concrete security of Goldreich's pseudorandom generator. *IEEE Trans. Inf. Theory*, 68(2):1329–1354, 2022. doi:10.1109/TIT.2021.3128315.

A Proofs of the DPR Recovery Guarantees

A.1 Independent Random Wiring

We prove Theorem 2 by first deriving the selected equation distribution under independent random wiring, and then bounding the work of the capped recovery algorithm, including incorrect guesses.

Selection and row distribution. Let P satisfy the condition of Theorem 2. For each guessed value $c \in \mathbb{F}_2$, let $S_c \subseteq [d]$ be the set of predicate input indices fixed to c, and let T_c be the set of indices appearing in the resulting XOR. Thus, for some $\gamma_c \in \mathbb{F}_2$,

$$|S_c| = r_{\mathrm{aff}}, \qquad \varnothing \neq T_c \subseteq [d] \setminus S_c, \qquad P\big|_{z_j=c\ (j\in S_c)} = \gamma_c + \sum_{j\in T_c} z_j. \tag{21}$$

Write $b_c = |T_c| \geq 1$ for the number of remaining XOR inputs. The seed is fixed independently of the public input tuples.

Choose a global guessed set $K \subseteq [n]$ of size k, independently of the public tuples, and put $\bar{K} = [n] \setminus K$, $N = n - k$. On branch $x_K = c\mathbf{1}_k$, select precisely the outputs whose predicate inputs indexed by S_c use seed positions in K, while all other inputs use positions in $\bar{K}$. This rule depends on public positions and the attempted value c, not on the observed output values. With $(u)_v = u(u-1)\cdots(u-v+1)$, its single-row probability is

$$p_{\mathrm{aff}}(k) = \frac{(k)_{r_{\mathrm{aff}}}(N)_{d-r_{\mathrm{aff}}}}{(n)_d}. \tag{22}$$

When $k \to \infty$ and $k = o(n)$, this is $(1 + o(1))(k/n)^{r_{\mathrm{aff}}}$. Substitution on a selected row gives

$$\sum_{j \in T_c} x_{\sigma_{i,j}} = y_i + \gamma_c. \tag{23}$$

The equation is guaranteed to hold for the true seed on a correct branch. On an incorrect branch it is merely an equation to be tested for consistency; the algorithm makes no assumption about its right-hand side.

For fixed K, c, symmetry implies that the coefficient support of each selected row is a uniform b_c-subset of $\bar{K}$. Since outputs are sampled independently and selection is performed separately for each output, conditioning on all selection indicators leaves the selected rows independent with this same distribution. Whether a fixed block contains at least e selected rows depends only on these indicators. Hence, conditional on this event, its first e selected rows are still independent uniform weight-b_c rows.

For row weights at least three, we use the fixed-weight rank bound recalled in Section 4.2. The same sufficient scale $e = \lceil C_b N \ln N \rceil$ covers weights one and two. For $b = 1$, the rows cover every coordinate with high probability, giving rank N. For $b = 2$, their supports form graph edges, and the incidence matrix has rank $N - 1$ when the graph is connected. Discarding repeated edges leaves $(1 - o(1))e$ uniformly sampled distinct edges with high probability, so connectivity follows from the classical threshold [ER59] for sufficiently large C_2.

We now prove Theorem 2 using these rank bounds. For $1 < s < r_{\mathrm{aff}} + 1$, put $\alpha = 1 - (s-1)/r_{\mathrm{aff}}$.

Proof. Take C_* large enough for these rank bounds at both weights b_0, b_1, and set

$$k = \left\lceil C n^{\alpha} (\ln n)^{1/r_{\mathrm{aff}}} \right\rceil, \qquad e_0 = \lceil C_* N \ln N \rceil,$$

where the fixed constant C will be sufficiently large. For all sufficiently large n, $k \to \infty$, $k = o(n)$, and $k \le \lfloor n/2 \rfloor$. For fixed K, c, the selected count E is binomial and

$$\mu = m p_{\mathrm{aff}}(k) = (1 + o(1)) C^{r_{\mathrm{aff}}} n \ln n.$$

Choose C so that eventually $\mu \ge 2e_0$. By eq. (2), $\Pr[E < e_0] \le \exp(-\mu/8)$.

Set

$$p_{\mathrm{branch}} = \frac{\binom{\lfloor n/2 \rfloor}{k}}{\binom{n}{k}}, \qquad R_{\max} = \left\lceil 3 p_{\mathrm{branch}}^{-1} \ln n \right\rceil,$$

and choose $R_{\max}$ independent uniform k-subsets, independently of the public graph. On each set try both constant values. For each branch, take the first e_0 rows selected by its membership rule, skipping the branch if there are too few. Form eq. (23) and use Gaussian elimination. Discard inconsistent systems and systems with more than one free variable. Otherwise enumerate their at most two solutions, combine them with the guessed values on K, and verify each complete candidate against every original output. Return the first verified seed, or report failure after the trial cap.

Consider a fixed correct K, c. Its system contains the true unknown seed. Conditional on $E \ge e_0$, the selected-row distribution and rank bounds above ensure at most one free variable with probability $1 - o(1)$. Thus this branch succeeds with probability $1 - o(1)$, uniformly in the seed, the correct set, and its constant value.

For a fixed seed, choose a value c_* occurring at least $\lceil n/2 \rceil$ times. A sampled set lies entirely in those positions with probability at least p_{branch}. The probability that none of the $R_{\max}$ sets does so is at most $(1 - p_{\mathrm{branch}})^{R_{\max}} \le n^{-3}$. The first such set, when it exists, depends only on the seed and the private sampled sets, and is independent of the public graph. The preceding correct-branch bound therefore applies to it. Unless an earlier branch has already recovered a seed, the algorithm visits and succeeds on this branch with

probability $1 - o(1)$. No independence between successes on different guesses using the same graph is assumed.

Every branch, including an incorrect guess, scans at most m rows, performs polynomial-time linear algebra on an e_0-by-N system, and verifies at most two candidates. In particular, inconsistent or rank-deficient branches cannot cause an uncapped enumeration. Finally,

$$\log_2 p_{\text{branch}}^{-1} = k + O(k^2/n + k/n), \qquad \log_2 R_{\max} = k + O(k^2/n + \log\log n) = O(k).$$

The fixed trial and candidate caps prove the stated worst-case time. These asymptotic choices need only apply for sufficiently large n; finitely many smaller seed lengths can use exhaustive search, with its cost absorbed into a predicate-dependent constant. □

At $s = r_{\text{aff}} + 1$, the same proof uses $k = \lceil C(\ln n)^{1/r_{\text{aff}}} \rceil$ and gives polynomial time, since $2^{O((\ln n)^{1/r_{\text{aff}}})}$ is polynomial for $r_{\text{aff}} \geq 1$. For any fixed $s > r_{\text{aff}} + 1$, run this boundary algorithm on the first $n^{r_{\text{aff}}+1}$ outputs and verify candidates against all m outputs. The true seed remains among the successful branch's candidates, and the additional verification is polynomial.

Sampling and the running-time cap. The analysis above uses independent uniform subsets as sampling primitives. It also admits a bounded random-bit implementation. In each of the $R_{\max}$ trials, independently assign each of the n positions a random b_{key}-bit key, where $b_{\text{key}} = \lceil \log_2 R_{\max} + 5\log_2 n \rceil$, and take the k smallest keys. Skip a trial if any keys coincide. Conditional on distinct keys, their order is a uniform permutation, and this event is independent of the order. A union bound gives collision probability at most $R_{\max} n(n-1)2^{-b_{\text{key}}-1} \leq n^{-3}/2$ over all trials. Conditional on no collision, the sets have exactly the independent uniform law used in the proof. Key generation and sorting take polynomial work per trial, because $b_{\text{key}} = O(k + \log n) = O(n)$. Thus bounded sampling changes success by only $o(1)$, while preserving the worst-case running-time bound for every possible sequence of random choices.

A.2 Layered Matching Wiring

We prove Theorem 3 by bounding the rank of the coefficient matrix of the entire selected pool. The rows within a matching layer are dependent, so the independent-row rank bound used above does not apply. Throughout, $r = (a+1)/2$, K is a fixed k-subset chosen independently of the wiring, and $U = [n] \setminus K$, $N = n - k$. We work with sufficiently large n, with $r \leq k \leq n/2$ and $n \geq 32b$.

Selection. On the branch guessing $x_K = c\mathbf{1}_k$, retain a block $E \subseteq U$ only when its first r majority inputs lie in K and its other majority inputs lie in U. Conditional on all matching layers, the completions of distinct blocks are independent. Every block contained in U therefore passes this rule independently with probability

$$p_{\text{lay}}(k) = \frac{(k)_r (N-b)_{a-r}}{(n-b)_a}. \tag{24}$$

Blocks meeting K are not retained. On a correct branch, each retained output gives the exact equation $\sum_{v \in E} x_v = y_E + c$. Let M be the incidence matrix on U of all retained blocks, and write $p = p_{\text{lay}}(k)$.

Rank after selection. Set $\eta_b = 16^{-(b-1)}$ and $\kappa_b = \eta_b/(8b)$. We claim that, uniformly over fixed K, c,

$$\Pr\big[\operatorname{rank} M < N - \mathbf{1}_{\{b \text{ even}\}}\big] \leq (1 + \exp(-\kappa_b L p))^N - 1 + \mathbf{1}_{\{b \text{ odd}\}} 2^N \exp(-\kappa_b L p N). \tag{25}$$

For $S \subseteq U$, its indicator lies in $\ker M$ precisely when every retained block meets S evenly. Fix first a support of size $1 \le z \le N/2$. In one layer, expose $\lceil z/(2b) \rceil$ blocks successively, each anchored at an unexposed vertex of S, and expose their completions. Before each probe fewer than $z/2$ vertices have been removed, so an anchor exists and at least $N - z - z/2 \ge N/4$ vertices of $U \setminus S$ remain. The anchor's partners are a uniform $(b-1)$-subset of the other unexposed vertices. Since $N \ge n/2$ and $n \ge 32b$, the probability that all partners lie in $U \setminus S$ is at least η_b: each factor in the sampling probability is at least $1/16$. The empty product handles $b = 1$. The resulting block meets S exactly once and is retained with probability p.

These bounds remain valid after earlier unsuccessful probes. Conditional on the exposed blocks, their completions depend only on those blocks, not on the partition of the remaining vertices. Even if a completion mentions unexposed vertices, observing it or its selection outcome leaves the remaining matching uniform. Thus each probe has conditional probability at least $\eta_b p$ of finding a retained block meeting S oddly. Independence between layers now gives

$$\Pr[\mathbf{1}_S \in \ker M] \le (1 - \eta_b p)^{L\lceil z/(2b) \rceil} \le \exp(-\kappa_b L p z).$$

The union bound over these small nonempty supports is at most $(1 + \exp(-\kappa_b L p))^N - 1$.

If b is even, $\mathbf{1}_U \in \ker M$, and any further nonzero kernel vector or its complement has weight at most $N/2$. This proves eq. (25) for even b. For odd b, fix $S \subseteq U$ with $|S| > N/2$. In each layer expose $\lfloor N/(4b) \rfloor \ge N/(8b)$ blocks, again anchored at fresh vertices of S. Before each probe fewer than $N/4$ vertices have been removed. The same partner bound gives probability at least η_b that the whole block lies in S. If retained, this block meets S oddly. Consequently

$$\Pr[\mathbf{1}_S \in \ker M] \le \exp(-\kappa_b L p N).$$

There are at most 2^N such supports, proving the claim. In particular, whenever

$$Lp \ge (3/\kappa_b) \ln n, \tag{26}$$

the failure probability is at most $2n^{-2} + \mathbf{1}_{\{b \text{ odd}\}}(2n^{-3})^N = o(1)$.

Recovery and complexity. For $1 < s < r + 1$, take k as in the theorem. Then $k \to \infty$, $k = o(n)$, and

$$p = (1 + o(1))(k/n)^r, \qquad Lp = (1 + o(1))bC^r \ln n.$$

A sufficiently large C ensures eq. (26). Apply the capped guessing procedure of Appendix A.1, with

$$\pi_0 = \frac{\binom{\lfloor n/2 \rfloor}{k}}{\binom{n}{k}}, \qquad R = \left\lceil 3\pi_0^{-1} \ln n \right\rceil,$$

trying both constants on each of R independent uniform guessed sets. Here elimination uses *all* retained rows, discarding inconsistent systems and systems with more than one free variable, and verifying at most two candidates per branch. Every branch, including an incorrect guess, has polynomial cost, since $m = Ln/b$ is polynomial in n.

For any fixed seed, choose a value occurring at least $n/2$ times. With probability at least $1 - n^{-3}$, one of the guessed sets contains only that value. The first such set depends only on the seed and the private guesses, so is independent of the wiring. The uniform bound in eq. (25) applies to this correct branch and guarantees at most two candidates, including the true seed. This yields success probability $1 - o(1)$, uniformly over fixed seeds; it does not require one wiring to work for all seeds simultaneously. Since $\pi_0 \ge 4^{-k}$ for large n, the worst-case cost is $2^{O(k)}\,\mathrm{poly}(n)$. The bounded random-bit sampling implementation in Appendix A.1 applies unchanged.

At $s = r+1$, set $k = \lceil C(\ln n)^{1/r} \rceil$; the same argument gives polynomial time. For fixed $s > r+1$, take $k = r$. Then $p = (1+o(1))r!n^{-r}$, so $Lp = (1+o(1))br!n^{s-1-r} \gg \ln n$, and $R = O(\ln n)$. Finally, the usual full DPR pool contains this proof subpool. On a correct branch all its equations are exact, and adding rows cannot reduce rank. Thus the same capped recovery guarantee also holds when using that full pool.