

Witness Encryption for NP from SNARGs and Groups

Zhengzhong Jin
Northeastern University

Abstract

We construct the first witness encryption for NP in the generic group model from succinct non-interactive arguments (SNARGs) for NP. Our construction applies to any SNARG with subexponential soundness and polylogarithmic online verification time after input preprocessing.

Central to our result is the first Karp–Levin reduction from satisfiability for circuits of size $\text{polylog}(\lambda)$ to the minimum-distance problem for linear codes (GapMDC) over a prime field of size $\lambda^{\omega(1)}$, with an approximation factor of $\omega(\log \lambda)$, where λ is the security parameter. We obtain this reduction by adapting Hair and Sahai’s recent hardness result for GapSVP. Combining our reduction with the witness encryption framework due to Barta, Ishai, Ostrovsky, and Wu [CRYPTO 2020], we obtain the first unconditional extractable witness encryption for circuits of size $\text{polylog}(\lambda)$ in the generic group model. These techniques may be of independent interest.

1 Introduction

Witness encryption [GGSW13] allows a sender to encrypt a message under an NP statement x . Anyone holding an NP witness for x can decrypt the message, whereas semantic security is required when x is false. Witness encryption enables numerous cryptographic applications, including attribute-based encryption for Turing machines [GKP⁺13], secret sharing for NP [KNY14], obfuscation for null circuits [WZ17, VWW22], registered attribute-based encryption and flexible broadcast encryption [FWW23], registered threshold encryption and more [GHK⁺25].

Previously, witness encryption for NP was known from three routes: indistinguishability obfuscation, a general but heavyweight mechanism [GGH⁺13, JLS21]; less well-studied assumptions such as evasive LWE [VWW22, Tsa22], whose general form has faced recent attacks [BÜW24, AMYY25, HHY25, DJM⁺25, HJL25]; and heuristic candidate constructions based on affine determinant programs [BIJ⁺20] which have also faced recent cryptanalysis [Jo26]. In this work, we ask:

Can we find new ways to construct witness encryption for NP?

The BIOW Route to Witness Encryption in the Generic Group Model. Barta, Ishai, Ostrovsky, and Wu [BIOW20] proposed another route to witness encryption for NP in the generic group model (GGM). They hypothesize that there exists a Karp-Levin reduction from an NP-complete language to the approximate minimum distance problem for linear codes (GapMDC) over prime-order fields of size $\lambda^{\omega(1)}$, with an approximation gap of $\omega(\log \lambda)$. Then they construct witness encryption for GapMDC under this parameter regime. However, since no such reduction is currently known, this does not yield an explicit witness encryption algorithm for NP. Constructing witness encryption for NP in the generic group model therefore remains an intriguing open problem.

1.1 Our Results

Theorem 1.1 (Witness Encryption for NP, Informal Version of Theorem 4.1). *Assume the existence of subexponentially sound SNARGs for NP with $\text{poly}(\log n, \lambda)$ proof size and online verification time in the input-preprocessing model, where λ is the security parameter and n is the size of the input instance. Then there exists a witness encryption scheme for NP in the generic group model.*

The SNARGs for NP required by Theorem 1.1 can be instantiated directly from many known SNARGs for NP based on knowledge-type assumptions (for example, [GGPR13, BCI⁺13]¹). It can also be constructed from any SNARK for NP with $\text{poly}(\log n, \lambda)$ proof size and $\text{poly}(n, \lambda)$ verification time using a “hash-and-prove” trick: instead of proving the original long statement x directly, we hash x to a short digest h and prove that h has a preimage x that is a true statement. Then the verification time becomes $\text{poly}(\log n, \lambda)$ given the hash value.

Our Approach: Boosting Witness Encryption via SNARGs. Our key insight is that constructing witness encryption for NP does not require *fully* resolving the NP-hardness hypothesis for GapMDC proposed in [BIOW20]. Instead, we show that a *subexponential-time* reduction that runs in $2^{O(n^\epsilon)}$ time suffices, where $\epsilon \in (0, 1)$ is a constant. Given such a reduction, we can construct a “tiny” witness encryption scheme supporting satisfiability for circuits of size $\text{polylog}(\lambda)$. We then boost this tiny witness encryption to support all of NP using SNARGs for NP: to encrypt under an NP statement x , we use tiny witness encryption to encrypt under the SNARG verification relation for this x . Assuming subexponential security of the SNARG, the verification circuit size can be reduced to $\text{polylog}(\lambda)$ through complexity leveraging. To invoke the soundness of the SNARG in the security proof, we require the tiny witness encryption scheme to be *extractable*. This is captured by the following theorem.

Theorem 1.2 (“Tiny” Witness Encryption for $\text{polylog}(\lambda)$ -Size Circuits, Informal Version of Theorem 3.5). *There exists an unconditional extractable witness encryption scheme for the satisfiability problem for any circuit of size $\text{polylog}(\lambda)$ in the generic group model, where λ is the security parameter.*

Central to the construction of such a tiny witness encryption is a new reduction from SAT to the approximate minimum distance problem for linear codes (GapMDC). We first recall the definition of GapMDC.

Approximate Minimum Distance for Linear Codes (GapMDC). Let $p = p(n)$ be prime, and let $\beta = \beta(n) \geq 1$ be a real-valued function. The problem β -GapMDC over the finite field $\mathbb{F}_p$ is defined as follows: given a parity-check matrix $\mathbf{H} \in \mathbb{F}_p^{n \times k}$ of a linear code and a distance bound d , decide between the following two cases.

- YES CASE: There exists a non-zero vector $\mathbf{c} \in \mathbb{F}_p^{n \times 1}$ such that $\mathbf{H}^\top \mathbf{c} = \mathbf{0}$ and $\text{wt}(\mathbf{c}) \leq d$, where the Hamming weight $\text{wt}(\cdot)$ is defined as the number of non-zero entries.
- NO CASE: For every non-zero vector $\mathbf{c} \in \mathbb{F}_p^{n \times 1}$ satisfying $\mathbf{H}^\top \mathbf{c} = \mathbf{0}$, we have $\text{wt}(\mathbf{c}) \geq \beta \cdot d$.

¹We cannot directly instantiate the SNARG in the random oracle model/GGM and claim witness encryption in the GGM alone, since our construction makes non-black-box use of the SNARG verifier.

Recall that the witness encryption for NP in [BIOW20] requires a reduction from an NP-complete language to GapMDC with an approximation gap $\beta = \omega(\log \lambda)$ over a prime-order field of size $\lambda^{\omega(1)}$. Earlier work [DMS03] gives polynomial-time reductions from NP-complete problems to GapMDC with an approximation factor $\beta = O(1)$ over fields of polynomial size. This constant factor can be amplified to a nearly polynomial factor $2^{(\log n)^{1-\epsilon}}$ for any constant $\epsilon > 0$ via a quasi-polynomial-time reduction. However, this amplification does not yield witness encryption even for $\text{polylog}(\lambda)$ -size SAT instances, since the reduction's running time still grows polynomially with the prime modulus p , while we need $p = \lambda^{\omega(1)}$. Later work [CW09, AK14, Mic14, BGLR25] derandomized, simplified, and refined these reductions. However, these reductions still run in time that grows with p .

In this work, we overcome this issue and obtain the following new hardness result for GapMDC over an exponentially large prime field by modifying the recent hardness reduction for GapSVP by Hair and Sahai [HS26].

Theorem 1.3 (Karp–Levin reduction from SAT to GapMDC, Simplified Version of Theorem 5.1). *For every constant $A > 0$, there exists a deterministic reduction that, for all sufficiently large n , takes as input a SAT instance φ of size at most $(\log n)^A$ and a prime $2^n < p < 2^{n+1}$, and outputs a $\lceil (\log_2 n)^2 \rceil$ -GapMDC instance over $\mathbb{F}_p$. The reduction runs in polynomial time in n and is a Levin reduction:*

- **Witness map for completeness.** *There exists a deterministic polynomial-time algorithm that maps any satisfying assignment of φ to a non-zero codeword satisfying the YES CASE condition of GapMDC.*
- **Witness recovery for soundness.** *There exists a deterministic algorithm that, given any non-zero codeword violating the NO CASE condition, recovers a satisfying assignment for the original SAT instance φ .*

Combining Theorem 1.3 with BIOW gives the tiny extractable witness encryption of Theorem 1.2. We note that the witness map property (“half-Levin” reduction) suffices for [BIOW20], whereas our extractability guarantee in Theorem 1.2 crucially relies on the witness recovery property in Theorem 1.3.

1.2 Technical Overview

Our starting point is the BIOW framework for constructing witness encryption via GapMDC in the generic group model. We first recall their framework.

Recall: BIOW Framework of Witness Encryption. The work of Barta, Ishai, Ostrovsky, and Wu [BIOW20] constructs witness encryption for NP through the following framework. They assume a Karp–Levin reduction from an NP-complete problem to $\omega(\log \lambda)$ -GapMDC over a $\lambda^{\omega(1)}$ -size field. Then they construct a witness encryption scheme for GapMDC for this parameter regime in the generic group model, as follows.

- **Encryption:** Given a β -GapMDC instance $(\mathbf{H} \in \mathbb{F}_p^{N \times K}, d)$, the encryption algorithm samples

$$\mathbf{r} \leftarrow \mathbb{F}_p^{K \times 1}, \quad s \leftarrow \mathbb{F}_p, \quad \mathbf{t} \leftarrow \mathbb{F}_p^{N \times 1}, \quad \mathbf{e} \leftarrow \mathcal{D}_{\epsilon/d}^{N \times 1},$$

where $\mathcal{D}_{\epsilon/d}$ is the distribution in which each coordinate is independently uniform over $\mathbb{F}_p$ with probability ϵ/d and is 0 with probability $1 - \epsilon/d$. We choose $\epsilon \in (0, 1)$ to be a small constant, such as $1/10$.

The ciphertext is

$$(g^{\mathbf{H}\mathbf{r}+s\cdot\mathbf{t}+\mathbf{e}}, \mathbf{t}, g^{s+m})$$

where g generates a cyclic group of prime order p .

- **Decryption:** Given a ciphertext $(g^{\mathbf{H}\mathbf{r}+s\cdot\mathbf{t}+\mathbf{e}}, \mathbf{t}, g^{s+m})$ and a non-zero codeword witness $\mathbf{c} \in \mathbb{F}_p^{N \times 1}$ where $\text{wt}(\mathbf{c}) \leq d$, the decryption algorithm first computes homomorphically

$$g^{\mathbf{c}^\top \cdot (\mathbf{H}\mathbf{r}+s\mathbf{t}+\mathbf{e})} = g^{s\mathbf{c}^\top \mathbf{t} + \mathbf{c}^\top \mathbf{e}},$$

because $\mathbf{c}^\top \mathbf{H} = 0$. The decryptor can also compute $\mathbf{c}^\top \mathbf{t}$ publicly because $\mathbf{t}$ is part of the ciphertext. For any fixed $\mathbf{c}$, a direct union bound over the randomness of $\mathbf{e}$ shows that $\mathbf{c}^\top \mathbf{e} = 0$ except with probability ϵ . The decryptor can therefore recover g^s from the equation above by computing the inverse of $\mathbf{c}^\top \mathbf{t}$ in $\mathbb{F}_p^\times$, and can then recover the bit m .

Extractability. Implicit in BLOW is that the witness encryption scheme for GapMDC above is extractable. In the GGM, the adversary receives the group elements in the ciphertext and can query the group oracles, which support linear group operations and zero testing. Intuitively, the adversary can obtain nontrivial information only through zero testing, and every group handle submitted for zero testing must be a linear combination of the group elements that the adversary received. Specifically, let $(\mathbf{c}_1, c_2) \in \mathbb{F}_p^N \times \mathbb{F}_p$ denote the coefficients of a queried linear combination of $(g^{\mathbf{H}\mathbf{r}+s\mathbf{t}+\mathbf{e}}, g^{s+m})$. Then the adversary can only obtain nontrivial information whenever $(\mathbf{c}_1, c_2) \neq \mathbf{0}$ but

$$\mathbf{c}_1^\top (\mathbf{H}\mathbf{r} + s\mathbf{t} + \mathbf{e}) + c_2(s + m) = 0.$$

Reorganizing the equation, we get

$$(\mathbf{c}_1^\top \mathbf{H}) \cdot \mathbf{r} + (\mathbf{c}_1^\top \mathbf{t} + c_2) \cdot s + \mathbf{c}_1^\top \mathbf{e} + c_2 m = 0.$$

Informally, whenever this happens for the first time in the adversary's queries, we can use the independence of $\mathbf{r}, s, \mathbf{e}$ from $\mathbf{c}_1, c_2$ to prove that $\mathbf{c}_1$ is a codeword with weight less than $\beta \cdot d$ in the code specified by $\mathbf{H}$. This gives an extractable witness encryption for GapMDC. We note that this overview is informal and we defer more details to Section 3.1. To achieve a negligible advantage, they need the prime-order group size p to be $\lambda^{\omega(1)}$ and set the approximation gap $\beta = \omega(\log \lambda)$.

However, to support all of NP, this construction requires an explicit Karp–Levin reduction from SAT to GapMDC for this parameter regime. Such a reduction is currently out of reach.

Our New Hope: Hair–Sahai Reduction for Hardness of GapSVP. Recent work of Hair and Sahai [HS26] suggests a new route to achieve deterministic hardness of GapSVP. GapSVP is the natural analogue of GapMDC for lattices: given an integer lattice spanned by a basis $\mathbf{B}$ and a threshold h , the problem asks whether the lattice contains a non-zero vector of ℓ_2 norm at most h or whether every non-zero vector has ℓ_2 norm greater than $\beta \cdot h$. Hair and Sahai give a deterministic subexponential-time reduction from SAT to GapSVP with a near-polynomial approximation factor.

Earlier works [Ajt98, CN98, Mic01, Kho06a, Kho05, HR07] show hardness of GapSVP through randomized reductions from GapCVP using *locally dense lattices*. Their completeness proofs rely on pigeonhole-type counting arguments to prove the existence of short lattice vectors, without directly providing an efficient algorithm to construct such vectors. Thus, these reductions are not Levin reductions since they do not give the witness maps required in Theorem 1.3. Hair and Sahai instead take a different route through Khot's quasi-random PCPs [Kho06b, KS16] to achieve a deterministic reduction. As we will explain later, we observe that this route makes *most* steps of their reduction Levin-style. The key property we need from their reduction is captured by the following theorem.

Theorem 1.4 ([HS26] Theorem 3.2, Lemma 5.4, Lemma 5.3). *There exists a reduction algorithm that takes any circuit-satisfiability instance of size n as input and outputs an integer matrix $\mathbf{B} \in \mathbb{Z}^{M \times N}$ and a bound $h \in \mathbb{Z}$ for GapSVP, where $\mathbf{B}$ is a basis for an integer lattice and h is the GapSVP bound. The running time of the reduction, N , and M are all bounded by*

$$\exp\left(2^{\sqrt{\log n}} (\log \log n)^{O(1)}\right).$$

Moreover, $\mathbf{B}$ consists of a matrix $\mathbf{C}$ arranged as

$$\mathbf{B} = \left[\underbrace{\mathbf{C} \mid \cdots \mid \mathbf{C}}_{2h \text{ copies}} \mid \mathbf{I} \right], \text{ where } \|\mathbf{C}\|_\infty \leq M$$

Furthermore, for any sufficiently large n ,

- If the input SAT instance is satisfiable, then there exists a non-zero $\mathbf{x} \in \{+1, 0, -1\}^{1 \times M}$ with $\text{wt}(\mathbf{x}) \leq h$ such that $\mathbf{x}\mathbf{C} = \mathbf{0}$.
- If the input SAT instance is unsatisfiable, then for every non-zero vector $\mathbf{x} \in \mathbb{Z}^{1 \times M}$ with $\text{wt}(\mathbf{x}) < 2h$, we have $\mathbf{x}\mathbf{C} \neq \mathbf{0}$.

In contrast to the preceding discussion, all arithmetic in this theorem statement is over $\mathbb{Z}$, rather than $\mathbb{F}_p$.

We next show that, for any sufficiently large prime p , these properties ensure that the linear code generated by

$$\mathbf{B}_p := \mathbf{B} \pmod{p},$$

together with the same threshold h , gives a hard instance of GapMDC with approximation gap $\beta = 2$ over the large prime-order field $\mathbb{F}_p$. We can then apply standard code tensoring to amplify this gap to $\omega(\log \lambda)$. Next, we prove $(\mathbf{B}_p, h)$ is an instance of 2-GapMDC.

Attempt to Prove the Correctness of the Reduction. We consider two cases.

- If the SAT instance is satisfiable, then Theorem 1.4 gives a non-zero vector $\mathbf{x} \in \{+1, 0, -1\}^{1 \times M}$ with $\text{wt}(\mathbf{x}) \leq h$ such that

$$\mathbf{x}\mathbf{B} = [\mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{x}] \pmod{p}.$$

Hence, $\text{wt}(\mathbf{x}\mathbf{B}_p \pmod{p}) = \text{wt}(\mathbf{x} \pmod{p}) \leq h$.

- If the SAT instance is unsatisfiable, we must show that $\text{wt}(\mathbf{x}\mathbf{B}_p \bmod p) \geq 2h$. We first observe that Theorem 1.4 gives the corresponding bound over $\mathbb{Z}$: every non-zero $\mathbf{x} \in \mathbb{Z}^{1 \times M}$ satisfies $\text{wt}(\mathbf{x}\mathbf{B}) \geq 2h$. Indeed,

$$\mathbf{x}\mathbf{B} = [\mathbf{x}\mathbf{C} \quad \mathbf{x}\mathbf{C} \quad \cdots \quad \mathbf{x}\mathbf{C} \quad \mathbf{x}].$$

If $\text{wt}(\mathbf{x}) < 2h$, then Theorem 1.4 implies that $\mathbf{x}\mathbf{C} \neq \mathbf{0}$, so the $2h$ copies of $\mathbf{x}\mathbf{C}$ give $\text{wt}(\mathbf{x}\mathbf{B}) \geq 2h$. Otherwise, the final block $\mathbf{x}$ alone has Hamming weight at least $2h$.

The remaining issue is reduction modulo p . The bound $\text{wt}(\mathbf{x}\mathbf{B}) \geq 2h$ over $\mathbb{Z}$ does not directly imply $\text{wt}(\mathbf{x}\mathbf{B}_p \bmod p) \geq 2h$, since a non-zero integer entry may become zero modulo p .

Dealing with Modulo p . Here we crucially use the bound $\|\mathbf{C}\|_\infty \leq M$. We first claim that, if the SAT instance is unsatisfiable, the submatrix $\mathbf{C}_S$ formed by any set S of fewer than $2h$ rows has full row rank $|S|$ over $\mathbb{Q}$. Otherwise, there would exist a non-zero vector $\mathbf{x} \in \mathbb{Q}^{1 \times M}$ supported on S such that $\mathbf{x}\mathbf{C} = \mathbf{0}$. Clearing denominators yields a non-zero integer vector with the same support whose product with $\mathbf{C}$ is zero, contradicting the soundness guarantee of Theorem 1.4.

We next prove that $\mathbf{C}_S$ remains full rank over $\mathbb{F}_p$ when p is sufficiently large. Because $\mathbf{C}_S$ has full row rank, it contains a square submatrix Δ of order $|S|$ such that $\det(\Delta) \neq 0$. By Hadamard's inequality on determinants,

$$|\det(\Delta)| \leq (M \cdot \|\Delta\|_\infty^2)^{M/2} \leq M^{3M/2}.$$

Thus, as long as $p > M^{3M/2}$, we have $\det(\Delta) \bmod p \neq 0$. This proves that $\mathbf{C}_S$ has full row rank over $\mathbb{F}_p$. Note that in our setting of witness encryption, even when $M = \text{poly}(\lambda)$, $p > M^{3M/2}$ can be satisfied if we choose p to be an $\Omega(M \log M)$ -bit prime.

The remainder of the proof is straightforward: because $\mathbf{C}_S$ has full row rank over $\mathbb{F}_p$, every non-zero vector $\mathbf{x} \in \mathbb{F}_p^{1 \times M}$ whose non-zero entries are fully contained in S satisfies $\mathbf{x}_S \mathbf{C}_S \neq \mathbf{0} \pmod{p}$. The rest of the proof for $\text{wt}(\mathbf{x}\mathbf{B} \bmod p) \geq 2h$ then follows the same strategy as the previous paragraph in the setting over $\mathbb{Z}$.

Constructing an Efficient Witness Map. The reduction above from SAT to GapMDC runs in $\exp(2^{\sqrt{\log n}(\log \log n)^{O(1)}})$ time. Thus, when $n = \text{polylog}(\lambda)$, this running time is polynomial in λ . This almost gives us our Theorem 1.3. The only remaining issue is that the Hair-Sahami reduction does not provide an efficient algorithm to compute an NP witness for GapMDC when the SAT instance is satisfiable. Indeed, all but the final step of the reduction given by Theorem 1.4 are “constructive”, so a witness can be traced through those steps given a satisfying assignment to the SAT instance. To construct a polynomial-time witness map as required by Theorem 1.3, we observe that Hair and Sahai indeed prove the following stronger property in their proof of the completeness case of Theorem 1.4.

Lemma 1.5 (Weaker version of [HS26] Claim 5.5). *If the SAT instance is satisfiable, then there exists a set of fewer than h rows $S \subseteq [M]$ such that the submatrix of $\mathbf{C}$ formed by the rows in S , denoted by $\mathbf{C}_S$, is “compressive”: the submatrix $\mathbf{C}_S$ has fewer than $|S|$ non-zero columns.*

[HS26, Claim 5.5] proves a stronger bound $|S|/\sqrt{n}$ on the non-zero columns of $\mathbf{C}_S$. The pigeonhole principle then implies the existence of the vector $\mathbf{x} \in \{+1, 0, -1\}^{1 \times M}$ in Theorem 1.4 such that $\mathbf{x}\mathbf{C} = \mathbf{0}$ over $\mathbb{Z}$. This application of the pigeonhole principle is the only “non-constructive” step in

their proof of completeness for their reduction. In our setting, however, the arithmetic is over $\mathbb{F}_p$. Given such a subset S , we can therefore use Gaussian elimination directly to obtain a vector $\mathbf{x} \in \mathbb{F}_p$ whose non-zero entries are in S such that $\mathbf{x}_S \mathbf{C}_S = \mathbf{0} \pmod{p}$. This gives us an efficient witness map.

Achieving Extractability of Witness Encryption via Witness Recovery. To construct the tiny extractable witness encryption for polylogarithmic-size circuits in our Theorem 1.2, the extractability in the witness encryption for GapMDC described earlier is not sufficient, since the extractor for the witness encryption for GapMDC can only produce a codeword $\mathbf{c}$ with $\text{wt}(\mathbf{c}) < \beta \cdot d$. To obtain a witness for the original NP problem, we need an algorithm to “reverse” the extracted codeword $\mathbf{c}$ back to the original satisfiable assignment of SAT. This property is formalized as the “witness recovery for soundness” condition in Theorem 1.3. A reduction that has both a “forward” witness map and a “backward” witness recovery algorithm is called a Levin reduction.

We observe that the reduction we described so far is indeed a Levin reduction. We prove this by examining every step of the reduction carefully and proving that each step is a Levin reduction. The following is an overview of the long chain of reductions in [HS26] that leads to Theorem 1.4 and Lemma 1.5.

$$\begin{array}{c} \text{SAT} \xrightarrow[\text{Elementary proof}]{\text{[BGLR25]}} \text{GapMDC}_{\mathbb{F}_2} \xrightarrow[\text{Expander amplification}]{\text{[HS26]}} \text{GapMDC}_{\mathbb{K}/\mathbb{F}_2} \xrightarrow[\text{PCP Techniques}]{\text{[Kho06b, KS16]}} \\ \text{Quasi-random PCP/Quasi-random dense hypergraph} \xrightarrow{\text{[HS26]}} \text{Hair–Sahai’s } (\mathbf{C}_\varphi, h_\varphi). \end{array}$$

The first step of their reduction is an elementary proof of NP-hardness for GapMDC over $\mathbb{F}_2$ with constant gap $\beta = \Theta(1)$ given by [BGLR25]. We show it is indeed a Karp–Levin reduction in Lemma A.1. The second step is an amplification of the gap of GapMDC in [HS26], whose proof uses standard expander techniques. We show this step is also a Karp–Levin reduction in Lemma A.4. Then the third step of the reduction utilizes PCP techniques developed in [Kho06b] and [KS16] to reduce the amplified GapMDC instance to an intermediate problem called *quasi-random dense hypergraph problem* QRDH (or, equivalently, quasi-random PCPs [Kho06b]). Indeed, most of the PCPs in the literature can be viewed as Karp–Levin reductions, since the honest PCP prover algorithm can be viewed as the witness map, and to prove the soundness, the common proof technique is to construct an NP witness from any PCP proof that passes the verification with high probability. This gives us the witness recovery algorithm in the Levin reduction. We verify in Lemmas A.7 and A.8 that this also holds for Khot’s quasi-random PCPs, with two modifications: we use the *conditional expectations* to derandomize the inner PCP’s witness-recovery procedure in [KS16], and use Reed–Muller list decoding [PW04] to recover the witness in Khot’s outer PCP [Kho06b]. Finally, we verify that the reduction in Hair–Sahai from the intermediate hypergraph problem QRDH to their matrix $\mathbf{C}$ in Theorem 1.4 is also a Levin reduction in Lemma A.14 and Theorem A.15.

This proves that our reduction from SAT to GapMDC is a (subexponential time) Karp–Levin reduction. We defer the details to Appendix A. In the appendix, we combine [HS26, Theorem 3.2, Claim 5.5] (restated as Theorem 1.4 and Lemma 1.5 above) into a single Theorem 5.2 and prove it is a Levin reduction.

Boosting from Polylogarithmic-Size Circuits to NP via SNARGs. Combining our subexponential time reduction from SAT to GapMDC with BIOW’s witness encryption for GapMDC, we

obtain a “tiny” witness encryption for $\text{polylog}(\lambda)$ size circuits. Our final idea is to boost this tiny witness encryption scheme for polylog -size circuits to all of **NP**. To achieve this, we use the tiny witness encryption to encrypt under the SNARG verification circuit. Specifically, let $V(\mathbf{d}_x, \pi)$ be the SNARG verification circuit, which takes as input a short digest $\mathbf{d}_x$ preprocessed from the instance x and a SNARG proof π , and verifies the SNARG proof. We then use witness encryption under the circuit $V(\mathbf{d}_x, \cdot)$, with $\mathbf{d}_x$ hardcoded into the circuit.

To prove security, we first use extractable witness encryption to obtain a witness for the SNARG verification circuit $V(\mathbf{d}_x, \cdot)$ and this breaks the soundness of the underlying SNARG for **NP**. This completes the security proof.

1.3 Organization of this paper

This paper is organized as follows. In Section 3, we construct a “tiny” extractable witness encryption scheme for **SAT** instances of size $\text{polylog}(\lambda)$. In Section 4, we use SNARGs to boost this scheme to support all of **NP**. In Section 5, we present our Karp–Levin reduction from **SAT** instances of size $\text{polylog}(\lambda)$ to **GapMDC**, with parameters suitable for witness encryption, assuming that the reduction to the Hair–Sahai matrix (Theorem 5.2) is Karp–Levin. Finally, in Appendix A, we prove that the reduction in Hair–Sahai indeed has the required Karp–Levin properties (Theorem 5.2).

2 Preliminaries

Notation. For a positive integer n , let $[n] := \{1, \dots, n\}$. We write $x \sim D$ or $x \leftarrow D$ for sampling x from a distribution D , and $x \xleftarrow{\$} S$ (or $x \leftarrow S$) for uniform sampling from a finite set S . For an algorithm $\mathcal{A}$, $y \leftarrow \mathcal{A}(x)$ denotes its output on input x , using fresh randomness if $\mathcal{A}$ is randomized. We use PPT for probabilistic polynomial time, $\text{poly}(t)$ for $t^{O(1)}$, $\text{polylog}(t)$ for $(\log t)^{O(1)}$, and $\text{negl}(t)$ for a nonnegative function smaller than t^{-c} for every constant $c > 0$ and all sufficiently large t . For a vector $\mathbf{x}$, $\text{Supp}(\mathbf{x})$ is the set of indices of its non-zero entries, $\text{wt}(\mathbf{x}) := |\text{Supp}(\mathbf{x})|$ is its Hamming weight, and $\mathbf{x}^\top$ denotes its transpose.

Hypergraphs. A d -uniform hypergraph $\mathcal{H} = (V, E)$ consists of a vertex set V and an indexed family E of d -element subsets of V , called hyperedges. We allow repeated hyperedges and count them with multiplicity.

2.1 SNARGs with Input Preprocessing

Let $L := \{x : \exists w \text{ such that } \mathcal{R}(x, w) = 1\}$ be an **NP** language, where $\mathcal{R}$ is the **NP** relation.

Definition 2.1 (Subexponentially sound SNARG with input preprocessing). *A SNARG with input preprocessing for language $L \in \text{NP}$ consists of a tuple of algorithms*

$$\text{SNARG} = (\text{Setup}, \text{Digest}, \text{Prove}, \text{Verify})$$

with the following syntax.

- **Setup**($1^\kappa, 1^n$) $\rightarrow$ **crs**: On input the security parameter κ and an instance-length bound n , the setup algorithm outputs a common reference string **crs**.

- $\text{Digest}(\text{crs}, x) \rightarrow \mathbf{d}_x$: On input crs and an instance $x \in \{0, 1\}^n$, the deterministic input-preprocessing algorithm outputs a public digest $\mathbf{d}_x$. We assume without loss of generality that Digest is deterministic, since its randomness can be included in crs .
- $\text{Prove}(\text{crs}, x, w) \rightarrow \pi$: On input crs and an instance-witness pair (x, w) , the prover outputs a proof π .
- $\text{Verify}(1^\kappa, \mathbf{d}_x, \pi) \rightarrow \{0, 1\}$: The online verifier receives only the digest $\mathbf{d}_x$ and the proof π , and it decides to accept or reject.

The algorithms satisfy the following properties.

- **Perfect Completeness.** For every κ, n and every (x, w) satisfying $\mathcal{R}(x, w) = 1$,

$$\Pr \left[\text{Verify}(1^\kappa, \mathbf{d}_x, \pi) = 1 \mid \begin{array}{l} \text{crs} \leftarrow \text{Setup}(1^\kappa, 1^n), \\ \mathbf{d}_x := \text{Digest}(\text{crs}, x), \\ \pi \leftarrow \text{Prove}(\text{crs}, x, w) \end{array} \right] = 1.$$

- **(T, ε) -soundness.** For every possibly non-uniform adversary $\mathcal{A}$ of running time at most $T(\kappa)$, every $n \leq T(\kappa)$, and every fixed $x \in \{0, 1\}^n \setminus L$,

$$\Pr \left[\text{Verify}(1^\kappa, \mathbf{d}_x, \pi) = 1 \mid \begin{array}{l} \text{crs} \leftarrow \text{Setup}(1^\kappa, 1^n), \\ \mathbf{d}_x := \text{Digest}(\text{crs}, x), \\ \pi \leftarrow \mathcal{A}(1^\kappa, \text{crs}, x, \mathbf{d}_x) \end{array} \right] \leq \varepsilon(\kappa).$$

We say that **SNARG** is subexponentially sound if there exists a constant $\alpha > 0$ such that it is (T, ε) -sound for

$$T(\kappa) = 2^{\kappa^\alpha} \quad \text{and} \quad \varepsilon(\kappa) = 2^{-\kappa^\alpha}.$$

The soundness notion above is non-adaptive, in that x is fixed before crs is sampled. Non-adaptive soundness is sufficient for our construction.

2.2 Generic Group Model

The following definition of the generic group model is taken from [Zim15, KLM⁺18, BLOW20]. The adversary is only given the public parameter pp and does not have the ability to use sk to generate an encoding.

Definition 2.2 (Generic group oracle). A generic group oracle $\mathcal{G}$ is a stateful oracle that supports the interfaces GGM.Setup , GGM.Encode , GGM.Add , and GGM.Test , defined as follows.

- $\text{GGM.Setup}(1^\lambda)$: The oracle samples

$$\text{pp}, \text{sk} \xleftarrow{\$} \{0, 1\}^\lambda$$

and a prime p . It initializes an empty table T , stores $(\text{pp}, \text{sk}, p, T)$, and returns $(\text{pp}, \text{sk}, p)$. Any subsequent setup query returns $\perp$.

- **GGM.Encode**(k, x): For $k \in \{0, 1\}^\lambda$ and $x \in \mathbb{F}_p$, the oracle first checks that $k = \mathbf{sk}$. If not, it returns $\perp$. Otherwise, it samples a fresh handle $h \xleftarrow{\$} \{0, 1\}^\lambda$, records $T[h] = x$, and returns h .
- **GGM.Add**(k, h_1, h_2): The oracle checks that $k = \mathbf{pp}$ and that $T[h_1] = x_1$ and $T[h_2] = x_2$ are defined. If any check fails, it returns $\perp$. Otherwise, it samples a fresh handle $h \xleftarrow{\$} \{0, 1\}^\lambda$, records

$$T[h] = x_1 + x_2 \in \mathbb{F}_p,$$

and returns h . Repeated oracle calls also implement scalar multiplication by arbitrary elements of $\mathbb{F}_p$.

- **GGM.Test**(k, h): The oracle checks that $k = \mathbf{pp}$ and that $T[h] = x$ is defined. If either check fails, it returns $\perp$. Otherwise, it returns **zero** if $x = 0$ and **nonzero** otherwise.

3 Tiny Extractable Witness Encryption for Polylog-Size Circuits

In this section, we construct an extractable witness encryption scheme for SAT instances of size $\text{polylog}(\lambda)$ (Theorem 3.5). In Section 3.1, we recall BLOW's witness encryption for GapMDC [BIOW20] and prove the extractability guarantee for this gap problem. In Section 3.2, we combine this scheme with our SAT-to-GapMDC reduction to obtain tiny extractable witness encryption.

3.1 BLOW Witness Encryption for GapMDC

In this subsection, we prove that BLOW witness encryption for GapMDC is extractable in the generic group model. The construction of witness encryption for β -GapMDC is essentially the same as that of [BIOW20]. The only difference is that in this work, we further need its extractability.

Prior work [FNV17] constructs *extractable* witness encryption from predictable arguments of knowledge, while BLOW's witness encryption is indeed from predictable arguments. For GapMDC, however, our extractor only outputs a codeword of weight below the NO CASE threshold $\beta \cdot d$, a setting their result does not directly cover. We therefore prove extractability directly for BLOW's witness encryption in the GGM.

Theorem 3.1 (Extractable Witness Encryption for GapMDC). *Let $\epsilon \in (0, 1)$ be a small constant, $\beta = \omega(\log \lambda)$, and let $p = \lambda^{\omega(1)}$ be prime. Then there exists a witness encryption scheme for β -GapMDC over $\mathbb{F}_p$ with correctness probability at least $1 - \epsilon - 1/p$. Moreover, there exists a straight-line PPT extractor $\text{Ext}_\beta^{(\cdot)}(\cdot)$ such that the following holds.*

For every family of distributions $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$ where

$$(\mathbf{aux}, \mathbf{H} \in \mathbb{F}_p^{N \times K}, d) \sim \mathcal{D}_\lambda,$$

and every PPT adversary $\mathcal{A}$ that distinguishes encryptions of 0 and 1 under the $(\mathbf{H}, d)$ sampled from $\mathcal{D}_\lambda$ with advantage $\delta = \delta(\lambda)$, the extractor $\text{Ext}_\beta^{\mathcal{A}}$ takes $(\mathbf{aux}, \mathbf{H}, d)$ as input, and outputs, with probability at least $\delta/2 - \text{negl}(\lambda)$, a non-zero codeword $\mathbf{c} \in \mathbb{F}_p^{N \times 1}$ satisfying

$$\mathbf{c}^\top \mathbf{H} = \mathbf{0} \quad \text{and} \quad \text{wt}(\mathbf{c}) < \beta \cdot d.$$

Thus, the extractor outputs a witness violating the NO CASE. In particular, the scheme is semantically secure on every promised NO CASE instance.

We first recall the construction of witness encryption in BLOW, prove its correctness in Lemma 3.2, and prove its extractability in Lemma 3.3.

BLOW Construction. Let $\beta = \omega(\log \lambda)$, and fix a constant $\epsilon \in (0, 1)$, such as $\epsilon = 1/10$.

- Enc($1^\lambda, (\mathbf{H}, d), m$): Given a β -GapMDC instance $(\mathbf{H}, d)$ over $\mathbb{F}_p$, where $\mathbf{H} \in \mathbb{F}_p^{N \times K}$, and a message $m \in \{0, 1\}$, sample a cyclic group $\mathbb{G} = \langle g \rangle$ of prime order $p = \lambda^{\omega(1)}$. Then sample

$$\mathbf{r} \leftarrow \mathbb{F}_p^{K \times 1}, \quad s \leftarrow \mathbb{F}_p, \quad \mathbf{t} \leftarrow \mathbb{F}_p^{N \times 1}, \quad \mathbf{e} \leftarrow \mathcal{D}_{\epsilon/d}^{N \times 1},$$

where each coordinate of $\mathbf{e}$ is independently uniform over $\mathbb{F}_p$ with probability ϵ/d and is 0 otherwise. Set

$$\mathbf{z} = \mathbf{H}\mathbf{r} + s\mathbf{t} + \mathbf{e}$$

and output

$$\text{ct} = (\mathbb{G}, p, g, \mathbf{t}, g^{\mathbf{z}}, g^{s+m}),$$

where $g^{\mathbf{z}} := (g^{z_1}, \dots, g^{z_N})$.

- Dec(ct, $\mathbf{c}$): Given a witness $\mathbf{c} = (c_1, \dots, c_N)^T \in \mathbb{F}_p^{N \times 1}$ satisfying $\mathbf{c}^T \mathbf{H} = \mathbf{0}$ and $0 < \text{wt}(\mathbf{c}) \leq d$, parse

$$\text{ct} = (\mathbb{G}, p, g, \mathbf{t}, g^{\mathbf{z}}, C^*)$$

and compute $a := \mathbf{c}^T \mathbf{t}$. If $a = 0$, output $\perp$. Otherwise, use $g^{\mathbf{z}}$ to compute

$$Y := \prod_{i=1}^N (g^{z_i})^{c_i} = g^{\mathbf{c}^T \mathbf{z}} = g^{sa + \mathbf{c}^T \mathbf{e}}$$

and set $S := Y^{a^{-1}}$. Finally, compute $D := C^* S^{-1}$. Output 0 if D is the identity element of $\mathbb{G}$, output 1 if $D = g$, and output $\perp$ otherwise.

Lemma 3.2 (Correctness). *For every witness $\mathbf{c} \in \mathbb{F}_p^{N \times 1}$ satisfying $\mathbf{c}^T \mathbf{H} = \mathbf{0}$ and $0 < \text{wt}(\mathbf{c}) \leq d$, and every $m \in \{0, 1\}$,*

$$\Pr \left[\text{Dec}(\text{Enc}(1^\lambda, (\mathbf{H}, d), m), \mathbf{c}) = m \right] \geq 1 - \epsilon - 1/p,$$

where the randomness is over the random coins of Enc.

Proof. Since $\mathbf{c} \neq \mathbf{0}$ and $\mathbf{t}$ is uniform over $\mathbb{F}_p^N$, the value $a = \mathbf{c}^T \mathbf{t}$ is uniform over $\mathbb{F}_p$ and hence non-zero except with probability $1/p$. Moreover, because $\text{wt}(\mathbf{c}) \leq d$, the support of $\mathbf{e}$ is disjoint from that of $\mathbf{c}$ with probability at least $1 - (\epsilon/d) \cdot \text{wt}(\mathbf{c}) \geq 1 - \epsilon$. On this event, $\mathbf{c}^T \mathbf{e} = 0$. Therefore,

$$Y = g^{\mathbf{c}^T \mathbf{z}} = g^{s\mathbf{c}^T \mathbf{t}} = g^{sa},$$

so $S = Y^{a^{-1}} = g^s$ and $D = g^{s+m} S^{-1} = g^m$. The decryption algorithm thus outputs m . The claimed bound follows by a union bound. $\square$

Lemma 3.3 (Extractability). *Let $\mathcal{A}$ be a PPT adversary making at most Q_{enc} , Q_{add} , and Q_{test} queries to GGM.Encode , GGM.Add , and GGM.Test , respectively. Let $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$ be a family of distributions, where each*

$$(\text{aux}, \mathbf{H}, d) \sim \mathcal{D}_\lambda$$

samples a parity-check matrix $\mathbf{H} \in \mathbb{F}_p^{N \times K}$ for a linear code of block length $N = N(\lambda)$ over the prime-order field $\mathbb{F}_p$ and a distance $d \geq 1$, together with an auxiliary input aux . Define the distinguishing advantage as

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{WE}}(\mathcal{D}_\lambda) := & \left| \Pr[(\text{aux}, (\mathbf{H}, d)) \leftarrow \mathcal{D}_\lambda : \mathcal{A}^{\text{GGM}}(\text{aux}, \text{Enc}(1^\lambda, (\mathbf{H}, d), 0)) = 1] \right. \\ & \left. - \Pr[(\text{aux}, (\mathbf{H}, d)) \leftarrow \mathcal{D}_\lambda : \mathcal{A}^{\text{GGM}}(\text{aux}, \text{Enc}(1^\lambda, (\mathbf{H}, d), 1)) = 1] \right|. \end{aligned}$$

Then there exists a straight-line PPT extractor Ext_β^A such that

$$\begin{aligned} & \Pr[(\text{aux}, \mathbf{H}, d) \sim \mathcal{D}_\lambda, \alpha \leftarrow \text{Ext}_\beta^A(\text{aux}, \mathbf{H}, d) : \alpha \neq \mathbf{0}, \alpha^\top \mathbf{H} = \mathbf{0}, \text{wt}(\alpha) < \beta d] \\ & \geq \frac{\text{Adv}_{\mathcal{A}}^{\text{WE}}(\mathcal{D}_\lambda)}{2} - Q_{\text{test}} \cdot \left(\exp(-\epsilon\beta) + \frac{3}{p} \right) - \frac{Q_{\text{enc}}}{2^\lambda} - \frac{\binom{N+Q_{\text{add}}+4}{2}}{2^\lambda}. \end{aligned}$$

In particular, when $Q_{\text{enc}}, Q_{\text{add}}, Q_{\text{test}}$ are all bounded above by a polynomial in λ and $\beta = \omega(\log \lambda), p = \lambda^{\omega(1)}$, the right-hand side of the inequality above is

$$\text{Adv}_{\mathcal{A}}^{\text{WE}}(\mathcal{D}_\lambda)/2 - \text{negl}(\lambda).$$

Remark 3.4. *The extractor above produces a non-zero codeword of weight less than βd , thereby violating the NO CASE; it does not necessarily produce a GapMDC witness of weight at most d . When $(\mathbf{H}, d)$ is generated by a Levin reduction (see Theorem 1.3), the witness-recovery algorithm recovers a satisfying assignment from every such codeword, composing that algorithm with Ext_β^A yields an extractor for the original NP witness.*

Proof of Lemma 3.3. We use the fresh-handle generic group oracle defined in Section 2.2. In this idealized model, the adversary $\mathcal{A}$ receives the public group identifier pp , while the secret encoding key sk remains hidden from $\mathcal{A}$. Let $(\text{aux}, \mathbf{H}, d) \sim \mathcal{D}_\lambda$ where $\mathbf{H} \in \mathbb{F}_p^{N \times K}$. Fix any $m \in \{0, 1\}$, and consider the following hybrids.

- **Hybrid₀^m:** This is the original experiment, in which $\mathcal{A}$ is given aux and the ciphertext $\text{Enc}(1^\lambda, (\mathbf{H}, d), m)$ and access to the GGM oracles.
- **Hybrid₁^m:** This hybrid is identical to the previous one, except that every query by $\mathcal{A}$ to either GGM.Setup or GGM.Encode is answered with $\perp$.

Since setup has already been invoked during encryption, every subsequent setup query returns $\perp$. Moreover, an encoding query succeeds only if $\mathcal{A}$ guesses sk , which occurs with probability at most $Q_{\text{enc}}/2^\lambda$. We also condition on the absence of collisions among pp , sk , the $N+2$ initial handles in the ciphertext ct , and the handles returned by at most Q_{add} addition queries. By a union bound, the probability of such a collision is at most

$$\frac{\binom{N+Q_{\text{add}}+4}{2}}{2^\lambda}.$$

Thus, the statistical distance between Hybrid_0^m and Hybrid_1^m is at most

$$\frac{Q_{\text{enc}}}{2^\lambda} + \frac{\binom{N+Q_{\text{add}}+4}{2}}{2^\lambda}.$$

From this hybrid onward, we may assume without loss of generality that $\mathcal{A}$ queries only GGM.Add and GGM.Test .

- Hybrid_2^m : In this hybrid, we maintain a “trace table”

$$\text{tr} : \{0, 1\}^\lambda \longrightarrow \mathbb{F}_p \times \mathbb{F}_p^N \times \mathbb{F}_p.$$

For each valid handle, the table records the coefficients that express its exponent as a linear combination of the initial encodings in the ciphertext.

Initially, tr is defined only on the GGM encodings contained in the ciphertext:

$$\text{tr}(g) = (1, \mathbf{0}, 0), \quad \text{tr}(g^{z_i}) = (0, \mathbf{e}_i, 0), \quad \text{tr}(C^\star) = (0, \mathbf{0}, 1),$$

where $\mathbf{e}_i$ is the i -th standard basis vector.

Whenever an accepted query (k, h_1, h_2) to GGM.Add returns a handle h , define

$$\text{tr}(h) = \text{tr}(h_1) + \text{tr}(h_2).$$

Thus, if $\text{tr}(h) = (\delta, \boldsymbol{\alpha}, \gamma)$, then the exponent represented by h is

$$\begin{aligned} F_m(\delta, \boldsymbol{\alpha}, \gamma) &= \delta + \boldsymbol{\alpha}^\top \mathbf{z} + \gamma(s + m) \\ &= \boldsymbol{\alpha}^\top \mathbf{H}\mathbf{r} + (\boldsymbol{\alpha}^\top \mathbf{t} + \gamma)s + \boldsymbol{\alpha}^\top \mathbf{e} + \delta + \gamma m. \end{aligned}$$

This hybrid is identically distributed to the previous one because the trace table is only internal bookkeeping and does not change any oracle output.

- Hybrid_3^m : This hybrid is identical to the previous one, except that GGM.Test is replaced by the following symbolic zero-test oracle.

Symbolic Zero-Test Oracle

- On input (k, h) , the oracle checks whether $k = \text{pp}$ and whether $\text{tr}(h)$ is defined. If either check fails, it returns $\perp$.
- It returns **zero** if $\text{tr}(h) = \mathbf{0}$ and **nonzero** otherwise.

The view of $\mathcal{A}$ in Hybrid_3^0 is identically distributed to its view in Hybrid_3^1 : all initial handles and all handles returned by GGM.Add are fresh uniform strings, and the symbolic zero-test oracle depends only on their formal traces. Hence, the distinguishing advantage of $\mathcal{A}$ in this hybrid is zero.

The difference between the previous hybrid and this one is characterized by the following event:

“CATCH_m: In the query history of $\mathcal{A}$, there is a query (k, h) to GGM.Test such that $k = \text{pp}$, $\text{tr}(h)$ is defined, $\text{tr}(h) \neq \mathbf{0}$, but $F_m(\text{tr}(h)) = 0$.”

Extractor Ext_β^A . The extractor Ext_β^A takes $(\text{aux}, \mathbf{H}, d)$ as input and does the following:

- Chooses $m \leftarrow \{0, 1\}$ uniformly, and encrypt $\text{ct} \leftarrow \text{Enc}(1^\lambda, (\mathbf{H}, d), m)$ with fresh randomness.
- Runs $\mathcal{A}(\text{aux}, \text{ct})$ once against the symbolic zero-test oracle of Hybrid_3^m . For every query (k, h) to GGM.Test ,
 - If CATCH_m occurs, it takes the trace $\text{tr}(h) = (\delta, \alpha, \gamma)$, outputs α and halt immediately;
 - Otherwise, if no CATCH_m occurs during the entire execution, output $\perp$.

The extractor is straight-line and runs in polynomial time for any polynomial-time adversary $\mathcal{A}$.

Analyzing the Extraction Probability. In the hybrid Hybrid_3^m , we call a CATCH_m event *good* if its trace $\text{tr}(h) = (\delta, \alpha, \gamma) \neq \mathbf{0}$ satisfies

$$\alpha \neq \mathbf{0}, \quad \alpha^\top \mathbf{H} = \mathbf{0}, \quad \text{wt}(\alpha) < \beta d.$$

All other CATCH_m events are *bad*. Now we upper bound the probability of a bad CATCH_m event.

We enumerate the zero-test queries in the symbolic execution as $1, \dots, Q_{\text{test}}$. For each j , let

$$T_j = (\delta_j, \alpha_j, \gamma_j)$$

denote the trace submitted in the j -th zero-test query, and let $\text{BadCatch}_{m,j}$ be the event that this query causes a bad CATCH_m . If the j -th valid zero-test query does not exist, define $\text{BadCatch}_{m,j}$ to be false.

Fix j . Condition on the sampled tuple $(\text{aux}, \mathbf{H}, d)$, the public vector $\mathbf{t}$, the random tape of $\mathcal{A}$, and the entire symbolic transcript up to and including the submission of the j -th zero-test query, but before privately evaluating $F_m(T_j)$. In Hybrid_3^m , this conditioned information is independent of $\mathbf{r}, s, \mathbf{e}$, because the symbolic oracle answers depend only on formal traces. In particular, T_j is now fixed while $\mathbf{r}, s, \mathbf{e}$ retain their original independent distributions.

Write $T_j = (\delta, \alpha, \gamma)$. If $\alpha^\top \mathbf{H} \neq \mathbf{0}$, uniformity of $\mathbf{r}$ implies

$$\Pr_{\mathbf{r}}[F_m(T_j) = 0] \leq \frac{1}{p}.$$

If $\alpha^\top \mathbf{H} = \mathbf{0}$ but $\alpha^\top \mathbf{t} + \gamma \neq 0$, uniformity of s gives the same bound. Otherwise,

$$F_m(T_j) = \alpha^\top \mathbf{e} + \delta + \gamma m.$$

If $\alpha = \mathbf{0}$, then $\gamma = 0$ because $\alpha^\top \mathbf{t} + \gamma = 0$; since $T_j \neq \mathbf{0}$, we have $\delta \neq 0$, so this query cannot be a catch. If $\alpha \neq \mathbf{0}$ and the query is a bad catch, then $\text{wt}(\alpha) \geq \beta d$. In this case we have, for any $q \in \mathbb{F}_p$,

$$\Pr[\mathbf{e} \leftarrow \mathcal{D}_{\epsilon/d}^{N \times 1} : \alpha^\top \mathbf{e} = q] \leq \left(1 - \frac{\epsilon}{d}\right)^{\beta d} + \frac{1}{p} \leq \exp(-\epsilon\beta) + \frac{1}{p}$$

By the union bound, we have

$$\Pr[\text{BadCatch}_{m,j}] \leq \exp(-\epsilon\beta) + \frac{3}{p}.$$

This bound holds for every conditioned symbolic transcript and every sampled $(\mathbf{aux}, \mathbf{H}, d)$; averaging over the conditioning therefore preserves it.

Let J be the index of the first query that causes CATCH_m . Although J may depend on $\mathbf{r}, s, \mathbf{e}$, the event that the first catch is bad is contained in the event that some query is a bad catch. Hence,

$$\begin{aligned} \Pr[J \text{ exists and the } J\text{-th catch is bad}] &\leq \Pr \left[\bigcup_{j=1}^{Q_{\text{test}}} \text{BadCatch}_{m,j} \right] \\ &\leq Q_{\text{test}} \cdot \left(\exp(-\epsilon\beta) + \frac{3}{p} \right). \end{aligned}$$

Let G_m be the probability that the first CATCH_m is good for m . The executions of Hybrid_2^m and Hybrid_3^m can be coupled so that they remain identical until the first catch. Combining this coupling with the bounds for Hybrid_0^m and Hybrid_1^m gives

$$\text{Adv}_{\mathcal{A}}^{\text{WE}}(\mathcal{D}_\lambda) \leq G_0 + G_1 + 2Q_{\text{test}} \cdot \left(\exp(-\epsilon\beta) + \frac{3}{p} \right) + \frac{2Q_{\text{enc}}}{2^\lambda} + \frac{2^{(N+Q_{\text{add}}+4)}}{2^\lambda}.$$

Because the extractor chooses m uniformly, its success probability is $(G_0 + G_1)/2$. Rearranging the preceding inequality yields exactly the bound in the lemma statement. $\square$

3.2 Tiny Witness Encryption

In this subsection, we prove the following theorem.

Theorem 3.5 (Witness Encryption for Polylog-Size Circuits, Formal Version of Theorem 1.2). *For every constant $A > 0$ and any fixed $\epsilon \in (0, 1)$, there exists a witness encryption scheme $\text{tWE} = (\text{Enc}, \text{Dec})$ with decryption correctness probability at least $1 - \epsilon - \text{negl}(\lambda)$ and a straight-line PPT extractor $\text{Ext}^{(\cdot)}(\cdot)$ such that, for every $\mathcal{D} = \{\mathcal{D}_\lambda\}_\lambda$ that is a family of distributions where each $\mathcal{D}_\lambda$ is a distribution over an auxiliary information $\mathbf{aux}$ and a circuit C :*

$$(\mathbf{aux}, C) \sim \mathcal{D}_\lambda, \text{ where } |C| \leq (\log \lambda)^A,$$

and for every PPT adversary $\mathcal{A}$ that distinguishes

$$(\mathbf{aux}, \text{Enc}(1^\lambda, C, 0)) \text{ and } (\mathbf{aux}, \text{Enc}(1^\lambda, C, 1)),$$

with advantage $\delta = \delta(\lambda)$, we have that $\text{Ext}^{\mathcal{A}}(\mathbf{aux}, C)$ outputs a satisfying assignment for C with probability at least $\delta/2 - \text{negl}(\lambda)$.

Proof. We compose the reduction in Theorem 5.1 with BLOW witness encryption. Let $\text{Red}_{\text{SAT} \rightarrow \text{MDC}}$ be the instance map, $\text{Map}_{\text{SAT} \rightarrow \text{MDC}}$ be the witness map and $\text{Ext}_{\text{SAT} \rightarrow \text{MDC}}$ be the witness recovery algorithm in Theorem 5.1.

Construction. Set $\beta = \lceil (\log_2 \lambda)^2 \rceil$. To encrypt m under C , choose a prime $2^\lambda < p < 2^{\lambda+1}$ and compute

$$(\mathbf{H}, d) \leftarrow \text{Red}_{\text{SAT} \rightarrow \text{MDC}}(C; \beta, p).$$

Run BLOW encryption on $(\mathbf{H}, d)$ and output $(C, \text{ct}_{\text{BLOW}})$. Given a satisfying assignment w for C , the decryptor computes $\mathbf{c} \leftarrow \text{Map}_{\text{SAT} \rightarrow \text{MDC}}(C, w; \beta, p)$ and uses $\mathbf{c}$ to decrypt the BLOW ciphertext.

Efficiency and correctness. The final guarantee of Theorem 5.1 gives polynomial running time in λ for the reduction and both witness algorithms when $|C| \leq (\log \lambda)^4$. Since $\log p = O(\lambda)$, BLOW encryption and decryption also run in polynomial time. Prime sampling can be implemented in polynomial time with negligible failure probability. The witness map produces a non-zero codeword of weight at most d , so Lemma 3.2 gives correctness probability at least $1 - \epsilon - \text{negl}(\lambda)$.

Extractability. Run the BLOW extractor from Lemma 3.3, including C in the auxiliary input and presenting its ciphertext to $\mathcal{A}$ as $(C, \text{ct}_{\text{BLOW}})$. Since $\beta = \omega(\log \lambda)$ and $p = \lambda^{\omega(1)}$, the extractor outputs, with probability at least $\delta/2 - \text{negl}(\lambda)$, a non-zero codeword $\mathbf{c}$ satisfying

$$\mathbf{H}^\top \mathbf{c} = \mathbf{0}, \quad \text{wt}(\mathbf{c}) < \beta d.$$

The witness-recovery guarantee of Theorem 5.1 therefore gives a satisfying assignment

$$w \leftarrow \text{Ext}_{\text{SAT} \rightarrow \text{MDC}}(C, \mathbf{c}; \beta, p).$$

This deterministic polynomial-time postprocessing preserves both the straight-line property and the extraction probability. On an unsatisfiable circuit, such an assignment cannot exist, so the distinguishing advantage of $\mathcal{A}$ is negligible. $\square$

4 Boosting Tiny Witness Encryption to Full NP

We now prove the main theorem of this work.

Theorem 4.1 (Witness Encryption for NP, Formal Version of Theorem 1.1). *Let L be an NP language, and let $\mathcal{R}$ be its NP relation. Suppose there exists a SNARG*

$$\text{SNARG} = (\text{Setup}, \text{Digest}, \text{Prove}, \text{Verify})$$

with input preprocessing for L as in Definition 2.1, with

- **Input Succinctness.** *There exists a fixed polynomial $q(\cdot, \cdot)$ such that, for every $x \in \{0, 1\}^n$, let $\mathbf{d}_x = \text{Digest}(\text{crs}, x)$ be the digest of x , and let $C_{\kappa, \mathbf{d}_x}$ as the following Boolean circuit with $\kappa, \mathbf{d}_x$ hardwired in it:*

$$C_{\kappa, \mathbf{d}_x}(\pi) := \text{Verify}(1^\kappa, \mathbf{d}_x, \pi),$$

then

$$\text{size}(C_{\kappa, \mathbf{d}_x}) \leq q(\kappa, \log n).$$

The algorithms Setup, Digest, and Prove run in time polynomial in κ, n , and the witness length.

- **Subexponential Soundness.** *There exists a constant $\alpha > 0$ such that the SNARG achieves $(2^{\kappa^\alpha}, 2^{-\kappa^\alpha})$ -non-adaptive soundness. Namely, fix any $x \notin L$ of length $n \leq 2^{\kappa^\alpha}$ before sampling $\text{crs} \leftarrow \text{Setup}(1^\kappa, 1^n)$, and let $\mathbf{d}_x = \text{Digest}(\text{crs}, x)$. Any non-uniform adversary running in time at most 2^{κ^α} , given $(1^\kappa, \text{crs}, x, \mathbf{d}_x)$, produces a proof π satisfying $\text{Verify}(1^\kappa, \mathbf{d}_x, \pi) = 1$ with probability at most $2^{-\kappa^\alpha}$.*

Then for every $\epsilon \in (0, 1)$, there exists a witness encryption scheme for L in the generic group model, with decryption correctness probability at least $1 - \epsilon - \text{negl}(\lambda)$.

Construction. Let λ be the security parameter of the witness encryption scheme and set the security parameter of SNARG to

$$\kappa = \kappa(\lambda) := \left\lceil (\log \lambda)^{2/\alpha} \right\rceil.$$

Our construction needs the extractable witness encryption $\text{tWE} = (\text{tWE.Enc}, \text{tWE.Dec})$ in Theorem 3.5 as an ingredient.

- $\text{Enc}(1^\lambda, x, m)$: Given an instance $x \in \{0, 1\}^n$ and a message m , generate crs for SNARG and a digest for x :

$$\text{crs} \leftarrow \text{Setup}(1^\kappa, 1^n), \quad \text{d}_x := \text{Digest}(\text{crs}, x),$$

and define the circuit

$$C_{\kappa, \text{d}_x}(\pi) := \text{Verify}(1^\kappa, \text{d}_x, \pi).$$

By input succinctness, the size of C_{κ, d_x} is $\text{polylog}(\lambda)$. Encrypt a message m under C_{κ, d_x} using tWE :

$$\text{ct}_{\text{tiny}} \leftarrow \text{tWE.Enc}(1^\lambda, C_{\kappa, \text{d}_x}, m).$$

The resulting ciphertext is

$$\text{ct} = (x, \text{crs}, \text{d}_x, \text{ct}_{\text{tiny}}).$$

- $\text{Dec}(\text{ct}, w)$: Given a witness w for x , the decryptor computes

$$\pi \leftarrow \text{Prove}(\text{crs}, x, w)$$

and uses π to decrypt ct_{tiny} :

$$m' \leftarrow \text{tWE.Dec}(\text{ct}_{\text{tiny}}, \pi).$$

Finally, it outputs m' as the decrypted message.

Proof of Theorem 4.1. The correctness of the above construction follows directly from the correctness of tWE and the completeness of the SNARG. Next, we prove the semantic security for false instances.

For every false instance $x \notin L$ and every PPT adversary $\mathcal{A}$, suppose $\mathcal{A}$ distinguishes between the ciphertext of 0 and the ciphertext of 1 with advantage $\delta = \delta(\lambda)$ in GGM, we construct a cheating prover $\mathcal{P}^*$ breaking the soundness of the underlying SNARG.

Given the crs from the challenger, $\mathcal{P}^*$ runs the deterministic Digest algorithm to obtain d_x , then $\mathcal{P}^*$ runs $\text{Ext}^{\mathcal{A}}(\text{aux}, C_{\kappa, \text{d}_x})$ by simulating all the GGM oracles and setting $\text{aux} = (x, \text{crs}, \text{d}_x)$. Let π^* be the extracted witness. Then $\mathcal{P}^*$ outputs π^* as the cheating SNARG proof.

By Theorem 3.5, we have that

$$\text{Verify}(1^\kappa, \text{d}_x, \pi^*) = C_{\kappa, \text{d}_x}(\pi^*) = 1,$$

with probability at least $\delta(\lambda)/2 - \text{negl}(\lambda)$. By the subexponential soundness of the SNARG (Definition 2.1), we have that

$$\delta(\lambda)/2 - \text{negl}(\lambda) \leq 2^{-\kappa^\alpha}.$$

Since we choose $\kappa \geq (\log \lambda)^{2/\alpha}$, the right-hand side is still negligible. Thus $\delta(\lambda)$ is also negligible. $\square$

5 Reduction From SAT to GapMDC

In this section, we prove the following theorem.

Theorem 5.1 (Karp–Levin reduction from SAT to GapMDC over large prime fields, Formal Version of Theorem 1.3). *There exists a deterministic algorithm $\text{Red}_{\text{SAT} \rightarrow \text{MDC}}$ and a function $T = T(n)$ where*

$$T(n) = \exp\left(2^{\sqrt{\log n}(\log \log n)^{O(1)}}\right)$$

such that the following holds. For every sufficiently large n , every SAT instance φ of size at most n , every integer $\beta \geq 2$, and every prime $p > 2^{T(n)}$, the algorithm takes as input φ, β, p and outputs

$$\text{Red}_{\text{SAT} \rightarrow \text{MDC}}(\varphi; \beta, p) = (\mathbf{H}, d), \quad \mathbf{H} \in \mathbb{F}_p^{N \times K},$$

where $1 \leq d \leq N/\beta$, $K \leq N$, and $N \leq T(n)^{O(\log \beta)}$. If φ is satisfiable, the linear code defined by the parity-check matrix $\mathbf{H}$ contains a non-zero codeword of weight at most d . If φ is unsatisfiable, every non-zero codeword has weight at least βd . Moreover, the reduction is a Levin reduction:

- **Witness map for completeness.** *There exists a deterministic algorithm $\text{Map}_{\text{SAT} \rightarrow \text{MDC}}$ that, given φ and any satisfying assignment w , outputs $\mathbf{c} \leftarrow \text{Map}_{\text{SAT} \rightarrow \text{MDC}}(\varphi, w; \beta, p)$ satisfying*

$$\mathbf{c} \in \mathbb{F}_p^{N \times 1} \setminus \{\mathbf{0}\}, \quad \mathbf{H}^T \mathbf{c} = \mathbf{0}, \quad \text{wt}(\mathbf{c}) \leq d.$$

- **Witness recovery for soundness.** *There exists a deterministic algorithm $\text{Ext}_{\text{SAT} \rightarrow \text{MDC}}$ that, given φ and any non-zero $\mathbf{c} \in \mathbb{F}_p^{N \times 1}$ satisfying*

$$\mathbf{H}^T \mathbf{c} = \mathbf{0}, \quad \text{wt}(\mathbf{c}) < \beta d,$$

outputs a satisfying assignment $w \leftarrow \text{Ext}_{\text{SAT} \rightarrow \text{MDC}}(\varphi, \mathbf{c}; \beta, p)$ for φ .

All three algorithms run in time $T(n)^{O(\log \beta)} \text{poly}(\log p)$.

In particular, for every fixed $A > 0$ and all sufficiently large λ , they run in polynomial time in λ on circuits of size at most $(\log \lambda)^A$ when

$$\beta = \lceil (\log_2 \lambda)^2 \rceil, \quad 2^\lambda < p < 2^{\lambda+1},$$

and the output block length is $N = \lambda^{o(1)}$.

This section proves Theorem 5.1 assuming the following theorem, which augments the Hair–Sahai reduction with the forward witness map and witness-recovery algorithm [HS26].

For an integer matrix $\mathbf{C} \in \mathbb{Z}^{M \times N}$ and a row set $S \subseteq [M]$, define its active columns by

$$\Gamma_{\mathbf{C}}(S) := \{j \in [N] : (\mathbf{C}_S)_{*,j} \neq \mathbf{0}\},$$

where $\mathbf{C}_S$ denotes the row-induced submatrix indexed by S and $(\mathbf{C}_S)_{*,j}$ is the j -th column of $\mathbf{C}_S$.

Theorem 5.2 (Levin reduction from SAT to the Hair–Sahai integer matrix). *There exists a deterministic algorithm Red_{HS} and a function $T_{\text{HS}} = T_{\text{HS}}(n)$ where*

$$T_{\text{HS}}(n) = \exp\left(2^{\sqrt{\log n}(\log \log n)^{O(1)}}\right),$$

such that the following holds for every sufficiently large n and every SAT instance φ of size n .

The reduction algorithm outputs

$$\text{Red}_{\text{HS}}(\varphi) = (\mathbf{C}_\varphi, h_\varphi), \quad \mathbf{C}_\varphi \in \mathbb{Z}^{M_\varphi \times N_\varphi}, \quad h_\varphi \geq 1,$$

where its running time, M_φ , and N_φ are at most $T_{\text{HS}}(n)$, while $1 \leq h_\varphi \leq M_\varphi/2$ and $\|\mathbf{C}_\varphi\|_\infty \leq M_\varphi$. The reduction has the following two witness guarantees.

- **Witness map for completeness.** There exists a deterministic algorithm Map_{HS} that, given φ and any satisfying assignment a , runs in time $T_{\text{HS}}(n)$ and outputs a non-empty set $S \subseteq [M_\varphi]$ such that

$$|S| \leq h_\varphi \quad \text{and} \quad |\Gamma_{\mathbf{C}_\varphi}(S)| < |S|.$$

- **Witness recovery for soundness.** There exists a deterministic algorithm Ext_{HS} that, given φ and any non-empty $S \subseteq [M_\varphi]$ satisfying

$$|S| < 2h_\varphi \quad \text{and} \quad \text{rank}_{\mathbb{Q}}((\mathbf{C}_\varphi)_S) < |S|,$$

outputs an assignment satisfying φ in time $T_{\text{HS}}(n)$.

We defer the proof of Theorem 5.2 to Appendix A. This section derives Theorem 5.1 from Theorem 5.2: first obtain gap 2 over a sufficiently large prime field in Section 5.1, then amplify to the requested gap β in Section 5.2. Both steps preserve the witness map and witness recovery guarantees.

5.1 From Hair–Sahai Matrix to 2-GapMDC over $\mathbb{F}_p$

This subsection proves the following lemma.

Lemma 5.3 (Karp–Levin reduction from the Hair–Sahai matrix to 2-GapMDC). *Let $(\mathbf{C}_\varphi, h_\varphi)$ be the output of Theorem 5.2 on a circuit φ of size n , and let $p > M_\varphi^{3M_\varphi/2}$ be prime. There exists a deterministic algorithm $\text{Red}_{\text{HS} \rightarrow \text{MDC}}$ that takes $(\mathbf{C}_\varphi, h_\varphi)$ and p as input and outputs*

$$(\mathbf{H}_\varphi, d_\varphi), \quad \mathbf{H}_\varphi \in \mathbb{F}_p^{L_\varphi \times (L_\varphi - M_\varphi)},$$

where

$$L_\varphi = 2h_\varphi N_\varphi + M_\varphi, \quad d_\varphi = h_\varphi.$$

If φ is satisfiable, this code contains a non-zero codeword of weight at most d_φ . If φ is unsatisfiable, every non-zero codeword has weight at least $2d_\varphi$. Moreover, the reduction is a Levin reduction:

- **Witness map for completeness.** There exists a deterministic algorithm $\text{Map}_{\text{HS} \rightarrow \text{MDC}}$ that, given φ , any satisfying assignment w , and p , outputs a non-zero codeword $\mathbf{c} \in \mathbb{F}_p^{L_\varphi \times 1}$ satisfying

$$\mathbf{H}_\varphi^\top \mathbf{c} = \mathbf{0}, \quad \text{wt}(\mathbf{c}) \leq d_\varphi.$$

- **Witness recovery for soundness.** There exists a deterministic algorithm $\text{Ext}_{\text{HS} \rightarrow \text{MDC}}$ that, given φ , p , and any non-zero $\mathbf{c} \in \mathbb{F}_p^{L_\varphi \times 1}$ satisfying

$$\mathbf{H}_\varphi^\top \mathbf{c} = \mathbf{0}, \quad \text{wt}(\mathbf{c}) < 2d_\varphi,$$

outputs a satisfying assignment for φ .

All three algorithms run in time $\text{poly}(T_{\text{HS}}(n), \log p)$, where $T_{\text{HS}}(n)$ is the bound in Theorem 5.2.

Proof. Our linear code construction follows almost the same integer lattice construction in [HS26], with the only change being that we reduce the resulting integer matrix modulo p . The main technical challenge is to prove soundness: taking modulo p does not change the approximate gap.

Construction. Repeat $\mathbf{C}_\varphi$ horizontally $2h_\varphi$ times and append an identity matrix. Reduce the resulting matrix modulo p to obtain

$$\mathbf{G}_\varphi := \left[\underbrace{\mathbf{C}_\varphi \mid \cdots \mid \mathbf{C}_\varphi}_{2h_\varphi \text{ copies}} \mid \mathbf{I}_{M_\varphi} \right] \bmod p \in \mathbb{F}_p^{M_\varphi \times L_\varphi}.$$

All arithmetic in the code is over $\mathbb{F}_p$.

Compute a full-column-rank parity-check matrix $\mathbf{H}_\varphi$ by Gaussian elimination, so that

$$\mathbf{H}_\varphi^\top \mathbf{c} = \mathbf{0} \iff \mathbf{c}^\top = \mathbf{x} \mathbf{G}_\varphi \text{ for a unique } \mathbf{x} \in \mathbb{F}_p^{1 \times M_\varphi}.$$

Output $(\mathbf{H}_\varphi, h_\varphi)$.

Witness map for completeness. Given φ and w , apply Map_{HS} to obtain a non-empty set S with

$$|S| \leq h_\varphi, \quad |\Gamma_{\mathbf{C}_\varphi}(S)| < |S|.$$

The submatrix $(\mathbf{C}_\varphi)_S \bmod p$ has fewer non-zero columns than rows, so Gaussian elimination over $\mathbb{F}_p$ finds a non-zero dependence. Extending it by zeros outside S gives $\mathbf{x} \in \mathbb{F}_p^{1 \times M_\varphi}$ with $\mathbf{x} \mathbf{C}_\varphi = \mathbf{0} \pmod{p}$. Output

$$\mathbf{c} = (\mathbf{x} \mathbf{G}_\varphi)^\top = (\mathbf{0}, \dots, \mathbf{0}, \mathbf{x})^\top.$$

It is non-zero and has weight at most $|S| \leq h_\varphi$.

Witness recovery for soundness. Given $\mathbf{c}$, read $\mathbf{x}$ from its final M_φ coordinates, so that $\mathbf{c}^\top = \mathbf{x} \mathbf{G}_\varphi$. Since $\text{wt}(\mathbf{c}) < 2h_\varphi$, the repeated blocks force $\mathbf{x} \mathbf{C}_\varphi = \mathbf{0} \pmod{p}$. Therefore $\mathbf{x} \neq \mathbf{0}$ and

$$S := \text{Supp}(\mathbf{x}) \text{ satisfies } |S| = \text{wt}(\mathbf{c}) < 2h_\varphi.$$

We will show that the rows of $\mathbf{C}_\varphi$ in S are linearly dependent over $\mathbb{Q}$, so that we can apply the witness-recovery algorithm in Theorem 5.2 to recover a satisfiable assignment of φ . We prove this by contradiction. Suppose $(\mathbf{C}_\varphi)_S$ is full row-rank over $\mathbb{Q}$, then there exists a $|S| \times |S|$ submatrix Δ of $(\mathbf{C}_\varphi)_S$ that is invertible over $\mathbb{Q}$. Hadamard's inequality gives

$$|\det \Delta| \leq (\sqrt{|S|} M_\varphi)^{|S|} \leq M_\varphi^{3M_\varphi/2} < p.$$

Since $\det \Delta \in \mathbb{Z}$ is non-zero, $\det \Delta$ is also non-zero modulo p . Hence, $(\mathbf{C}_\varphi)_S$ is full row-rank over $\mathbb{F}_p$. But then we reach a contradiction, since $\mathbf{x} \mathbf{C}_\varphi = \mathbf{0} \pmod{p}$. Hence, $(\mathbf{C}_\varphi)_S$ is not full row-rank over $\mathbb{Q}$.

Apply $\text{Ext}_{\text{HS}}(\varphi, S)$ to recover a satisfying assignment. This also proves the claimed NO CASE bound.

The matrix dimensions are polynomial in $T_{\text{HS}}(n)$, and Gaussian elimination over $\mathbb{F}_p$ takes polynomial time in these dimensions and $\log p$. Together with Theorem 5.2, these bounds give the stated running times. $\square$

5.2 Amplifying the Approximation Gap

We now amplify the gap 2 achieved in Lemma 5.3 by tensoring the code and therefore prove Theorem 5.1.

Proof of Theorem 5.1. Apply Theorem 5.2 and Lemma 5.3 to φ and obtain $(\mathbf{H}_0, d_0)$. Let $\mathbf{G}_0$ be the full row-rank generator matrix of $\mathbf{H}_0$, and let N_0 be its block length. Recall that d_0 is the upper bound of the minimal weight when φ is satisfiable. Choose the universal constants in $T(n)$ large enough that

$$N_0 \leq T(n), \quad \frac{3}{2} M_\varphi \log_2 M_\varphi \leq T(n).$$

This is possible because both quantities are polynomial in $T_{\text{HS}}(n)$. The condition $p > 2^{T(n)}$ then implies $p > M_\varphi^{3M_\varphi/2}$, as required by Lemma 5.3.

Construction. Set

$$s := \lceil \log_2 \beta \rceil, \quad \mathbf{G} := \mathbf{G}_0^{\otimes s}, \quad N := N_0^s, \quad d := d_0^s.$$

These parameters satisfy $1 \leq d \leq N/\beta$. The reduction $\text{Red}_{\text{SAT} \rightarrow \text{MDC}}$ computes a full-column-rank parity-check matrix $\mathbf{H}$ for the row span of $\mathbf{G}$ by Gaussian elimination and outputs $(\mathbf{H}, d)$.

Witness map for completeness. Given w satisfying φ , apply the witness map of Lemma 5.3 to obtain a non-zero codeword $\mathbf{c}_0$ of weight at most d_0 . The algorithm $\text{Map}_{\text{SAT} \rightarrow \text{MDC}}$ outputs $\mathbf{c} = \mathbf{c}_0^{\otimes s}$, which is non-zero and satisfies

$$\mathbf{H}^\top \mathbf{c} = \mathbf{0}, \quad \text{wt}(\mathbf{c}) = \text{wt}(\mathbf{c}_0)^s \leq d_0^s = d.$$

Witness recovery for soundness. Let $\mathbf{c}$ be a non-zero codeword with $\text{wt}(\mathbf{c}) < \beta d \leq (2d_0)^s$. We show how to recover a codeword for $\mathbf{G}_0$ of weight less than $2d_0$, so that we can apply witness recovery of Lemma 5.3 to it. If $s = 1$, use $\mathbf{c}$ itself. Otherwise, view $\mathbf{c}$ as an $N_0 \times N_0^{s-1}$ matrix. Each column belongs to the base code generated by $\mathbf{G}_0$, and each row belongs to the $(s-1)$ -fold tensor code of $\mathbf{G}_0$.

If a non-zero column has weight less than $2d_0$, output it. Otherwise, any non-zero column has weight at least $2d_0$, so the matrix $\mathbf{c}$ has at least $2d_0$ non-zero rows. Since its total weight is less than $(2d_0)^s$, some non-zero row has weight less than $(2d_0)^{s-1}$. Find such a row, and recurse on the remaining $s-1$ coordinates. Once we find a codeword of $\mathbf{G}_0$ with weight less than $2d_0$, the witness-recovery algorithm of Lemma 5.3 applies. This gives $\text{Ext}_{\text{SAT} \rightarrow \text{MDC}}$ and proves the NO CASE bound.

Parameters and running time. The final block length is $N = N_0^s \leq T(n)^{O(\log \beta)}$. Constructing the tensor code and its parity-check matrix, mapping a witness, and searching the rows and columns during recovery all take polynomial time in N , s , and $\log p$, in addition to the base reduction's cost from Lemma 5.3. Together with Lemma 5.3, this gives the stated running time for all three algorithms.

For the witness-encryption specialization, take $n = \lceil (\log \lambda)^A \rceil$ and $\beta = \lceil (\log_2 \lambda)^2 \rceil$. For every fixed A ,

$$\log T(n) = 2^{O(\sqrt{\log \log \lambda} (\log \log \log \lambda)^{O(1)})} = (\log \lambda)^{o(1)}.$$

Thus $T(n) < \lambda$ for sufficiently large λ , and every prime $2^\lambda < p < 2^{\lambda+1}$ meets the field-size requirement. Moreover,

$$T(n)^{O(\log \beta)} = \exp(O(\log \log \lambda) \log T(n)) = \lambda^{o(1)}.$$

Since $\log p = O(\lambda)$, all three algorithms run in polynomial time in λ . $\square$

6 AI Disclosure

The reduction from SAT to GapMDC (Theorem 5.1) was discovered through interactions with GPT-5.6 Sol. GPT-5.6 Sol and GPT-6 Astra, including through Codex, assisted with mathematical discussion, theorem and proof drafting, literature comparisons, and checking arguments and parameters. During the writing process, we supplied AI tools with high-level proof strategies, draft theorem statements, and paragraph plans, and used iterative follow-up prompts to examine proposed arguments and request corrections. We then substantially revised most of the AI-generated text based on our own understanding of the proofs. AI tools also assisted with exploratory Lean formalization, which we chose not to include in this paper.

Independent verification. We independently verified the proofs and take full responsibility for the accuracy, attribution, and correctness of all material in the paper, including material developed with AI assistance. In particular, we independently studied the prior works underlying the reductions behind Theorem 5.1, including Hair–Sahai’s recent work on GapSVP [HS26], Khot’s reductions and quasi-random PCPs [Kho06b], the Khot–Saket generalization [KS16], and the BGLR reduction [BGLR25]. We checked the relevant definitions, constructions, parameter choices, and proof details against these sources and drafted Appendix A to explain the reduction chain and prove the required witness maps and witness-recovery guarantees for Levin reductions.

Acknowledgement

The author is supported in part by a 2025 Amazon Research Award.

References

- [Ajt98] Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *30th ACM STOC*, pages 10–19. ACM Press, May 1998. 5
- [AK14] Per Austrin and Subhash Khot. A simple deterministic reduction for the gap minimum distance of code problem. *IEEE Transactions on Information Theory*, 60(10):6636–6645, 2014. 3
- [AMYY25] Shweta Agrawal, Anuja Modi, Anshu Yadav, and Shota Yamada. Zeroizing attacks against evasive and circular evasive LWE. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part II*, volume 16269 of *LNCS*, pages 259–290. Springer, Cham, December 2025. 1
- [BCI⁺13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In Amit Sahai, editor,

- TCC 2013*, volume 7785 of *LNCS*, pages 315–333. Springer, Berlin, Heidelberg, March 2013. 2
- [BGLR25] Vijay Bhattiprolu, Venkatesan Guruswami, Euiwoong Lee, and Xuandi Ren. Inapproximability of finding sparse vectors in codes, subspaces, and lattices. In *66th FOCS*, pages 1295–1303. IEEE Computer Society Press, December 2025. 3, 7, 22, 26, 27, 28, 29, 30
- [BIJ⁺20] James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, Amit Sahai, and Mark Zhandry. Affine determinant programs: A framework for obfuscation and witness encryption. In Thomas Vidick, editor, *ITCS 2020*, volume 151, pages 82:1–82:39. LIPIcs, January 2020. 1
- [BIOW20] Ohad Barta, Yuval Ishai, Rafail Ostrovsky, and David J. Wu. On succinct arguments and witness encryption from groups. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 776–806. Springer, Cham, August 2020. 1, 2, 3, 9, 10
- [BÜW24] Chris Brzuska, Akin Ünal, and Ivy K. Y. Woo. Evasive LWE assumptions: Definitions, classes, and counterexamples. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 418–449. Springer, Singapore, December 2024. 1
- [CN98] Jin-Yi Cai and A. Nerurkar. Approximating the svp to within a factor $(1-1/\dim/\sup/\spl\epsilon/\epsilon)$ is np-hard under randomized conditions. In *Proceedings. Thirteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat. No.98CB36247)*, pages 46–55, 1998. 5
- [CW09] Qi Cheng and Daqing Wan. A deterministic reduction for the gap minimum distance problem: [extended abstract]. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 33–38. ACM Press, May / June 2009. 3
- [DJM⁺25] Nico Döttling, Abhishek Jain, Giulio Malavolta, Surya Mathialagan, and Vinod Vaikuntanathan. Simple and general counterexamples for private-coin evasive LWE. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 73–92. Springer, Cham, August 2025. 1
- [DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003. 3
- [FNV17] Antonio Faonio, Jesper Buus Nielsen, and Daniele Venturi. Predictable arguments of knowledge. In Serge Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 121–150. Springer, Berlin, Heidelberg, March 2017. 10
- [FWW23] Cody Freitag, Brent Waters, and David J. Wu. How to use (plain) witness encryption: Registered ABE, flexible broadcast, and more. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part IV*, volume 14084 of *LNCS*, pages 498–531. Springer, Cham, August 2023. 1

- [GGH⁺13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013. [1](#)
- [GGPR13] Rosario Gemaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Berlin, Heidelberg, May 2013. [2](#)
- [GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013. [1](#)
- [GHK⁺25] Sanjam Garg, Mohammad Hajiabadi, Dimitris Kolonelos, Abhiram Kothapalli, and Guru-Vamsi Policharla. A framework for witness encryption from linearly verifiable SNARKs and applications. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 504–539. Springer, Cham, August 2025. [1](#)
- [GKP⁺13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run Turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Berlin, Heidelberg, August 2013. [1](#)
- [HHY25] Tzu-Hsiang Huang, Wei-Hsiang Hung, and Shota Yamada. A note on obfuscation-based attacks on private-coin evasive LWE. Cryptology ePrint Archive, Report 2025/421, 2025. [1](#)
- [HJL25] Yao-Ching Hsieh, Aayush Jain, and Huijia Lin. Lattice-based post-quantum iO from circular security with random opening assumption. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 3–38. Springer, Cham, August 2025. [1](#)
- [HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 469–477. ACM Press, June 2007. [5](#)
- [HS26] Isaac M Hair and Amit Sahai. Deterministic hardness of approximation for svp in all finite ℓ_p norms, 2026. [3](#), [4](#), [5](#), [6](#), [7](#), [18](#), [20](#), [22](#), [26](#), [31](#), [34](#), [35](#), [36](#), [39](#), [40](#), [41](#), [44](#), [45](#), [46](#), [47](#), [48](#), [49](#), [50](#), [51](#)
- [JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021. [1](#)
- [Jo26] Sunghyeon Jo. Cryptanalysis of a candidate witness encryption scheme for affine determinant programs. Cryptology ePrint Archive, Paper 2026/1583, 2026. [1](#)
- [Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM (JACM)*, 52(5):789–808, 2005. [5](#)

- [Kho06a] Subhash Khot. Hardness of approximating the shortest vector problem in high ℓ_p norms. *Journal of Computer and System Sciences*, 72(2):206–219, 2006. 5
- [Kho06b] Subhash Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006. 5, 7, 22, 26, 36, 37, 38, 39, 40, 41, 42, 43
- [KLM⁺18] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 544–562. Springer, Cham, September 2018. 9
- [KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 254–273. Springer, Berlin, Heidelberg, December 2014. 1
- [KS16] Subhash Khot and Rishi Saket. Hardness of Bipartite Expansion. In Piotr Sankowski and Christos Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:17, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 5, 7, 22, 26, 40, 41, 42, 43
- [Mic01] Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM journal on Computing*, 30(6):2008–2035, 2001. 5
- [Mic14] Daniele Micciancio. Locally dense codes. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 90–97. IEEE, 2014. 3
- [PW04] Ruud Pellikaan and Xin-Wen Wu. List decoding of q -ary Reed–Muller codes. *IEEE Transactions on Information Theory*, 50(4):679–682, 2004. 7, 43
- [Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 535–559. Springer, Cham, August 2022. 1
- [VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 195–221. Springer, Cham, December 2022. 1
- [WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017. 1
- [Zim15] Joe Zimmerman. How to obfuscate programs directly. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 439–467. Springer, Berlin, Heidelberg, April 2015. 9

A Levin Reduction From SAT to Hair–Sahai Matrix

This appendix proves Theorem 5.2. The “instance map” Red_{HS} of the reduction is the same as the previous work [HS26]. The main purpose of this appendix is to prove that this reduction also has the “forward” *witness map* Map_{HS} and “backward” *witness recovery* Ext_{HS} as required in Theorem 5.2. We show this by examining every stage of the long chain of the reduction in [HS26].

Roadmap. This section is organized as follows. The full reduction in [HS26] starts with [BGLR25], then it applies the earlier work of quasi-random PCP [Kho06b] and the later generalization [KS16] to obtain an instance of QRDH. Finally, [HS26] reduces the QRDH instance to their basis matrix in GapSVP. Each subsection of this appendix corresponds to one step in the reduction chain and proves that this step is a Karp–Levin reduction.

$$\begin{array}{c} \text{SAT} \xrightarrow[\text{Section A.1}]{[\text{BGLR25}]} \text{GapMDC}_{\mathbb{F}_2} \xrightarrow[\text{Section A.2}]{[\text{HS26}] \text{ Amplification}} \text{GapMDC}_{\mathbb{K}/\mathbb{F}_2} \xrightarrow[\text{Section A.3}]{[\text{Kho06b}]} \text{HomAlgCSP} \\ \xrightarrow[\text{Section A.4}]{[\text{Kho06b}] \text{ Outer verifier} + [\text{KS16}] \text{ Inner verifier}} \text{QRDH} \xrightarrow[\text{Sections A.5 and A.6}]{[\text{HS26}]} (\mathbf{C}_\varphi, h_\varphi). \end{array}$$

A.1 BGLR Reduction From SAT to GapMDC over $\mathbb{F}_2$

This subsection proves the following lemma.

Lemma A.1 (BGLR reduction). *There exist universal constants $0 < \alpha_c < \alpha_s < 1$ and a deterministic polynomial-time algorithm Red_{BGLR} from SAT to GapMDC over $\mathbb{F}_2$ with relative completeness threshold α_c and relative soundness threshold α_s . The algorithm Red_{BGLR} takes as input any Boolean circuit φ of fan-in at most two and n gates, and outputs*

$$\text{Red}_{\text{BGLR}}(\varphi) = \mathbf{G}_\varphi, \quad \mathbf{G}_\varphi \in \mathbb{F}_2^{M_0 \times N_0},$$

where $\mathbf{G}_\varphi$ has full row rank and $1 \leq M_0 \leq N_0$. If φ is satisfiable, then the row span of $\mathbf{G}_\varphi$ contains a non-zero vector of Hamming weight at most $\alpha_c N_0$. If φ is unsatisfiable, then every non-zero codeword in the row span of $\mathbf{G}_\varphi$ has Hamming weight at least $\alpha_s N_0$.

Furthermore, we have the following two properties.

- **Witness map for completeness.** *There exists a deterministic polynomial-time algorithm Map_{BGLR} such that, given φ and any input assignment w satisfying $\varphi(w) = 1$, the algorithm $\text{Map}_{\text{BGLR}}(\varphi, w)$ outputs a non-zero message*

$$\mathbf{m}_w \in \mathbb{F}_2^{1 \times M_0} \text{ satisfying } \text{wt}(\mathbf{m}_w \cdot \mathbf{G}_\varphi) \leq \alpha_c N_0.$$

- **Witness recovery for soundness.** *There exists a deterministic polynomial-time algorithm Ext_{BGLR} such that, given φ and any non-zero message*

$$\mathbf{m} \in \mathbb{F}_2^{1 \times M_0} \text{ satisfying } \text{wt}(\mathbf{m} \cdot \mathbf{G}_\varphi) < \alpha_s N_0,$$

the algorithm $\text{Ext}_{\text{BGLR}}(\varphi, \mathbf{m})$ outputs an input assignment w satisfying $\varphi(w) = 1$.

Consequently, Red_{BGLR} , Map_{BGLR} , and Ext_{BGLR} constitute a polynomial-time Karp–Levin reduction from SAT to GapMDC over $\mathbb{F}_2$ with fixed relative thresholds $\alpha_c < \alpha_s$.

The instance map Red_{BGLR} is already given in [BGLR25]. The main purpose of this lemma is to show the existence of the “forward” witness map Map_{BGLR} and the “backward” witness recovery algorithm Ext_{BGLR} . To show they exist, we examine each step of the reduction in [BGLR25] and show that they are Karp–Levin reductions. The reduction Red_{BGLR} in [BGLR25] consists of two steps:

$$\text{SAT} \xrightarrow{\text{Lemma A.2}} \text{HomQuad} \xrightarrow{\text{Lemma A.3}} \text{GapMDC}_{\mathbb{F}_2},$$

where the homogeneous quadratic equation problem (HomQuad) is the following problem.

Homogeneous Quadratic Equations (HomQuad). The instance is a set of quadratic forms $Q_1, \dots, Q_{m_{\text{HQ}}}$ over n_{HQ} variables $x_1, \dots, x_n$ in $\mathbb{F}_2$, where each quadratic form Q_i is homogeneous:

$$Q_i = \sum_{j \leq k} q_i^{(j,k)} \cdot x_j x_k.$$

We denote such homogeneous quadratic form Q_i as $Q_i(\mathbf{x}\mathbf{x}^\top)$. The intended meaning for Q_i is to represent a quadratic equation $Q_i(\mathbf{x}\mathbf{x}^\top) = 0$. The problem is to decide, given such an instance, whether there exists a non-zero solution $\mathbf{x} \in \mathbb{F}_2^n$ such that

$$Q_1(\mathbf{x}\mathbf{x}^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}\mathbf{x}^\top) = 0.$$

We first show that the reduction from SAT to HomQuad in [BGLR25] is a Karp–Levin reduction.

Lemma A.2 (Karp–Levin reduction from SAT to HomQuad, implicit in [BGLR25]). *There exists a deterministic polynomial-time algorithm Red_{HQ} with the following property. Given a SAT instance φ represented by a Boolean circuit with n gates, the algorithm outputs*

$$\text{Red}_{\text{HQ}}(\varphi) = (Q_1, \dots, Q_{m_{\text{HQ}}}),$$

where the Q_i ’s are homogeneous quadratic forms over $\mathbb{F}_2$ in n_{HQ} variables, with $n_{\text{HQ}}, m_{\text{HQ}} = O(n)$. Moreover, the reduction is a Levin reduction:

- **Witness map.** *There exists a deterministic polynomial-time algorithm Map_{HQ} that, given φ and any satisfying assignment w , outputs a non-zero vector $\mathbf{x}_w \in \mathbb{F}_2^{n_{\text{HQ}}}$ such that*

$$Q_1(\mathbf{x}_w \mathbf{x}_w^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}_w \mathbf{x}_w^\top) = 0.$$

- **Witness recovery.** *There exists a deterministic polynomial-time algorithm Ext_{HQ} that, given φ and any non-zero vector $\mathbf{x} \in \mathbb{F}_2^{n_{\text{HQ}}}$ such that*

$$Q_1(\mathbf{x}\mathbf{x}^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}\mathbf{x}^\top) = 0,$$

outputs a satisfying assignment w for φ .

Proof. We first briefly recall the reduction Red_{HQ} in [BGLR25, Proposition 2.1 and Appendix A], and then show that it is indeed a Levin reduction. Let $g_1, \dots, g_n$ be the gates of φ , including its input gates, and let g_o be its output gate. For each gate g_i , introduce a variable x_i representing its value. Introduce one additional variable c , whose intended value is 1. Thus, the resulting homogeneous quadratic system has $n_{\text{HQ}} = n + 1$ variables $x_1, \dots, x_n, c$.

For every $i \in [n]$, include the equation

$$x_i(x_i - c) = 0.$$

This equation enforces that, if $c = 0$, then $x_i = 0$.

The computation performed by each non-input gate is represented by one homogeneous quadratic equation. For every AND gate

$$g_k = g_i \wedge g_j,$$

include

$$x_k^2 - x_i x_j = 0.$$

For every NOT gate

$$g_k = \neg g_i,$$

include

$$c^2 - x_k^2 - x_i^2 = 0.$$

Finally, enforce that the output gate evaluates to 1 by including

$$c^2 - x_o^2 = 0.$$

Every equation is homogeneous of degree two. The system contains n equations of the form $x_i(x_i - c) = 0$, one equation for each non-input gate, and one output equation. Therefore, $m_{\text{HQ}} = O(n)$

Witness map Map_{HQ} . This map is indeed implicit in the proof of completeness for the reduction. Given a satisfying assignment w , the algorithm Map_{HQ} evaluates the circuit φ on w . Let $y_i \in \{0, 1\}$ be the resulting value of gate g_i . The algorithm outputs

$$\mathbf{x}_w = (y_1, \dots, y_n, c = 1).$$

This vector is non-zero because its final coordinate is $c = 1$. The Booleanity equations hold because each $y_i \in \{0, 1\}$. The remaining equations follow from the computations of the corresponding gates, and the output equation holds because w satisfies φ . Consequently,

$$Q_1(\mathbf{x}_w \mathbf{x}_w^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}_w \mathbf{x}_w^\top) = 0.$$

Witness recovery Ext_{HQ} . The algorithm Ext_{HQ} is implicit in the soundness proof of the reduction. Let

$$\mathbf{x} = (x_1, \dots, x_n, c) \in \mathbb{F}_2^{m_{\text{HQ}}}$$

be any non-zero solution to the homogeneous quadratic system. If $c = 0$, then the equations $x_i(x_i - c) = 0$ imply $x_i = 0$ for every $i \in [n]$. This would make $\mathbf{x}$ the all-zero vector, contradicting the assumption that $\mathbf{x}$ is non-zero. Therefore, $c = 1$.

Then each gate equation implies that the x_i 's form a correct computation of every gate of φ . The output equation gives $x_o = 1$. Thus, the restriction of $(x_1, \dots, x_n)$ to the input gates is a satisfying assignment for φ . The algorithm Ext_{HQ} simply outputs this restriction to the input gates. $\square$

Next, we show the reduction from HomQuad to GapMDC in [BGLR25] is also a Karp–Levin reduction.

Lemma A.3 (Karp–Levin reduction from HomQuad to GapMDC, implicit in [BGLR25]). *There exist universal constants $0 < \alpha_c < \alpha_s < 1$ and a deterministic polynomial-time algorithm $\text{Red}_{\text{HQ} \rightarrow \text{MDC}}$ with the following property.*

Given a HomQuad instance

$$\mathcal{Q} = (Q_1, \dots, Q_{m_{\text{HQ}}})$$

over $\mathbb{F}_2$ in n_{HQ} variables, the algorithm outputs

$$\text{Red}_{\text{HQ} \rightarrow \text{MDC}}(\mathcal{Q}) = \mathbf{G}_{\mathcal{Q}}, \quad \mathbf{G}_{\mathcal{Q}} \in \mathbb{F}_2^{M_{\mathcal{Q}} \times N_{\mathcal{Q}}},$$

where $\mathbf{G}_{\mathcal{Q}}$ has full row rank, $M_{\mathcal{Q}} = O(n_{\text{HQ}}^2)$, $N_{\mathcal{Q}} = O(n_{\text{HQ}}^4)$.

If $\mathcal{Q}$ has a non-zero solution, then the row span of $\mathbf{G}_{\mathcal{Q}}$ contains a non-zero codeword of Hamming weight at most $\alpha_c N_{\mathcal{Q}}$. If $\mathcal{Q}$ has no non-zero solution, then every non-zero codeword in the row span of $\mathbf{G}_{\mathcal{Q}}$ has Hamming weight at least $\alpha_s N_{\mathcal{Q}}$.

Moreover, the reduction is a Levin reduction:

- **Witness map.** *There exists a deterministic polynomial-time algorithm $\text{Map}_{\text{HQ} \rightarrow \text{MDC}}$ that, given as input $\mathcal{Q}$ and any non-zero vector*

$$\mathbf{x} \in \mathbb{F}_2^{n_{\text{HQ}}} \text{ satisfying } Q_1(\mathbf{x}\mathbf{x}^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}\mathbf{x}^\top) = 0,$$

outputs a non-zero message

$$\mathbf{m}_{\mathbf{x}} \in \mathbb{F}_2^{1 \times M_{\mathcal{Q}}} \text{ such that } \text{wt}(\mathbf{m}_{\mathbf{x}} \mathbf{G}_{\mathcal{Q}}) \leq \alpha_c N_{\mathcal{Q}}.$$

- **Witness recovery.** *There exists a deterministic polynomial-time algorithm $\text{Ext}_{\text{HQ} \rightarrow \text{MDC}}$ that, given $\mathcal{Q}$ and any non-zero message*

$$\mathbf{m} \in \mathbb{F}_2^{1 \times M_{\mathcal{Q}}} \text{ satisfying } \text{wt}(\mathbf{m} \mathbf{G}_{\mathcal{Q}}) < \alpha_s N_{\mathcal{Q}},$$

outputs a non-zero vector

$$\mathbf{x} \in \mathbb{F}_2^{n_{\text{HQ}}} \text{ satisfying } Q_1(\mathbf{x}\mathbf{x}^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}\mathbf{x}^\top) = 0.$$

Proof sketch. The reduction $\text{Red}_{\text{HQ} \rightarrow \text{MDC}}$ is given in [BGLR25, Sections 3.1–3.2]. We first briefly recall the reduction and its analysis, then we verify the witness-map and witness-recovery guarantees.

Construction of the code $\mathbf{G}_{\mathcal{Q}}$. Define the following linear space over $\mathbb{F}_2$

$$\mathcal{X}_{\mathcal{Q}} := \left\{ X \in \mathbb{F}_2^{n_{\text{HQ}} \times n_{\text{HQ}}} : X^\top = X \text{ and } Q_i(X) = 0 \text{ for every } i \in [m_{\text{HQ}}] \right\}.$$

The space $\mathcal{X}_{\mathcal{Q}}$ captures the linearized constraints of the quadratic system

$$Q_1(\mathbf{x}\mathbf{x}^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}\mathbf{x}^\top) = 0$$

under the substitution $X = \mathbf{x}\mathbf{x}^\top$. The remaining condition is that X must have rank one, whereas $\mathcal{X}_Q$ may also contain matrices of higher rank.

To translate this rank-1 constraint into a Hamming-weight gap, BGLR constructs a full-column-rank matrix

$$\mathbf{E} \in \mathbb{F}_2^{L \times n_{\text{HQ}}}, \text{ where } L = O(n_{\text{HQ}}^2),$$

together with universal constants $0 < \alpha_c < \alpha_s < 1$, satisfying the following properties:

- For every $\mathbf{x} \in \mathbb{F}_2^{n_{\text{HQ}}}$, let $X = \mathbf{x}\mathbf{x}^\top$, then $\text{wt}(\mathbf{E}X\mathbf{E}^\top) \leq \alpha_c L^2$.
- For every $X \in \mathbb{F}_2^{n_{\text{HQ}} \times n_{\text{HQ}}}$ with $\text{rank}(X) \geq 2$, $\text{wt}(\mathbf{E}X\mathbf{E}^\top) \geq \alpha_s L^2$.

The matrix $\mathbf{E}$ can be constructed in polynomial time from n_{HQ} using the balanced-code construction and the rank-two weight bound of [BGLR25, Lemmas 2.4 and 3.2].

Given $\mathbf{E}$, define the linear code

$$\mathcal{C}_Q := \left\{ \text{vec}(\mathbf{E}X\mathbf{E}^\top) : X \in \mathcal{X}_Q \right\} \subseteq \mathbb{F}_2^{L^2}.$$

The reduction computes a full-row-rank generator matrix $\mathbf{G}_Q$ for $\mathcal{C}_Q$ by Gaussian elimination and outputs

$$\text{Red}_{\text{HQ} \rightarrow \text{MDC}}(Q) = \mathbf{G}_Q.$$

Since $L = O(n_{\text{HQ}}^2)$, the block length of $\mathcal{C}_Q$ is $N_Q = L^2 = O(n_{\text{HQ}}^4)$.

Witness map $\text{Map}_{\text{HQ} \rightarrow \text{MDC}}$. Let $\mathbf{x} \in \mathbb{F}_2^{n_{\text{HQ}}} \setminus \{\mathbf{0}\}$ satisfy

$$Q_1(\mathbf{x}\mathbf{x}^\top) = \dots = Q_{m_{\text{HQ}}}(\mathbf{x}\mathbf{x}^\top) = 0,$$

and set

$$X := \mathbf{x}\mathbf{x}^\top.$$

Then $X \in \mathcal{X}_Q$, so

$$\text{vec}(\mathbf{E}X\mathbf{E}^\top) \in \mathcal{C}_Q.$$

Using Gaussian elimination, compute the unique message $\mathbf{m}_\mathbf{x}$ satisfying

$$\mathbf{m}_\mathbf{x} \mathbf{G}_Q = \text{vec}(\mathbf{E}X\mathbf{E}^\top).$$

The message $\mathbf{m}_\mathbf{x}$ is non-zero because $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{E}$ is injective. Moreover, the first property of $\mathbf{E}$ above gives

$$\text{wt}(\mathbf{m}_\mathbf{x} \mathbf{G}_Q) \leq \alpha_c N_Q.$$

Witness recovery $\text{Ext}_{\text{HQ} \rightarrow \text{MDC}}$. Let $\mathbf{m} \neq \mathbf{0}$ satisfy

$$\text{wt}(\mathbf{m} \mathbf{G}_Q) < \alpha_s N_Q.$$

Since $\mathbf{G}_Q$ has full row rank, $\mathbf{m} \mathbf{G}_Q$ is non-zero. Let Y be the matrix satisfying

$$\text{vec}(Y) = \mathbf{m} \mathbf{G}_Q.$$

Because $\text{vec}(Y) \in \mathcal{C}_{\mathcal{Q}}$, there is a unique matrix $X \in \mathcal{X}_{\mathcal{Q}}$ such that

$$Y = \mathbf{E}X\mathbf{E}^{\top}.$$

This matrix can be recovered efficiently using a left inverse of $\mathbf{E}$.

If $\text{rank}(X) \geq 2$, the second property of $\mathbf{E}$ above would imply

$$\text{wt}(\mathbf{m}\mathbf{G}_{\mathcal{Q}}) = \text{wt}(\mathbf{E}X\mathbf{E}^{\top}) \geq \alpha_s N_{\mathcal{Q}},$$

contradicting the assumed strict inequality. Since $Y \neq 0$ and the map $X \mapsto \mathbf{E}X\mathbf{E}^{\top}$ is injective, we also have $X \neq 0$. Therefore,

$$\text{rank}(X) = 1.$$

Since X is a non-zero symmetric rank-one matrix over $\mathbb{F}_2$, there exists a non-zero vector $\mathbf{x} \in \mathbb{F}_2^{n_{\text{HQ}}}$ such that

$$X = \mathbf{x}\mathbf{x}^{\top}.$$

Such a vector can be recovered in polynomial time by finding an index j with $X_{j,j} = 1$ and taking the j -th column of X .

Finally, because $X \in \mathcal{X}_{\mathcal{Q}}$,

$$Q_i(\mathbf{x}\mathbf{x}^{\top}) = Q_i(X) = 0$$

for every $i \in [m_{\text{HQ}}]$. The algorithm $\text{Ext}_{\text{HQ} \rightarrow \text{MDC}}$ outputs this vector $\mathbf{x}$. This completes the proof. $\square$

We finish this subsection by proving Lemma A.1.

Proof of Lemma A.1. For $\mathcal{Q} := \text{Red}_{\text{HQ}}(\varphi)$, define

$$\text{Red}_{\text{BGLR}}(\varphi) := \text{Red}_{\text{HQ} \rightarrow \text{MDC}}(\mathcal{Q}), \quad \text{Map}_{\text{BGLR}}(\varphi, w) := \text{Map}_{\text{HQ} \rightarrow \text{MDC}}(\mathcal{Q}, \text{Map}_{\text{HQ}}(\varphi, w)),$$

and

$$\text{Ext}_{\text{BGLR}}(\varphi, \mathbf{m}) := \text{Ext}_{\text{HQ}}(\varphi, \text{Ext}_{\text{HQ} \rightarrow \text{MDC}}(\mathcal{Q}, \mathbf{m})).$$

The preceding two lemmas establish the claimed completeness, soundness, witness-map, and witness-recovery guarantees. Moreover, $n_{\text{HQ}} = O(n)$ and $N_{\mathcal{Q}} = O(n_{\text{HQ}}^4) = O(n^4)$, and all three algorithms run in polynomial time. $\square$

A.2 Hair–Sahai Amplification of GapMDC

This subsection proves that the GapMDC gap amplification in [HS26], applied before Khot’s quasi-random PCP construction, is a Karp–Levin reduction.

Lemma A.4 (Karp–Levin amplification of GapMDC). *For any fixed constants $0 < \alpha_c < \alpha_s < 1$, there exist efficiently computable functions $\varepsilon: \mathbb{N} \rightarrow (0, 1)$ and $\ell: \mathbb{N} \rightarrow \mathbb{N}$ such that $\varepsilon(N) = N^{-\Omega(\log \log N)}$ and $\ell(N)$ is a power of two for every N . For every N_0 , fix an efficiently constructible representation of the field*

$$\mathbb{K} = \mathbb{K}_{N_0} := \mathbb{F}_{2^{\ell(N_0)}}.$$

There is a deterministic $N_0^{O(\log \log N_0)}$ -time algorithm $\text{Red}_{\text{HS-pre}}$ with the following property.

Given a full-row-rank matrix

$$\mathbf{G}_0 \in \mathbb{F}_2^{M_0 \times N_0}, \quad 1 \leq M_0 \leq N_0,$$

the algorithm outputs

$$\text{Red}_{\text{HS-pre}}(\mathbf{G}_0) = \mathbf{G}_1 \in \mathbb{K}_{N_0}^{M_1 \times N_1}.$$

The matrix $\mathbf{G}_1$ has full row rank,

$$M_1 = N_0^{O(\log \log N_0)}, \quad N_1 = N_0^{\Theta(\log \log N_0)},$$

and

$$N_1 \leq |\mathbb{K}_{N_0}| = 2^{\ell(N_0)} \leq N_1^2.$$

- If the row span of $\mathbf{G}_0$ contains a non-zero codeword of Hamming weight at most $\alpha_c N_0$, then the row span of $\mathbf{G}_1$ contains a non-zero codeword of Hamming weight at most $N_1/2$.
- If every non-zero codeword in the row span of $\mathbf{G}_0$ has Hamming weight at least $\alpha_s N_0$, then every non-zero codeword in the row span of $\mathbf{G}_1$ has Hamming weight at least $(1 - \varepsilon(N_0))N_1$.

Furthermore, the reduction is a Levin reduction:

- **Witness map for completeness.** There exists a deterministic $N_0^{O(\log \log N_0)}$ -time algorithm $\text{Map}_{\text{HS-pre}}$ such that, given $\mathbf{G}_0$ and any non-zero message

$$\mathbf{m}_0 \in \mathbb{F}_2^{1 \times M_0} \text{ satisfying } \text{wt}(\mathbf{m}_0 \mathbf{G}_0) \leq \alpha_c N_0,$$

the algorithm $\text{Map}_{\text{HS-pre}}(\mathbf{G}_0, \mathbf{m}_0)$ outputs a non-zero message

$$\mathbf{m}_1 \in \mathbb{K}_{N_0}^{1 \times M_1} \text{ satisfying } \text{wt}(\mathbf{m}_1 \mathbf{G}_1) \leq \frac{N_1}{2}.$$

- **Witness recovery for soundness.** There exists a deterministic $N_0^{O(\log \log N_0)}$ -time algorithm $\text{Ext}_{\text{HS-pre}}$ such that, given $\mathbf{G}_0$ and any non-zero message

$$\mathbf{m}_1 \in \mathbb{K}_{N_0}^{1 \times M_1} \text{ satisfying } \text{wt}(\mathbf{m}_1 \mathbf{G}_1) < (1 - \varepsilon(N_0))N_1,$$

the algorithm $\text{Ext}_{\text{HS-pre}}(\mathbf{G}_0, \mathbf{m}_1)$ outputs a non-zero message

$$\mathbf{m}_0 \in \mathbb{F}_2^{1 \times M_0} \text{ satisfying } \text{wt}(\mathbf{m}_0 \mathbf{G}_0) < \alpha_s N_0.$$

The Hair–Sahai amplification in Lemma A.4 repeatedly uses two subroutines: a tensor-product amplifier, and an expander–Vandermonde amplifier, which we show in Lemmas A.5 and A.6 that they are both Levin reductions. We defer the proof of Lemma A.4 to the end of this subsection.

Lemma A.5 (Karp–Levin tensor amplifier for GapMDC). *Let $\mathbb{K}$ be a finite field. For every $i \in \{1, 2\}$, let $0 < \alpha_i < \beta_i < 1$ and let*

$$\mathbf{G}_i \in \mathbb{K}^{M_i \times N_i}$$

be a full-row-rank matrix. There exists a deterministic polynomial-time algorithm $\text{Red}_{\otimes}$ that outputs

$$\text{Red}_{\otimes}(\mathbf{G}_1, \mathbf{G}_2) = \mathbf{G}_{\otimes} := \mathbf{G}_1 \otimes \mathbf{G}_2 \in \mathbb{K}^{M_1 M_2 \times N_1 N_2}.$$

The matrix $\mathbf{G}_{\otimes}$ has full row rank and satisfies the following properties.

- If, for each $i \in \{1, 2\}$, the row span of $\mathbf{G}_i$ contains a non-zero codeword of Hamming weight at most $\alpha_i N_i$, then the row span of $\mathbf{G}_\otimes$ contains a non-zero codeword of Hamming weight at most $\alpha_1 \alpha_2 N_1 N_2$.
- If, for each $i \in \{1, 2\}$, every non-zero codeword in the row span of $\mathbf{G}_i$ has Hamming weight at least $\beta_i N_i$, then every non-zero codeword in the row span of $\mathbf{G}_\otimes$ has Hamming weight at least $\beta_1 \beta_2 N_1 N_2$.

Moreover, the transformation has the following two-input Levin properties.

- **Witness map for completeness.** There exists a deterministic polynomial-time algorithm $\text{Map}_\otimes$ such that, given non-zero messages

$$\mathbf{m}_i \in \mathbb{K}^{1 \times M_i} \quad \text{with} \quad \text{wt}(\mathbf{m}_i \mathbf{G}_i) \leq \alpha_i N_i \quad \text{for every } i \in \{1, 2\},$$

the algorithm

$$\text{Map}_\otimes(\mathbf{G}_1, \mathbf{G}_2, \mathbf{m}_1, \mathbf{m}_2)$$

outputs the non-zero message $\mathbf{m}_1 \otimes \mathbf{m}_2$ satisfying

$$\text{wt}((\mathbf{m}_1 \otimes \mathbf{m}_2) \mathbf{G}_\otimes) \leq \alpha_1 \alpha_2 N_1 N_2.$$

- **Witness recovery for soundness.** There exists a deterministic polynomial-time algorithm $\text{Ext}_\otimes$ such that, given any non-zero message

$$\mathbf{z} \in \mathbb{K}^{1 \times M_1 M_2} \quad \text{with} \quad \text{wt}(\mathbf{z} \mathbf{G}_\otimes) < \beta_1 \beta_2 N_1 N_2,$$

the algorithm

$$\text{Ext}_\otimes(\mathbf{G}_1, \mathbf{G}_2, \mathbf{z})$$

outputs an index $i \in \{1, 2\}$ and a non-zero message $\mathbf{m}_i \in \mathbb{K}^{1 \times M_i}$ satisfying

$$\text{wt}(\mathbf{m}_i \mathbf{G}_i) < \beta_i N_i.$$

Proof. The witness map follows from

$$(\mathbf{m}_1 \otimes \mathbf{m}_2)(\mathbf{G}_1 \otimes \mathbf{G}_2) = (\mathbf{m}_1 \mathbf{G}_1) \otimes (\mathbf{m}_2 \mathbf{G}_2)$$

and hence

$$\text{wt}((\mathbf{m}_1 \otimes \mathbf{m}_2) \mathbf{G}_\otimes) = \text{wt}(\mathbf{m}_1 \mathbf{G}_1) \cdot \text{wt}(\mathbf{m}_2 \mathbf{G}_2).$$

For witness recovery, reshape $\mathbf{z}$ as a matrix $Z \in \mathbb{K}^{M_1 \times M_2}$. Under the natural indexing, the tensor codeword $\mathbf{z} \mathbf{G}_\otimes$ is the vectorization of

$$Y := \mathbf{G}_1^\top Z \mathbf{G}_2 \in \mathbb{K}^{N_1 \times N_2}.$$

Every row of Y is a codeword in the row span of $\mathbf{G}_2$, and the transpose of every column of Y is a codeword in the row span of $\mathbf{G}_1$. Moreover, Y is non-zero because $\mathbf{G}_\otimes$ has full row rank and $\mathbf{z}$ is non-zero.

If Y has a non-zero row of weight less than $\beta_2 N_2$, output its corresponding message for $\mathbf{G}_2$. Otherwise, every non-zero row has weight at least $\beta_2 N_2$. Since the total number of non-zero entries of Y is $\text{wt}(\mathbf{z} \mathbf{G}_\otimes) < \beta_1 \beta_2 N_1 N_2$, the matrix Y has fewer than $\beta_1 N_1$ non-zero rows. Any non-zero column of Y is therefore a codeword in the row span of $\mathbf{G}_1$ of weight less than $\beta_1 N_1$. The corresponding factor messages are obtained directly from $\mathbf{G}_1^\top Z$ and $Z \mathbf{G}_2$. This gives $\text{Ext}_\otimes$ and also proves the soundness claim. $\square$

Lemma A.6 (Karp–Levin expander–Vandermonde amplifier for GapMDC). *Let $\mathbb{K}$ be a finite field, let r, w be positive integers satisfying $r \leq w < |\mathbb{K}|$, and let $\delta \in (0, 1)$. The values r, w, δ are parameters of the reduction algorithm. There exists a deterministic algorithm Red_{EV} that takes any full-row-rank matrix $\mathbf{G} \in \mathbb{K}^{M \times N}$ as input, and outputs a full-row-rank matrix*

$$\text{Red}_{\text{EV}}(\mathbf{G}; r, w, \delta) = \mathbf{G}' \in \mathbb{K}^{M \times N'}, \text{ where } N' \leq O\left(Nw(1/\delta)^{O(r)}\right).$$

The reduction satisfies the following properties. Let $0 < \alpha < \beta < 1$ be two constants.

- *If the row span of $\mathbf{G}$ contains a non-zero codeword of Hamming weight at most αN , then the row span of $\mathbf{G}'$ contains a non-zero codeword of Hamming weight at most $\alpha' N'$, where*

$$\alpha' := r\alpha.$$

- *If every non-zero codeword in the row span of $\mathbf{G}$ has Hamming weight at least βN , then every non-zero codeword in the row span of $\mathbf{G}'$ has Hamming weight at least $\beta' N'$, where*

$$\beta' := 1 - \left((1 - \beta + \delta)^r + \frac{r}{w}\right).$$

Moreover, the reduction is a Levin reduction.

- **Witness map for completeness.** *There exists a deterministic polynomial-time algorithm Map_{EV} such that, given any non-zero message*

$$\mathbf{m} \in \mathbb{K}^{1 \times M} \quad \text{with} \quad \text{wt}(\mathbf{m}\mathbf{G}) \leq \alpha N,$$

the algorithm $\text{Map}_{\text{EV}}(\mathbf{G}, \mathbf{m}) = \mathbf{m}$ outputs a non-zero message satisfying

$$\text{wt}(\mathbf{m}\mathbf{G}') \leq \alpha' N'.$$

- **Witness recovery for soundness.** *There exists a deterministic polynomial-time algorithm Ext_{EV} such that, given any non-zero message*

$$\mathbf{m}' \in \mathbb{K}^{1 \times M} \quad \text{with} \quad \text{wt}(\mathbf{m}'\mathbf{G}') < \beta' N',$$

the algorithm $\text{Ext}_{\text{EV}}(\mathbf{G}, \mathbf{m}') = \mathbf{m}'$ outputs a non-zero message satisfying

$$\text{wt}(\mathbf{m}'\mathbf{G}) < \beta N.$$

Proof sketch. We first recall the construction of [HS26, Lemma 6.10]. Construct an explicit regular expander graph H whose N vertices index the columns of $\mathbf{G}$ and such that, for every $S \subseteq [N]$, the fraction of length- r walks contained entirely in S is at most $(|S|/N + \delta)^r$. Fix an $r \times w$ Vandermonde matrix V over $\mathbb{K}$. For every length- r walk $\pi = (v_1, \dots, v_r)$ in H , define

$$C_\pi := [\mathbf{G}_{*,v_1} \quad \dots \quad \mathbf{G}_{*,v_r}] \in \mathbb{K}^{M \times r} \quad \text{and} \quad \mathbf{G}_\pi := C_\pi V \in \mathbb{K}^{M \times w}.$$

The matrix $\mathbf{G}'$ is the horizontal concatenation of the blocks $\mathbf{G}_\pi$ over all length- r walks π .

Fix any non-zero message $\mathbf{m} \in \mathbb{K}^{1 \times M}$. If $\text{wt}(\mathbf{m}\mathbf{G}) \leq \alpha N$, then a block $\mathbf{m}\mathbf{G}_\pi$ can be non-zero only if π visits the support of $\mathbf{m}\mathbf{G}$. By regularity and a union bound, at most an $r\alpha$ fraction of the walks do so. Hence

$$\text{wt}(\mathbf{m}\mathbf{G}') \leq r\alpha N' = \alpha' N'.$$

Thus, the witness map outputs the same message.

Conversely, suppose that $\text{wt}(\mathbf{m}\mathbf{G}) \geq \beta N$. The expander property implies that at most a $(1 - \beta + \delta)^r$ fraction of the walks satisfy $\mathbf{m}\mathbf{C}_\pi = \mathbf{0}$. For every remaining walk, $\mathbf{m}\mathbf{C}_\pi$ is non-zero, so the Vandermonde property implies that $\mathbf{m}\mathbf{C}_\pi V$ has at most r zero coordinates among its w coordinates. Therefore, the fraction of zero coordinates in $\mathbf{m}\mathbf{G}'$ is at most

$$(1 - \beta + \delta)^r + \frac{r}{w},$$

and hence $\text{wt}(\mathbf{m}\mathbf{G}') \geq \beta' N'$. Taking the contrapositive proves the claimed witness recovery with the same message. The construction also preserves full row rank, and its number of columns and running time satisfy the stated bounds. $\square$

We finish this subsection by proving Lemma A.4.

Proof of Lemma A.4. The amplification in the proof of [HS26, Theorem 6.4] is an $O(\log \log N_0)$ -fold composition of the tensor and expander–Vandermonde amplifiers from Lemmas A.5 and A.6; the initial scalar extension from $\mathbb{F}_2$ to $\mathbb{K}/\mathbb{F}_2$ is immediately a Levin reduction.

In every tensor amplification step, both input codes are derived from the same input code $\mathbf{G}_0$, so either branch returned by the tensor witness-recovery algorithm can be traced back to $\mathbf{G}_0$. Composing the corresponding witness maps and witness-recovery algorithms proves the lemma; the stated parameters and running time are those proved in [HS26, Theorem 6.4]. $\square$

A.3 Khot’s Reduction From GapMDC to HomAlgCSP

Homogeneous Algebraic CSP (HomAlgCSP). A HomAlgCSP instance over a finite field $\mathbb{K}$ is specified by integers M, k, d, m and a multiset $\mathcal{C}$ of M constraints. Each constraint is of the form

$$C = (\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(k)}, H),$$

where $\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(k)} \in \mathbb{K}^m$ and $H \subseteq \mathbb{K}^k$ is a linear subspace. An assignment is a non-zero polynomial $f \in \mathbb{K}[X_1, \dots, X_m]$, and C is satisfied by f if

$$(f(\mathbf{p}^{(1)}), \dots, f(\mathbf{p}^{(k)})) \in H^\perp.$$

The goal is to maximize the fraction of constraints satisfied by a polynomial of total degree at most d . For $1 \geq \alpha > \beta \geq 0$, the $(\mathbb{K}, M, k, d, m, \alpha, \beta)$ -gap version distinguishes whether

- YES CASE: some non-zero polynomial of degree at most d satisfies at least an α fraction of the constraints, or
- NO CASE: every non-zero polynomial of degree at most $1000d$ satisfies at most a β fraction.

HomAlgCSP resembles a polynomial PCP or a one-round PIOP in which the polynomial is the proof oracle, and each constraint is a k -query test with a linear function as its verifier; unlike those proof systems, **HomAlgCSP** here requires a gap in the degree of the polynomials (d vs $1000d$) between the YES CASE and NO CASE.

This subsection proves the following lemma.

Lemma A.7 (Karp-Levin reduction from **GapMDC** to **HomAlgCSP**). *There exists a universal constant $c_K > 0$ for which the following holds. Let $1 \leq M \leq N$, let $0 \leq \alpha < \beta \leq 1$, and let m^*, d^* be positive integers satisfying*

$$\binom{m^*}{d^*} \geq \max\{N, M\}.$$

Let λ satisfy $N \leq 2^\lambda \leq N^2$, and set $\mathbb{K} = \mathbb{F}_{2^\lambda}$. There is a deterministic

$$\exp\left(O((m^*d^*)^{O(1)})\right)\text{-time}$$

algorithm $\text{Red}_{\text{MDC} \rightarrow \text{HA}}$ that, given a full-row-rank matrix $\mathbf{G} \in \mathbb{K}^{M \times N}$, outputs an instance $\mathcal{A}_{\mathbf{G}}$ of

$$(\mathbb{K}, M', k, d', m', \alpha', \beta')\text{-HomAlgCSP},$$

where

$$\begin{aligned} M' &= \exp\left(O((m^*d^*)^{O(1)})\right), & k &= 21, \\ d' &= O(d^*), & m' &= O((m^*d^*)^{O(1)}), \\ \alpha' &= 1 - \alpha, & \beta' &= \max\left\{1 - \beta, \frac{c_K d^*}{2^\lambda}\right\}. \end{aligned}$$

Whenever $\beta' < \alpha'$, these parameters define a non-trivial gap. Moreover, the reduction has the following Levin properties.

- **Witness map for completeness.** *There exists a deterministic $\exp(O((m^*d^*)^{O(1)}))$ -time algorithm $\text{Map}_{\text{MDC} \rightarrow \text{HA}}$ such that, given $\mathbf{G}$ and any non-zero message $\mathbf{m} \in \mathbb{K}^{1 \times M}$ satisfying*

$$\text{wt}(\mathbf{m}\mathbf{G}) \leq \alpha N,$$

it outputs a non-zero assignment polynomial $\Phi_{\mathbf{m}} \in \mathbb{K}[X_1, \dots, X_{m'}]$ of total degree at most d' that satisfies at least an α' fraction of the constraints of $\mathcal{A}_{\mathbf{G}}$.

- **Witness recovery for soundness.** *There exists a deterministic $\exp(O((m^*d^*)^{O(1)}))$ -time algorithm $\text{Ext}_{\text{MDC} \rightarrow \text{HA}}$ such that, given $\mathbf{G}$ and any non-zero assignment polynomial $\Phi \in \mathbb{K}[X_1, \dots, X_{m'}]$ of total degree at most $1000d'$ that satisfies more than a β' fraction of the constraints of $\mathcal{A}_{\mathbf{G}}$, it outputs a non-zero message $\mathbf{m} \in \mathbb{K}^{1 \times M}$ satisfying*

$$\text{wt}(\mathbf{m}\mathbf{G}) < \beta N.$$

This is the Levin form of [HS26, Theorem 6.19].

Proof sketch. We first briefly recall the reduction and analysis in [Kho06b, Sections 3–6], and then we verify that this reduction is indeed a Levin reduction at the end of this proof.

Encoding codewords as polynomials. The first step is to encode every codeword generated by $\mathbf{G}$ into the coefficients of a homogeneous multilinear polynomial of degree d^* . Specifically, choose a family $\mathcal{F}_0 \subseteq \binom{[m^*]}{d^*}$ of size N . For any $\mathbf{a} \in \mathbb{K}^N$, identify its coordinates with the sets in $\mathcal{F}_0$ and write $\mathbf{a} = (a_J)_{J \in \mathcal{F}_0}$. For $J \in \mathcal{F}_0$, let $X_J := \prod_{j \in J} X_j$. Khot encodes $\mathbf{a}$ as

$$P_{\mathbf{a}}(\mathbf{X}) := \sum_{J \in \mathcal{F}_0} a_J X_J.$$

If $\mathbf{p}_J \in \{0, 1\}^{m^*}$ denotes the indicator vector of $J \in \mathcal{F}_0$, then $P_{\mathbf{a}}(\mathbf{p}_J) = a_J$: indeed, $X_{J'}(\mathbf{p}_J) = 1$ if and only if $J' = J$, since all sets in $\mathcal{F}_0$ have cardinality d^* .

We next apply this encoding to the rows of a systematic generator matrix for the same code. Choose a subfamily $\mathcal{F} \subseteq \mathcal{F}_0$ of size M . Since $\mathbf{G}$ has full row rank, after permuting its columns, we may assume that its first M columns form an invertible matrix R . Gaussian elimination gives

$$\hat{\mathbf{G}} = R^{-1}\mathbf{G} = [I_M \quad \mathbf{G}'].$$

This matrix generates the same code, up to the preceding coordinate permutation. Index its first M columns and its rows by $\mathcal{F}$, and all its columns by $\mathcal{F}_0$. For $I \in \mathcal{F}$, encode the row indexed by I as

$$f_I(\mathbf{X}) := \sum_{J \in \mathcal{F}_0} \hat{G}_{I,J} X_J.$$

Thus $f_I = P_{\hat{\mathbf{G}}_{I,*}}$. Consequently, for every $\mathbf{z} \in \mathbb{K}^{1 \times M}$,

$$f_{\mathbf{z}} := \sum_{I \in \mathcal{F}} z_I f_I = P_{\mathbf{z}\hat{\mathbf{G}}}, \quad (f_{\mathbf{z}}(\mathbf{p}_J))_{J \in \mathcal{F}_0} = \mathbf{z}\hat{\mathbf{G}}.$$

Hence the minimum-distance problem is equivalent to finding a non-zero polynomial in $\text{span}\{f_I : I \in \mathcal{F}\}$ that is non-zero on as few points $\mathbf{p}_J$ as possible:

$$\min_{0 \neq f \in \text{span}\{f_I : I \in \mathcal{F}\}} \Pr_{J \leftarrow \mathcal{F}_0} [f(\mathbf{p}_J) \neq 0].$$

Looking ahead, we will include such an f in the witness polynomial for **HomAlgCSP**. The construction so far is Levin-style: given any codeword, we can construct its encoding f in polynomial time; conversely, given any optimal f , we can recover a codeword in polynomial time by evaluating f at the points $\{\mathbf{p}_J\}_{J \in \mathcal{F}_0}$.

Recall that the goal here is to convert the linear subspace membership constraint $f \in \text{span}\{f_I : I \in \mathcal{F}\}$ into a “very local” ($k = O(1)$) **HomAlgCSP** instance. Khot achieves this by designing a generalized sum-check protocol to verify this linear subspace membership. Specifically, define

$$\Psi(\mathbf{q}, \mathbf{X}) := \sum_{I \in \mathcal{F}} f_I(\mathbf{q}) X_I.$$

Khot observes that a polynomial f belongs to $\text{span}\{f_I : I \in \mathcal{F}\}$ if and only if the formal identity

$$f(\mathbf{q}) = \sum_{I \in \mathcal{F}} \Psi(\mathbf{q}, \mathbf{p}_I) f(\mathbf{p}_I) \tag{1}$$

holds. For intuition, consider an interactive proof for (1). The verifier first sends a uniformly random point $\mathbf{q}^* \leftarrow \mathbb{K}^{m^*}$. The right-hand side is then the sum $\sum_{I \in \mathcal{F}} \Psi(\mathbf{q}^*, \mathbf{p}_I) f(\mathbf{p}_I)$, suggesting a sum-check protocol to further verify this identity. However, this sum ranges over $\mathcal{F}$ rather than the Boolean hypercube used in standard sum-check. Khot therefore introduced a generalized sum-check protocol for such sums over $\mathcal{F}$ [Kho06b, Section 5].

Generalized sum-check. The generalized sum-check theorem of Khot [Kho06b, Section 5] states the following. Given a degree- D polynomial P in m^* variables, a family $\mathcal{F} \subseteq \binom{[m^*]}{d^*}$, and a claimed value a , there is an “oracle proof” between a prover and a verifier for the identity

$$a = \sum_{I \in \mathcal{F}} P(\mathbf{p}_I),$$

where the verifier is given oracle access to degree- $(D + d^*)$ polynomials

$$g_{b,r}(\boldsymbol{\alpha}, \boldsymbol{\theta}; X), \quad b \in \{0, 1\}, \quad 0 \leq r < m^*,$$

where $g_{b,r}$ is premised to depend only on $\alpha_1, \dots, \alpha_r, \theta_1, \dots, \theta_r, X$. Intuitively, this “one-round oracle proof” collapses an m^* -round generalized sum-check protocol by giving out all $g_{b,r}$ ’s as oracles in one shot, where $g_{b,r}, b = 0, 1$ represents the “prover’s strategy” in the r -th round of the protocol. The verifier queries one value of P and two values of each $g_{b,r}$, and performs one homogeneous linear test.

The completeness ensures that if $a = \sum_{I \in \mathcal{F}} P(\mathbf{p}_I)$, there exist polynomials $g_{b,r}$ that make the verifier accept with probability one; the soundness guarantees that, for any incorrect value of a , every such collection $\{g_{b,r}\}_{b,r}$ is accepted with probability at most $O((D + d^*)/|\mathbb{K}|)$. Khot applies this theorem to $P(\mathbf{X}) = \Psi(\mathbf{q}^*, \mathbf{X})f(\mathbf{X})$, with $D = 2d^*$ and $a = f(\mathbf{q}^*)$, to test (1).

The verifier at this stage still takes $\Omega(m^*)$ queries to these polynomials $g_{b,r}$. The clubbing step below further reduces the $\Omega(m^*)$ queries to a constant number of queries.

Clubbing the prover polynomials in sum-check. Khot encodes all prover strategies $\{g_{b,r}\}_{b,r}$ into a single polynomial serving as the NP witness for HomAlgCSP. Hence, the verifier’s message $\mathbf{q}^*$ should also be treated as a formal variable $\mathbf{q}$. Write

$$g_{b,r}(\mathbf{q}, \boldsymbol{\alpha}, \boldsymbol{\theta}; X) = \sum_t g_{b,r,t}(\mathbf{q}, \boldsymbol{\alpha}, \boldsymbol{\theta}) X^t.$$

To combine all $g_{b,r,t}$ together into one polynomial, Khot gives each coefficient polynomial a private copy of precisely the variables on which it may depend and clubs them into

$$g(\{\mathbf{y}^{(b,r,t)}, S_{b,r,t}\}_{b,r,t}) = \sum_{b,r,t} g_{b,r,t}(\mathbf{y}^{(b,r,t)}) S_{b,r,t},$$

where $S_{b,r,t}$ ’s are newly introduced variables that can be viewed as linear “selectors”. This combines the polynomials $\{g_{b,r}\}$ into a single polynomial g as required by the HomAlgCSP instance. However, Khot further needs to ensure this combined polynomial g is well-formed. Namely, for any maliciously chosen g^* , one can extract $g_{b,r,t}^*$ by setting the values of those selectors appropriately, and then one needs to ensure $g_{b,r,t}^*$ depends only on the variables in $\mathbf{y}^{(b,r,t)}$. Therefore, the reduction also tests that g is “splittable” across the variable blocks $(\mathbf{y}^{(b,r,t)}, S_{b,r,t})$ [Kho06b, Section 6].

Splittable tests. A polynomial in disjoint variable blocks $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(s)}$ is *splittable* if it is a sum $\sum_{j=1}^s g_j(\mathbf{y}^{(j)})$. Khot’s splittability theorem [Kho06b, Section 4] states that, for every degree- D polynomial g , there is a degree- $2D$ certificate polynomial h and a verifier that reads one value of g and four values of h . If g is splittable, there is a unique h that is accepted with probability one; if g is not splittable, or if an incorrect h is used, the acceptance probability is at most $O(D/|\mathbb{K}|)$. Moreover, $g \equiv 0$ has the certificate $h \equiv 0$. Together with linearity in the selectors, this test forces the variable dependence required by generalized sum-check.

Clubbing all polynomials f, g, h . Finally, Khot places f, g, h on disjoint variable blocks and clubs them into

$$\Phi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, U, V, W) := Uf(\mathbf{X}) + Vg(\mathbf{Y}) + Wh(\mathbf{Z}).$$

Linearity in U, V, W and three block-independence tests force this form. Together with the test $f(\mathbf{p}_J) = 0$ for uniformly random $J \in \mathcal{F}_0$, all the preceding tests become constraints consisting of homogeneous linear equations on $k = 21$ evaluations of Φ ; see [Kho06b, Sections 6.1–6.2]. Their total degree and number of variables give the stated d' and m' .

Witness map for completeness. Given a non-zero message $\mathbf{m}$, let $\mathbf{z} = \mathbf{m}R$. Then $\mathbf{z}\hat{\mathbf{G}} = \mathbf{m}\mathbf{G}$, so $f_{\mathbf{z}} = P_{\mathbf{m}\mathbf{G}}$. Construct the canonical generalized sum-check polynomial g and its splittability certificate h , and output the clubbed polynomial $\Phi_{\mathbf{m}}$. All structural and subspace-membership tests accept with probability one, while the last test accepts exactly when $(\mathbf{m}\mathbf{G})_J = 0$. Hence $\Phi_{\mathbf{m}}$ is non-zero and satisfies a $1 - \text{wt}(\mathbf{m}\mathbf{G})/N$ fraction of the constraints. This defines $\text{Map}_{\text{MDC} \rightarrow \text{HA}}$. We observe that all the honest prover polynomials, including $g_{b,r}, f, g, h$ above can be constructed in $\exp((m^*d^*)^{O(1)})$ time.

Witness recovery for soundness. Given a non-zero Φ satisfying more than a β' fraction of the constraints, define $f(\mathbf{X}) = \Phi(\mathbf{X}, \mathbf{0}, \mathbf{0}, 1, 0, 0)$. Since its acceptance fraction exceeds $\beta' \geq c_K d^*/|\mathbb{K}|$, the soundness of the clubbing, splittability, and generalized sum-check tests forces f to be non-zero and to satisfy (1). Set $z_I = f(\mathbf{p}_I)$ for $I \in \mathcal{F}$ and output $\mathbf{m} = \mathbf{z}R^{-1}$. This defines $\text{Ext}_{\text{MDC} \rightarrow \text{HA}}$.

Equation (1) gives $f = f_{\mathbf{z}}$ and hence $(f(\mathbf{p}_J))_{J \in \mathcal{F}_0} = \mathbf{m}\mathbf{G}$. Since the acceptance fraction is greater than $1 - \beta$, this codeword has weight less than βN . The same analysis for degree at most $1000d'$ only changes the universal constant in the $O(d^*/|\mathbb{K}|)$ term, as captured by [HS26, Theorem 6.19]. $\square$

A.4 Reduction From HomAlgCSP to QRDH

In this subsection, we show that Khot's reduction from HomAlgCSP to the quasi-random dense hypergraph problem (QRDH) (equivalently, to a quasi-random PCP) is a Karp–Levin reduction. We first recall the definition of QRDH in [HS26].

Quasi-random dense hypergraph problem (QRDH). For a d_Q -uniform hypergraph $\mathcal{H} = (V, E)$, where E may be a multiset, and a set $U \subseteq V$, let

$$e_{\mathcal{H}}(U) := |\{e \in E : e \subseteq U\}|$$

denote the number of hyperedges that are fully contained in U . An $(N_Q, M_Q, d_Q, r, \alpha, \beta)$ -QRDH instance is a d_Q -uniform hypergraph with $|V| = N_Q$ and $|E| = M_Q$, and distinguishes between

- YES CASE: there exists $U \subseteq V$ such that

$$|U| \leq \frac{N_Q}{r} \quad \text{and} \quad e_{\mathcal{H}}(U) \geq \alpha \left(\frac{1}{r}\right)^{d_Q-1} M_Q;$$

- NO CASE: every $U \subseteq V$ satisfies

$$e_{\mathcal{H}}(U) \leq \left(\frac{|U|}{N_Q}\right)^{d_Q} M_Q + \beta M_Q.$$

Note that the term $(|U|/N_Q)^{d_Q}$ is the probability that d_Q independent uniform vertices all lie in U . Thus the NO CASE bounds the number of induced edges by this random baseline, up to an additive βM_Q . At density $1/r$, the YES CASE guarantees at least αr times the baseline number of edges, since its exponent is $d_Q - 1$ rather than d_Q .

Combining Khot's quasi-random PCP for HomAlgCSP from [Kho06b, Sections 7–10] with the Khot–Saket inner verifier from [KS16, Section 5] gives a reduction from HomAlgCSP to QRDH. Hair and Sahai abstract this existing reduction in [HS26, Theorem 6.21]. In this subsection, we briefly recall the reduction without modification and prove that it is a Levin reduction. This is captured by the following lemma.

Lemma A.8 (Karp–Levin reduction from HomAlgCSP to QRDH). *There exist universal constants $C > 1$ and $c \in (0, 1)$ for which the following holds. Let M, d, m be positive integers, let $1 \geq \alpha > \beta \geq 0$, and let λ, λ' be positive integers such that $\lambda' \mid \lambda$ and $2^\lambda \geq C(md/\alpha)^C$. Set*

$$\mathbb{K} = \mathbb{F}_{2^\lambda}, \quad T_Q := (M2^\lambda)^{O((md)^{O(1)})}.$$

There is a deterministic algorithm $\text{Red}_{\text{HA} \rightarrow \text{QR}}$ running in time T_Q . Given a HomAlgCSP instance $\mathcal{A}$ with parameters $(\mathbb{K}, M, k = 21, d, m, \alpha, \beta)$, it outputs a d_Q -uniform hypergraph

$$\text{Red}_{\text{HA} \rightarrow \text{QR}}(\mathcal{A}) = \mathcal{H}_\mathcal{A} = (V_\mathcal{A}, E_\mathcal{A})$$

that is an $(N_Q, M_Q, d_Q, r, \alpha/2, \beta_Q)$ -QRDH instance, where

$$N_Q, M_Q \leq T_Q, \quad d_Q = O(d), \quad r = 2^{\lambda'}, \quad \beta_Q = C\beta^c + 2^{-c\lambda}.$$

All hidden constants are universal and fixed by the reduction. Moreover, the reduction has the following Levin properties.

- **Witness map for completeness.** *There exists a deterministic algorithm $\text{Map}_{\text{HA} \rightarrow \text{QR}}$ running in time T_Q . Given $\mathcal{A}$ and any non-zero polynomial $\Phi \in \mathbb{K}[X_1, \dots, X_m]$ of total degree at most d that satisfies at least an α fraction of the constraints, it outputs a set $U_\Phi \subseteq V_\mathcal{A}$ satisfying*

$$|U_\Phi| \leq \frac{N_Q}{r} \quad \text{and} \quad e_{\mathcal{H}_\mathcal{A}}(U_\Phi) \geq \frac{\alpha}{2} \left(\frac{1}{r}\right)^{d_Q-1} M_Q.$$

- **Witness recovery for soundness.** *There exists a deterministic algorithm $\text{Ext}_{\text{HA} \rightarrow \text{QR}}$ running in time T_Q . Given $\mathcal{A}$ and any $U \subseteq V_\mathcal{A}$ satisfying*

$$e_{\mathcal{H}_\mathcal{A}}(U) > \left(\frac{|U|}{N_Q}\right)^{d_Q} M_Q + \beta_Q M_Q,$$

it outputs a non-zero polynomial $\Phi \in \mathbb{K}[X_1, \dots, X_m]$ of total degree at most $1000d$ that satisfies more than a β fraction of the constraints of $\mathcal{A}$.

Proof sketch. The main purpose of this lemma is to verify the witness map and witness recovery properties in Khot's reduction. We first recall the definition of quasi-random PCPs and their equivalence to QRDH.

Quasi-random PCPs and QRDH. Let V be the set of proof locations of a d_Q -query PCP, and let $Q_\rho \subseteq V$ be the query set selected by verifier randomness ρ . Khot calls such a PCP *quasi-random* if its query pattern has the following properties [Kho06b, Definition 1.8]:

- in the YES CASE, some distinguished half-set $U_0 \subseteq V$ contains all d_Q queries for at least a $(1 - o(1))2^{-(d_Q-1)}$ fraction of the tests;
- in the NO CASE, every half-set $U \subseteq V$ contains all d_Q queries for a $(1 \pm o(1))2^{-d_Q}$ fraction of the tests.

Thus the query pattern behaves like independent uniform queries in the NO CASE, but has a noticeably denser distinguished set in the YES CASE. This definition concerns only the query locations, not the verifier’s acceptance predicate.

QRDH is a hypergraph formulation of this property. Create one vertex for every coordinate in the PCP proof and one hyperedge Q_ρ for every verifier random string ρ . Then, for every $U \subseteq V$,

$$\frac{e_{\mathcal{H}}(U)}{|E|} = \Pr_{\rho}[Q_{\rho} \subseteq U].$$

This proves the equivalence between quasi-random PCPs and QRDH instances. Khot [Kho06b] constructed quasi-random PCPs for HomAlgCSP, and Hair and Sahai [HS26] interpreted this construction as a reduction from HomAlgCSP to QRDH. We next briefly recall Khot’s quasi-random PCP construction. We present the construction only at a high level, since our verification of the Levin property relies only on its completeness and soundness analysis.

To construct a quasi-random PCP for HomAlgCSP, Khot [Kho06b] first constructs an “outer verifier” whose PCP proof is over the large alphabet $\mathbb{K}$ rather than a binary alphabet. Consequently, if the $\mathbb{K}$ -symbols are interpreted directly as binary strings, the verifier’s queries are not quasi-random. Khot therefore composes the outer verifier with an inner verifier based on the Hadamard code, obtaining a PCP over a binary alphabet. Next we briefly describe its outer verifier.

Outer verifier V_{out} . The PCP proof is a table $f: \mathbb{K}^m \rightarrow \mathbb{K}$, intended to contain the full evaluations of Φ on every input. To verify this proof, the naive strategy is to randomly sample a constraint from the HomAlgCSP instance. However, for a fixed constraint, these query points lack the randomness required of a quasi-random PCP. To resolve this issue, Khot chooses a random degree- $(k+2)$ curve γ through these fixed query points. For an honest degree- d polynomial, $f \circ \gamma$ has degree at most $(k+2)d$, so $s := (k+2)d + 1$ random evaluations on the curve determine its values at the original points in the constraint. They further combine these curve-interpolation queries with Arora–Sudan’s low-degree testing to ensure f is indeed a low-degree polynomial. In total, the number of queries for the outer verifier is $2s$; see [Kho06b, Section 7.1 and Figure 2].

Outer soundness. The soundness of the outer verifier can be captured by the following lemma, which is a consequence of [Kho06b, Theorem 7.6 and 7.7] (restated in [KS16, Theorem 5.3]).

Lemma A.9 (Outer soundness). *There exist universal constants $c_0, c_1, c_2 > 1$ such that the following holds. Suppose V_{out} accepts a table $f: \mathbb{K}^m \rightarrow \mathbb{K}$ with probability $\tau > 0$, averaged over the constraints, and $|\mathbb{K}| \geq c_0(ms/\tau)^{c_0}$. Set*

$$\delta := \frac{(\tau/4)^{c_1}}{c_2}.$$

Then the list of all degree-at-most- s polynomials agreeing with f on at least a δ fraction of $\mathbb{K}^m$ has size at most $2/\delta$ and contains a non-zero polynomial satisfying at least a $\tau\delta/4$ fraction of the constraints.

Proof. For any such f , at least a $\tau/2$ fraction of constraints have conditional acceptance probability at least $\tau/2$ under V_{out} . Applying [Kho06b, Theorem 7.6], each such constraint is satisfied by a non-zero degree-at-most- s polynomial in the δ -agreement list, which has size at most $2/\delta$. By the pigeonhole principle, some non-zero polynomial satisfies at least $(\tau/2)/(2/\delta) = \tau\delta/4$ fraction of all constraints. $\square$

Inner verifier. The inner verifier encodes each symbol in the outer PCP proof using a Hadamard code over $\mathbb{F}_r$, where $r = 2^{\lambda'}$; Khot's original construction uses $r = 2$ and [KS16] extended to general $r = 2^{\lambda'}$. It reads one location from each of $4s$ encoded outer symbols, including additional line and curve samples for consistency. Thus $d_Q = 4s = O(d)$; see [Kho06b, Sections 9–10] and [KS16, Sections 5.2–5.3].

Inner soundness and decoding. To prove the soundness of the PCP obtained by composing the outer and inner verifiers, they show that any subset $U \subseteq V$ containing an unusually large fraction of hyperedges can be decoded into a PCP proof that the outer verifier accepts with non-trivial probability. The following lemma captures this Fourier-analytic decoding step. The lemma is abstracted from the proofs in [Kho06b, Lemma 10.3 and Section 10.5] and its extension to $\mathbb{F}_r$ and arbitrary densities in [KS16, Lemma 5.5 and Section 5.4.2].

Lemma A.10 (Inner soundness and decoding). *For every set $U \subseteq V$ of inner proof locations, define*

$$\eta := \Pr_{\rho}[Q_{\rho} \subseteq U] - \left(\frac{|U|}{|V|}\right)^{4s}.$$

Let $\ell = \lambda/\lambda'$ and identify $\mathbb{K}$ with $\mathbb{F}_r^{\ell}$ using the basis of the inner encoding. For each outer location $v \in \mathbb{K}^m$, define the indicator of U in its Hadamard block by

$$A_v(z) := \mathbf{1}[(v, z) \in U], \quad z \in \mathbb{F}_r^{\ell}.$$

For $a \in \mathbb{K} \cong \mathbb{F}_r^{\ell}$, define its Fourier coefficient

$$\widehat{A}_v(a) := \mathbb{E}_{z \leftarrow \mathbb{F}_r^{\ell}} \left[A_v(z) (-1)^{\text{Tr}_{\mathbb{F}_r/\mathbb{F}_2}(\langle a, z \rangle)} \right].$$

Sample the outer symbols independently according to

$$\Pr[f(v) = a] = \begin{cases} \widehat{A}_v(a)^2, & a \neq 0, \\ 1 - \sum_{b \in \mathbb{K} \setminus \{0\}} \widehat{A}_v(b)^2, & a = 0. \end{cases}$$

These distributions are computable deterministically from U in time $|\mathbb{K}|^{O(m)}$ and satisfy

$$\mathbb{E}_f[\Pr[V_{\text{out}}^f(\mathcal{A}) = 1]] \geq (\max\{\eta - \varepsilon_{\text{mix}}, 0\})^2, \quad \varepsilon_{\text{mix}} = O(s^2 |\mathbb{K}|^{-1/3}).$$

This lemma only gives a randomized $|\mathbb{K}|^{O(m)}$ -time witness recovery algorithm for the inner verifier. We observe that this procedure can be derandomized using conditional expectations.

Lemma A.11 (Deterministic inner decoding). *In the setting of Lemma A.10, let $\mathcal{D}_U$ be the product distribution over tables $f: \mathbb{K}^m \rightarrow \mathbb{K}$ defined there. There is a deterministic algorithm that, given U , runs in time $M|\mathbb{K}|^{O(ms)}$ and outputs a fixed table $f^*: \mathbb{K}^m \rightarrow \mathbb{K}$ satisfying*

$$\Pr[V_{\text{out}}^{f^*}(\mathcal{A}) = 1] \geq \mathbb{E}_{f \leftarrow \mathcal{D}_U} [\Pr[V_{\text{out}}^f(\mathcal{A}) = 1]] \geq (\max\{\eta - \varepsilon_{\text{mix}}, 0\})^2.$$

Proof. Fix the entries of f one at a time to maximize conditional expected acceptance; this cannot decrease the expectation. For each of the M constraints, a local test is specified by $O(s)$ query points in $\mathbb{K}^m$ and its checking parameters. According to the outer verifier in [Kho06b], this uses $O(ms)$ field elements in total. Hence there are at most $M|\mathbb{K}|^{O(ms)}$ tests. For each test, enumerate assignments to its at most $2s$ unfixed queried entries and sum their product probabilities over accepting assignments. Averaging over the verifier's test distribution computes the conditional expectation. Trying all $|\mathbb{K}|$ values for each of the $|\mathbb{K}|^m$ entries therefore takes

$$|\mathbb{K}|^{m+1} \cdot M|\mathbb{K}|^{O(ms)} \cdot |\mathbb{K}|^{2s} = M|\mathbb{K}|^{O(ms)}.$$

□

Witness map for completeness. The algorithm $\text{Map}_{\text{HA} \rightarrow \text{QR}}$ runs the honest prover on Φ and takes U_Φ to be the zero locations of its PCP, omitting zero-symbol blocks of the outer PCP. The guarantees follow from quasi-random PCP completeness [Kho06b, Lemma 10.2] and [KS16, Lemma 5.4].

Reed–Muller list decoding. The preceding Lemma A.11 gives a “cheating” PCP proof for the outer verifier, and Lemma A.9 shows that this cheating PCP proof is close to a suitable polynomial. We recover that polynomial using the global list-decoding algorithm of Pellikaan–Wu [PW04, Algorithm 3.1 and Theorem 3.1].

Lemma A.12 (Reed–Muller list decoding). *Let m, s be positive integers with $s < |\mathbb{K}|$, and let $\sqrt{s/|\mathbb{K}|} < \delta \leq 1$. Given the full table of $f: \mathbb{K}^m \rightarrow \mathbb{K}$, a deterministic algorithm lists all polynomials $\Phi \in \mathbb{K}[X_1, \dots, X_m]$ of total degree at most s satisfying*

$$\Pr_{v \leftarrow \mathbb{K}^m} [\Phi(v) = f(v)] \geq \delta$$

in time $|\mathbb{K}|^{O(m)}$.

Witness recovery for soundness. Let $U \subseteq V$ violate the QRDH NO CASE condition. By the aforementioned correspondence between QRDH and quasi-random PCP, this translates to the following inequality for the quasi-random PCP:

$$\Pr_\rho[Q_\rho \subseteq U] > \left(\frac{|U|}{|V|}\right)^{d_Q} + \beta_Q.$$

Use Lemma A.11 to construct a table f with outer acceptance $\tau \geq (\eta - \varepsilon_{\text{mix}})^2$. The premise gives $\eta > \max\{C\beta^c, |\mathbb{K}|^{-c}\}$. Choosing c small and then C large ensures

$$\eta > 2\varepsilon_{\text{mix}}, \quad |\mathbb{K}| \geq c_0(ms/\tau)^{c_0}, \quad \frac{\tau\delta}{4} > \beta,$$

where $\delta = (\tau/4)^{c_1}/c_2$ as in Lemma A.9. Moreover, $\tau \geq |\mathbb{K}|^{-2c}/4$ and $s = O(|\mathbb{K}|^{1/C})$, so

$$\frac{|\mathbb{K}|\delta^2}{s} = \Omega\left(|\mathbb{K}|^{1-4cc_1-1/C}\right) > 1$$

for these choices of constants. Thus Lemma A.12 applies. List the candidate polynomials, discard zero, and output one satisfying more than a β fraction of the constraints; Lemma A.9 guarantees that one exists. Its degree is at most $s \leq (k+3)d < 1000d$, and the total running time is $M|\mathbb{K}|^{O(ms)} \leq T_Q$. This defines the deterministic algorithm $\text{Ext}_{\text{HA} \rightarrow \text{QR}}$. $\square$

A.5 Hair–Sahai Vandermonde-Fortified Tensor Amplification

Before presenting the reduction in this subsection, we first compose the preceding chain of reductions from SAT.

Lemma A.13 (Karp–Levin reduction from SAT to QRDH, composition of Lemmas A.1, A.4, A.7 and A.8). *There exist an integer n_0 , and a deterministic algorithm $\text{Red}_{\text{SAT} \rightarrow \text{QRDH}}$ with the following property. For every $n \geq n_0$ and every SAT instance φ of size n , the algorithm runs in time*

$$T_{\text{QR}}(n) := \exp\left(2^{O(\sqrt{\log n} \log^4 \log n)}\right)$$

and outputs an $(N, M, D, r, \alpha, \beta)$ -QRDH instance

$$\text{Red}_{\text{SAT} \rightarrow \text{QRDH}}(\varphi) = \mathcal{H}_\varphi = (V, E),$$

where

$$N, M \leq T_{\text{QR}}(n), \quad D = O\left(\frac{\sqrt{\log n}}{\log^2 \log n}\right), \quad r = 2^{\Theta(\sqrt{\log n})}, \quad \alpha = \frac{1}{4}, \quad \beta = n^{-\Omega(\log \log n)}.$$

Moreover, the reduction has the following Levin properties.

- **Witness map for completeness.** *There exists a deterministic $T_{\text{QR}}(n)$ -time algorithm $\text{Map}_{\text{SAT} \rightarrow \text{QRDH}}$ such that, given φ and any satisfying assignment w , it outputs $U_w \subseteq V$ satisfying*

$$|U_w| \leq \frac{N}{r} \quad \text{and} \quad e_{\mathcal{H}_\varphi}(U_w) \geq \alpha r^{-(D-1)} M.$$

- **Witness recovery for soundness.** *There exists a deterministic $T_{\text{QR}}(n)$ -time algorithm $\text{Ext}_{\text{SAT} \rightarrow \text{QRDH}}$ such that, given φ and any $U \subseteq V$ satisfying*

$$e_{\mathcal{H}_\varphi}(U) > \left[\left(\frac{|U|}{N} \right)^D + \beta \right] \cdot M,$$

it outputs a satisfying assignment for φ .

Up to the $\log \log \log M$ factors suppressed in the overview of [HS26], these parameters are written there as

$$N \leq M^{O(1)}, \quad D = \log \log M, \quad r = \log M, \quad \beta = (\log M)^{-100 \log \log M}.$$

Proof sketch. Compose Lemmas A.1, A.4, A.7 and A.8. The witness map applies their maps in this order, while witness recovery applies their recovery algorithms in reverse order. For the HomAlgCSP input, $\log |\mathbb{K}| = \Theta(\log n \log \log n)$ and $\log(md) = O(\sqrt{\log n} \log^4 \log n)$, so the large-field condition in Lemma A.8 holds for sufficiently large n . Its completeness loss changes $1/2$ to $1/4$; the remaining parameter bounds and running time are those of [HS26, Theorems 2.1 and 5.1, Corollary 6.20]. $\square$

From QRDH to edge–vertex expansion. Hair–Sahai then view this QRDH instance as an edge–vertex expansion problem. To see this, view any matrix $\mathbf{P} \in \mathbb{F}^{M \times N}$ over a field $\mathbb{F}$ as a hypergraph whose rows are edges and whose columns are vertices: an edge i contains a vertex j if and only if $P_{i,j} \neq 0$. For any set of edges $S \subseteq [M]$, define

$$\Gamma_{\mathbf{P}}(S) := \{j \in [N] : P_{i,j} \neq 0 \text{ for some } i \in S\}.$$

Thus $|\Gamma_{\mathbf{P}}(S)|$ is the number of vertices touched by S . Let $\mathcal{H}_\varphi$ be the QRDH instance from Lemma A.13, let $\mathbf{P} \in \{0,1\}^{M \times N}$ be its edge–vertex incidence matrix, and set

$$k := \frac{N}{r}, \quad h := \left\lceil \alpha r^{-(D-1)} M \right\rceil.$$

Then $\mathbf{P}$ is automatically an instance of the following edge–vertex expansion problem.

- YES CASE: There exists a subset of hyperedges $S \subseteq [M]$ such that

$$|S| = h \quad \text{and} \quad |\Gamma_{\mathbf{P}}(S)| \leq k.$$

This follows from the YES CASE of the QRDH instance: the dense sub-hypergraph induced by U_w contains sufficiently many hyperedges supported on the small vertex set U_w , and hence yields a set of hyperedges with small edge–vertex expansion.

- NO CASE: Every $S \subseteq [M]$ satisfying $|S| > \beta M$ has

$$|\Gamma_{\mathbf{P}}(S)| \geq N \left(\frac{|S|}{M} - \beta \right)^{1/D}.$$

This follows by applying the soundness of NO CASE for the QRDH instance to $U := \Gamma_{\mathbf{P}}(S)$.

To understand the edge–vertex expansion gap between the YES CASE and the NO CASE above, we examine the expansion of NO CASE when $|S| = h$. Since $\beta = o(\alpha r^{-(D-1)})$, we have $h > \beta M$ for sufficiently large n , so the NO CASE applies to sets of h edges. Ignoring the harmless rounding in h , it gives

$$|\Gamma_{\mathbf{P}}(S)| \geq k(\alpha r - \beta r^D)^{1/D}.$$

For the above parameters $\alpha = 1/4$, $r = \log M$, and $D = \log \log M$, the last factor is $2 - o(1)$. Thus every set of h edges touches at least $(2 - o(1))k$ vertices, as in [HS26, Section 2.1].

Amplify the edge–vertex expansion. Hair–Sahai then amplify the edge–vertex expansion gap from $2 - o(1)$ to $(2 - o(1))^q$. They observe that direct q -fold tensoring gives the desired amplification when every factor S_i of a product set $S_1 \times \cdots \times S_q$ is sufficiently large, since the NO CASE above guarantees an expansion lower bound only for sufficiently large edge sets $|S|$. To handle small factors, Hair–Sahai fortify the tensor product with Vandermonde matrices. The resulting amplification is summarized in the following lemma.

Lemma A.14 (Karp–Levin Vandermonde-fortified tensor amplification). *Let $\mathcal{H} = (V, E)$ be a D -uniform hypergraph with $D \geq 1$, N vertices and $M \geq 1$ indexed edges, and let $\mathbf{P} \in \{0,1\}^{M \times N}$ be its incidence matrix. Let $\beta \in (0,1)$ be a real. Let $t, q \geq 1$ be integers with*

$$t \mid M \quad \text{and} \quad 1/t > \beta.$$

There is a deterministic algorithm Red_{VF} that takes $\mathbf{P}, t, q$ as input, and outputs

$$\text{Red}_{\text{VF}}(\mathbf{P}; t, q) = (\mathbf{A}, \mathbf{Q}), \quad \mathbf{Q} = \mathbf{P}^{\otimes q},$$

where

$$\mathbf{A} \in \mathbb{Z}^{M^q \times qM^q/t}, \quad \mathbf{Q} \in \{0, 1\}^{M^q \times N^q}, \quad \|\mathbf{A}\|_\infty < 3M^q.$$

All algorithms in this lemma run in time polynomial in q, M^q, N^q . The reduction has the following Levin-reduction properties.

- **Witness map for completeness.** There exists a deterministic algorithm Map_{VF} that, given $\mathbf{P}, t, q$ and any nonempty set $F \subseteq E$ with $|F| = h$ and $|\Gamma_{\mathbf{P}}(F)| \leq k$, outputs $S_F = \text{Map}_{\text{VF}}(\mathbf{P}, F; t, q) \subseteq E^q$ satisfying

$$|S_F| = h^q, \quad |\Gamma_{\mathbf{Q}}(S_F)| \leq k^q, \quad |\Gamma_{\mathbf{A}}(S_F)| \leq qh^{q-1} \frac{M}{t}.$$

- **Witness recovery for soundness.** There exists a deterministic algorithm Ext_{VF} that, given the parameters $\mathbf{P}, t, q, \beta$ and a non-empty set $S \subseteq [M^q]$ such that there exists $\mathbf{x} \in \mathbb{Q}^{1 \times M^q} \setminus \{\mathbf{0}\}$ with $\mathbf{x}\mathbf{A} = \mathbf{0}$, $\text{Supp}(\mathbf{x}) = S$ and

$$|\Gamma_{\mathbf{Q}}(S)| < N^q \left(\frac{(1 - \beta t)^q |S|}{M^q} \right)^{1/D},$$

outputs a vertex set $U = \text{Ext}_{\text{VF}}(\mathbf{P}, S; t, q, \beta) \subseteq V$ satisfying

$$e_{\mathcal{H}}(U) > M \left(\frac{|U|}{N} \right)^D + \beta M.$$

This combines [HS26, Theorem 4.9 and Lemma 4.7], with the witness map and witness recovery added.

Proof. We briefly recall the construction in [HS26, Definition 4.4].

The matrices $\mathbf{Q}$ and $\mathbf{A}$. The rows of $\mathbf{Q} = \mathbf{P}^{\otimes q}$ are indexed by tuples $\mathbf{e} = (e_1, \dots, e_q) \in E^q$, and its columns are indexed by tuples $\mathbf{v} = (v_1, \dots, v_q) \in V^q$. The entry $Q_{\mathbf{e}, \mathbf{v}}$ is non-zero exactly when $v_i \in e_i$ for every $i \in [q]$. Thus, the $\mathbf{e}$ -th row of $\mathbf{Q}$ has 1's in the columns indexed by $e_1 \times \dots \times e_q$ and 0's elsewhere.

An *axis-parallel line* in E^q fixes all but one edge coordinate and lets the remaining coordinate range over E . For example, when $q = 2$, these are the rows and columns of the grid $E \times E$. There are qM^{q-1} such indexed lines, each containing M edges.

Choose a prime $M^q < p_V < 3M^q$ and form the reduced Vandermonde matrix

$$\mathbf{V} \in \mathbb{Z}^{(p_V-1) \times M/t}, \quad V_{i,j} = i^{j-1} \bmod p_V,$$

where each entry is its least non-negative residue. Any $s \leq M/t$ rows are linearly independent over $\mathbb{Q}$: their first s columns form a square Vandermonde matrix invertible modulo p_V , and hence also over $\mathbb{Q}$ [HS26, Lemma 4.3]. Label the first M^q rows of $\mathbf{V}$ by E^q . For each line L , form

$\mathbf{A}^{(L)} \in \mathbb{Z}^{M^q \times M/t}$ by retaining these rows on L and setting all other rows to zero. Its non-zero part has M rows and M/t columns. Horizontally concatenate all these blocks to obtain $\mathbf{A}$. This gives the claimed dimensions and entry bound.

Call $S \subseteq E^q$ (t, q) -legal if every axis-parallel line L satisfies $|S \cap L| = 0$ or $|S \cap L| \geq M/t$. If $S = \text{Supp}(\mathbf{x})$ is not legal, some line L has $0 < |S \cap L| < M/t$. Then $\mathbf{x}\mathbf{A}^{(L)}$ is a non-zero combination of fewer than M/t independent Vandermonde rows, so it cannot vanish. Consequently,

$$\mathbf{x}\mathbf{A} = \mathbf{0} \implies \text{Supp}(\mathbf{x}) \text{ is } (t, q)\text{-legal}.$$

Witness map for completeness. The algorithm Map_{VF} outputs $S_F = F^q$. These $|F^q| = h^q$ rows of $\mathbf{Q}$ have non-zero entries in exactly the columns indexed by $\Gamma_{\mathbf{P}}(F)^q$, which contains at most k^q vertices. This proves $|\Gamma_{\mathbf{Q}}(S_F)| \leq k^q$.

We now bound $|\Gamma_{\mathbf{A}}(S_F)|$. Recall that $\mathbf{A} = [\dots | \mathbf{A}^{(L)} | \dots]$, where L ranges over all indexed axis-parallel lines. The rows indexed by $S_F = F^q$ meet exactly qh^{q-1} such lines: for each varying coordinate $i \in [q]$, the remaining $q-1$ coordinates can be fixed in h^{q-1} ways. Since each block $\mathbf{A}^{(L)}$ has width M/t , $|\Gamma_{\mathbf{A}}(S_F)| \leq qh^{q-1} \cdot M/t$.

Recall: the proof of expansion amplification. Before specifying Ext_{VF} , we recall the expansion bound [HS26, Theorem 4.9]: it proves that if $\mathcal{H}$ is a NO CASE instance of QRDH, then for every (t, q) -legal set $S \subseteq E^q$, we have

$$|\Gamma_{\mathbf{Q}}(S)| \geq N^q \left(\frac{(1 - \beta t)^q |S|}{M^q} \right)^{1/D}. \quad (2)$$

Given any (t, q) -legal S that violates this inequality, our witness recovery algorithm Ext_{VF} essentially traces the proof backward to find a violation of the QRDH NO CASE inequality.

Now we recall the proof in [HS26, Theorem 4.9]. Assume the QRDH soundness inequality holds.

Base Case $q = 1$. For any $F \subseteq E$ with $|F| \geq M/t$, we have

$$|F| \leq M \left(\frac{|\Gamma_{\mathbf{P}}(F)|}{N} \right)^D + \beta M \leq M \left(\frac{|\Gamma_{\mathbf{P}}(F)|}{N} \right)^D + \beta t |F|.$$

Moving the additive error term to the left, we obtain

$$|F| \leq \frac{M}{N^D(1 - \beta t)} |\Gamma_{\mathbf{P}}(F)|^D. \quad (3)$$

This proves the expansion bound (2) for $q = 1$.

Induction on q . For $q \geq 2$, group coordinates $2, \dots, q$ into one block. Let $S \subseteq E \times E^{q-1}$ be (t, q) -legal. For $e \in E$ and $\mathbf{v} \in V^{q-1}$, define

$$S_e := \{\mathbf{e}' : (e, \mathbf{e}') \in S\}, \quad X_{\mathbf{v}} := \{v : (v, \mathbf{v}) \in \Gamma_{\mathbf{Q}}(S)\}, \quad T_e := \{\mathbf{v} : e \subseteq X_{\mathbf{v}}\}.$$

Here T_e consists of the tuples paired in $\Gamma_{\mathbf{Q}}(S)$ with every vertex of e . Each S_e is $(t, q-1)$ -legal, and its neighborhood in the remaining coordinates is contained in T_e . The induction hypothesis

therefore gives

$$|S| = \sum_{e \in E} |S_e| \leq \left(\frac{M}{N^D(1-\beta t)} \right)^{q-1} \sum_{e \in E} |T_e|^D. \quad (4)$$

To account for overlap between the T_e 's, count the sum on the right by choosing the tuples first:

$$\sum_{e \in E} |T_e|^D = \sum_{\mathbf{v}_1, \dots, \mathbf{v}_D \in V^{q-1}} e_{\mathcal{H}} \left(\bigcap_{i=1}^D X_{\mathbf{v}_i} \right). \quad (5)$$

Both sides count an edge e and an ordered tuple $(\mathbf{v}_1, \dots, \mathbf{v}_D)$ such that $e \subseteq X_{\mathbf{v}_i}$ for every i .

If $X_{\mathbf{v}}$ is nonempty, choose $(e, \mathbf{e}') \in S$ whose last $q-1$ coordinates touch $\mathbf{v}$. Legality gives at least M/t choices of e' with $(e', \mathbf{e}') \in S$, all contained in $X_{\mathbf{v}}$. Thus $e_{\mathcal{H}}(X_{\mathbf{v}}) \geq M/t$. Applying QRDH soundness and absorbing $\beta M \leq \beta t e_{\mathcal{H}}(X_{\mathbf{v}})$ gives $e_{\mathcal{H}}(X_{\mathbf{v}}) \leq \frac{M}{N^D(1-\beta t)} |X_{\mathbf{v}}|^D$, also valid when $X_{\mathbf{v}}$ is empty. Consequently,

$$e_{\mathcal{H}} \left(\bigcap_{i=1}^D X_{\mathbf{v}_i} \right) \leq \min_i e_{\mathcal{H}}(X_{\mathbf{v}_i}) \leq \frac{M}{N^D(1-\beta t)} \prod_{i=1}^D |X_{\mathbf{v}_i}|. \quad (6)$$

Substituting into (5), we obtain

$$\begin{aligned} \sum_{e \in E} |T_e|^D &\leq \frac{M}{N^D(1-\beta t)} \sum_{\mathbf{v}_1, \dots, \mathbf{v}_D \in V^{q-1}} \prod_{i=1}^D |X_{\mathbf{v}_i}| \\ &= \frac{M}{N^D(1-\beta t)} \left(\sum_{\mathbf{v} \in V^{q-1}} |X_{\mathbf{v}}| \right)^D = \frac{M}{N^D(1-\beta t)} |\Gamma_{\mathbf{Q}}(S)|^D. \end{aligned}$$

Together with (4), this gives

$$|S| \leq \frac{M^q}{N^{qD}(1-\beta t)^q} |\Gamma_{\mathbf{Q}}(S)|^D,$$

which rearranges to (2); see [HS26, Claims 4.11–4.13].

Witness recovery for soundness. We now drop the assumption that $\mathcal{H}$ is a NO CASE instance and make the contrapositive of [HS26, Theorem 4.9] constructive. By the premise on S , S must be (t, q) -legal. Our $\text{Ext}_{\mathbf{V}_F}$ starts with S , and recursively maintains a (t, j) -legal set $S \subseteq E^j$ ($j = q, q-1, \dots, 1$), with

$$|S| > \left(\frac{M}{N^D(1-\beta t)} \right)^j |\Gamma_{\mathbf{P}^{\otimes j}}(S)|^D, \quad (7)$$

starting with $j = q$. If $j = 1$, output $U := \Gamma_{\mathbf{P}}(S)$. This satisfies the requirements of the lemma because $|S| \geq M/t$ and the premise of witness recovery for NO CASE gives

$$e_{\mathcal{H}}(\Gamma_{\mathbf{P}}(S)) \geq |S| > M \left(\frac{|\Gamma_{\mathbf{P}}(S)|}{N} \right)^D + \beta M.$$

For $j > 1$, compute the first-coordinate “edge slices” S_e and their neighborhoods. If some slice satisfies

$$|S_e| > \left(\frac{M}{N^D(1 - \beta t)} \right)^{j-1} |\Gamma_{\mathbf{P}^{\otimes(j-1)}}(S_e)|^D, \quad (8)$$

recurse on that legal slice with $j - 1$ coordinates. Otherwise, for each $\mathbf{v} \in V^{j-1}$, compute the “vertex slice” $X_{\mathbf{v}} = \{v \in V : (v, \mathbf{v}) \in \Gamma_{\mathbf{P}^{\otimes j}}(S)\}$ and test it for

$$e_{\mathcal{H}}(X_{\mathbf{v}}) > M \left(\frac{|X_{\mathbf{v}}|}{N} \right)^D + \beta M. \quad (9)$$

If this inequality holds, then output $U := X_{\mathbf{v}}$.

At least one must pass this test: if none did, then we know Equation (8) and Equation (9) don’t hold for all “edge slices” and “vertex slices”, and the double counting identity (Equation (5)) would contradict the maintained strict inequality Equation (7).

The recursion follows one slice S_e satisfying Equation (8) at each level and has depth at most q . Each step constructs explicit slices $S_e, X_{\mathbf{v}}$ and counts $|\Gamma_{\mathbf{P}^{\otimes(j-1)}}(S_e)|, e_{\mathcal{H}}(X_{\mathbf{v}})$. Note that we do not enumerate the D -tuples in the double-counting identity. These operations take polynomial time in the stated quantities. This proves the witness-recovery guarantee. $\square$

A.6 From Edge–Vertex Expansion Problem to the Hair–Sahai Matrix

We prove Theorem 5.2 in this subsection. This theorem proves that the reduction from SAT to Hair–Sahai’s matrix $\mathbf{C}$ in [HS26, Section 5] is indeed a Levin reduction.

Theorem A.15 (Karp–Levin reduction from SAT to the Hair–Sahai matrix, Theorem 5.2 Restated). *There exists a deterministic algorithm Red_{HS} with the following property. For all sufficiently large n and every SAT instance φ of size n , the algorithm outputs*

$$\text{Red}_{\text{HS}}(\varphi) = (\mathbf{C}_{\varphi}, h_{\varphi}), \quad \mathbf{C}_{\varphi} \in \mathbb{Z}^{M_{\varphi} \times N_{\varphi}},$$

where

$$M_{\varphi}, N_{\varphi} \leq T_{\text{HS}}(n) = \exp\left(2^{\sqrt{\log n}(\log \log n)^{O(1)}}\right), \quad 1 \leq h_{\varphi} \leq M_{\varphi}/2, \quad \|\mathbf{C}_{\varphi}\|_{\infty} \leq M_{\varphi}.$$

The reduction runs in time $T_{\text{HS}}(n)$. Moreover, the following two properties hold.

- **Witness map for completeness.** *There exists a deterministic algorithm Map_{HS} . Given φ and any satisfying assignment w , it runs in time $T_{\text{HS}}(n)$ and outputs a nonempty set $S \subseteq [M_{\varphi}]$ satisfying*

$$|S| \leq h_{\varphi} \quad \text{and} \quad |\Gamma_{\mathbf{C}_{\varphi}}(S)| < |S|.$$

- **Witness recovery for soundness.** *There exists a deterministic algorithm Ext_{HS} such that, given φ and any nonempty set $S \subseteq [M_{\varphi}]$ satisfying*

$$|S| < 2h_{\varphi} \quad \text{and} \quad \text{rank}_{\mathbb{Q}}((\mathbf{C}_{\varphi})_S) < |S|,$$

it outputs a satisfying assignment for φ in time $T_{\text{HS}}(n)$.

Proof sketch. The instance map Red_{HS} is already given by [HS26, Section 5], which turns the expansion bound in Lemma A.14 into a gap in the support size of non-zero kernel vectors. The main purpose of this theorem is to show that the reduction also gives a witness map and a witness recovery algorithm. We first briefly recall the construction of $\mathbf{C}$ in [HS26], and then verify the witness map and witness recovery property. Note that all parameters we use are identical to [HS26] and hence we omit them.

Construction of the matrix $\mathbf{C}$ in Hair–Sahai. First apply Lemma A.13 to φ to obtain $\mathcal{H}^-$ with vertex set V , N vertices, M^- edges and parameters D, r, α, β . Then we will apply Lemma A.14 to amplify the edge–vertex expansion. We omit the concrete parameter choice of t, q used by [HS26]. To ensure the number of hyperedges is a multiple of t , Hair–Sahai replicates each indexed edge of $\mathcal{H}^-$ $n^2 t N$ times, obtaining a hypergraph $\mathcal{H} = (V, E)$ with $M = n^2 t N M^-$ edges. This ensures $t \mid M$, without changing the fraction of edges contained in any subset of vertices.

Apply Lemma A.14 to the incidence matrix $\mathbf{P}$ of $\mathcal{H}$ to obtain $(\mathbf{A}, \mathbf{Q}) = \text{Red}_{\text{VF}}(\mathbf{P}; t, q)$. Hair–Sahai retains $\mathbf{A}$ and replaces each column of $\mathbf{Q}$ with a block matrix. For each new block, use reduced Vandermonde matrices modulo a prime in $(M^q, 3M^q)$, as in the preceding proof. Index their first M^q rows by E^q in the same order as $\mathbf{A}$ and $\mathbf{Q}$. Take the first M^q rows of a width- w_R reduced Vandermonde matrix. We omit the parameter choice of w_R in Hair–Sahai. For each tensor vertex $\mathbf{v} \in V^q$, they form a matrix $\mathbf{R}^{(\mathbf{v})} \in \mathbb{Z}^{M^q \times w_R}$ by retaining the $\mathbf{e}$ -th row of this reduced Vandermonde matrix when $Q_{\mathbf{e}, \mathbf{v}} = 1$ and replacing it by a zero row otherwise. Thus $\mathbf{R}^{(\mathbf{v})}$ is supported exactly on the edges incident to $\mathbf{v}$. Let $\mathbf{R}$ be the horizontal concatenation of these N^q blocks. The Vandermonde row-independence property will force each touched vertex to be incident to at least w_R supported rows.

Let $\mathbf{W}$ consist of the first M^q rows of a width- h_φ/n reduced Vandermonde matrix. We will set h_φ such that $n \mid h_\varphi$ later. This block rules out non-zero kernel vectors supported on fewer than h_φ/n rows. Set

$$\tilde{\mathbf{C}} := \mathbf{A} \parallel \mathbf{R} \parallel \mathbf{W} \in \mathbb{Z}^{M^q \times (qM^q/t + N^q w_R + h_\varphi/n)}.$$

All entries have magnitude less than $3M^q$. To bound them by the number of rows, pad with an identity block:

$$\mathbf{C}_\varphi := \begin{bmatrix} \tilde{\mathbf{C}} & 0 \\ 0 & I_{2M^q} \end{bmatrix}.$$

It has $M_\varphi = 3M^q$ rows and $\|\mathbf{C}_\varphi\|_\infty < M_\varphi$. Both matrix dimensions and the construction time are bounded by $T_{\text{HS}}(n)$. Finally, the reduction chooses the output threshold h_φ as a positive integer satisfying

$$h^q \leq h_\varphi \leq (2\alpha r^{-(D-1)} M)^q, \text{ where } h := \lceil \alpha r^{-(D-1)} M \rceil.$$

We proceed to prove that this reduction from SAT to the matrix $\mathbf{C}_\varphi$ described above is indeed a Levin reduction.

Witness map for completeness. Given a satisfying assignment w , apply $\text{Map}_{\text{SAT} \rightarrow \text{QRDH}}$ to obtain $U_w \subseteq V$ of size at most $k := N/r$. After replication, U_w contains at least h indexed edges; select a set F_w of exactly h such edges. Set $S = \text{Map}_{\text{VF}}(\mathbf{P}, F_w; t, q) = F_w^q$. Then $|S| = h^q \leq h_\varphi$, and

Lemma A.14 gives

$$|\Gamma_{\tilde{\mathbf{C}}}(S)| \leq qh^{q-1}\frac{M}{t} + k^q w_R + \frac{h_\varphi}{n}.$$

The three terms count the columns contributed by $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{W}$, respectively. For $\mathbf{R}$, each of the at most k^q columns of $\mathbf{Q}$ touched by S contributes a block of width w_R . Hair–Sahai’s [HS26, Claim 5.5] proves that, by their parameter choice, each term is $o(|S|/\sqrt{n})$. This proves $|\Gamma_{\tilde{\mathbf{C}}}(S)| < |S|$ for all sufficiently large n .

Our Map_{HS} outputs S as a subset of the first M^q rows of $\mathbf{C}_\varphi$. The identity padding adds no columns touched by S .

Witness recovery for soundness. Given S satisfying the witness-recovery premise, since the rows of $(\mathbf{C}_\varphi)_S$ are linearly dependent over $\mathbb{Q}$, we can always shrink S to a minimal linearly dependent subset using rank tests. Then there exists $\mathbf{x} \in \mathbb{Q}^{1 \times M_\varphi} \setminus \{\mathbf{0}\}$ with

$$\mathbf{x}\mathbf{C}_\varphi = \mathbf{0}, \quad \text{Supp}(\mathbf{x}) = S.$$

Note that we do not need to explicitly compute this $\mathbf{x}$. Throughout this proof we only need the existence of such a $\mathbf{x}$. Write $\mathbf{x} = (\mathbf{y}, \mathbf{z})$, where $\mathbf{y}$ consists of its first M^q coordinates. The identity block forces $\mathbf{z} = \mathbf{0}$, so $\mathbf{y} \neq \mathbf{0}$ and $\mathbf{y}\tilde{\mathbf{C}} = \mathbf{0}$. Since S is minimal, $S \subseteq [M^q]$. Since any set of fewer than h_φ/n rows of $\mathbf{W}$ is linearly independent, $\mathbf{y}\mathbf{W} = \mathbf{0}$ and the premise of witness recovery imply

$$\frac{h_\varphi}{n} \leq |S| < 2h_\varphi$$

[HS26, Claim 5.7]. We also have $\mathbf{y}\mathbf{A} = \mathbf{0}$.

It remains to show that S touches too few columns of $\mathbf{Q}$, and once that is proven, we can apply the witness-recovery algorithm of Lemma A.14 to obtain a witness recovery algorithm. To show $|\Gamma_{\mathbf{Q}}(S)|$ is small, first observe that, for any fixed tensor vertex $\mathbf{v} \in N^q$ touched by S , if the $\mathbf{v}$ -th column of $\mathbf{Q}_S$ contains fewer than w_R of 1’s, then the corresponding non-zero rows of $\mathbf{R}^{(\mathbf{v})}$ would be independent, contradicting $\mathbf{y}\mathbf{R}^{(\mathbf{v})} = \mathbf{0}$. Thus every column $\mathbf{v} \in \Gamma_{\mathbf{Q}}(S)$ contains at least w_R of 1’s. On the other hand, every row of $\mathbf{Q}$ contains exactly D^q 1’s, so the total number of 1’s in submatrix $\mathbf{Q}_S$ is $|S|D^q$. Hence,

$$|\Gamma_{\mathbf{Q}}(S)| \leq |S|D^q/w_R < 2h_\varphi D^q/w_R.$$

The parameter choices in Hair–Sahai give the following (See [HS26, Claim 5.8 and Proposition 5.10]):

$$\begin{aligned} \frac{2h_\varphi D^q}{w_R} &< N^q \left(\frac{(1 - \beta t)^q h_\varphi}{n M^q} \right)^{1/D} \\ &\leq N^q \left(\frac{(1 - \beta t)^q |S|}{M^q} \right)^{1/D}. \end{aligned}$$

Thus S satisfies the witness-recovery premise of Lemma A.14. Applying $\text{Ext}_{\text{VF}}(\mathbf{P}, S; t, q, \beta)$ produces a vertex set $U \subseteq V$ violating QRDH soundness. Since replication preserves the fraction of edges contained in U , the same set violates the bound for $\mathcal{H}^-$. Finally, $\text{Ext}_{\text{SAT} \rightarrow \text{QRDH}}(\varphi, U)$ returns a satisfying assignment for φ . $\square$