Hardness and Algorithms for Batch LPN under Dependent Noise

Xin Li* Songtao Mao† Zhaienhe Zhou‡

Abstract

We study the Batch Learning Parity with Noise (LPN) variant, where the oracle returns k samples in a batch, and draws the noise vector from a joint noise distribution $\mathcal{D}$ on $\mathbb{F}_2^k$ (instead of i.i.d.). This model captures a broad range of correlated or structured noise patterns studied in cryptography and learning theory, and was formally defined in recent work by Golowich, Moitra, and Rohatgi (FOCS 2024). Consequently, understanding which distributions preserve the hardness of LPN has become an important question.

On the hardness side, we design several reductions from standard LPN to Batch LPN. Our reductions provide a more comprehensive characterization of hard distributions. Specifically, we show that a Batch LPN instance is as hard as standard LPN with noise rate $\eta := \frac{1}{2} - \varepsilon$ provided that its noise distribution $\mathcal{D}$ satisfies one of the following:

1. The noise distribution $\mathcal{D}$ satisfies a mild Fourier-analytic condition (specifically, $\sum_{s\neq 0} |\widehat{P}_{\mathcal{D}}(s)| \leq 2\varepsilon$).
2. The noise distribution $\mathcal{D}$ is $\Omega(\eta \cdot k2^{-k})$-dense (i.e., every error pattern occurs with probability at least $\Omega(\eta \cdot k2^{-k})$) for $\eta < 1/k$.
3. The noise distribution $\mathcal{D}$ is a δ-Santha-Vazirani source. Our reduction improves the allowable bias δ from $O(2^{-k}\varepsilon)$ (in Golowich et al.) to $O(2^{-k/2}\varepsilon)$.

On the algorithmic side, we design an algorithm for solving Batch LPN whenever the noise distribution assigns sufficiently small probability to at least one point, which gives an algorithm–hardness separation for Batch LPN. Our algorithm can be seen as an extension of Arora and Ge's (ICALP 2011) linearization attack.

Our reduction is based on random affine transformations, developed and analyzed through the lens of Fourier analysis, providing a general framework for studying various LPN variants.

1 Introduction

The Learning Parity with Noise (LPN) problem asks an algorithm to solve a noisy system of linear binary equations (or equivalently, decode from random linear codes). The problem, along with its variants, is a central topic in post-quantum cryptography, coding theory, and learning theory. As a well-established security assumption, LPN has various applications in cryptography [Pie12], to name a few, the construction of pseudorandom generators [BFKL94, AIK08], pseudorandom functions [GGM86], hashing [YZW+17], and pseudorandom codes [AAC+25].

Formally, we denote the computational problem of recovering the secret $\mathbf{sk}$ from samples drawn from $\mathsf{LPN}_{n,\eta}(\mathbf{sk})$ (defined below) as *Search* $\mathsf{LPN}_{n,\eta}$ (or simply $\mathsf{LPN}_{n,\eta}$).

Definition 1.1 (Standard LPN distribution)**.** Let $n \in \mathbb{N}$ and let $\mathbf{sk} \in \mathbb{F}_2^n$ be the hidden secret vector. We define the $\mathsf{LPN}_{n,\eta}(\mathbf{sk})$ distribution as the distribution of the random variable

$$(\mathbf{u}, \langle \mathbf{u}, \mathbf{sk} \rangle + e),$$

where $\mathbf{u} \sim \mathrm{Unif}(\mathbb{F}_2^n)$ and $e \sim \mathrm{Ber}(\eta)$ are independent.

*lixints@cs.jhu.edu, Department of Computer Science, Johns Hopkins University. Supported in part by NSF Award CCF-2127575

†smao13@jhu.edu, Department of Computer Science, Johns Hopkins University. Supported in part by NSF Award CCF-2127575

‡zhaienhezhou@gmail.com, School of the Gifted Young & College of Computer Science, University of Science and Technology of China

Despite its simple algebraic structure and extensive study, LPN remains computationally hard. The most efficient algorithm for solving standard LPN with a constant noise rate remains the classical BKW algorithm [BKW03], which requires roughly $2^{O(n/\log n)}$ time and samples. Although subsequent works have improved BKW regarding sample complexity [Lyu05] and practical runtime [LF06], these optimizations do not improve the asymptotic time complexity.

Even in the low-noise regime, where the noise rate η approaches zero, the best known algorithm still requires $e^{O(\eta n)}$ time. This approach essentially involves repeatedly drawing samples until n noiseless equations are found, which occurs with probability $(1-\eta)^n \approx e^{-\eta n}$. No polynomial-time algorithm is known for any nontrivial noise level; it is conjectured that standard LPN remains hard even for noise rates as low as $\eta = O(\log^2 n/n)$ [BLVW19].

LPN Variants. To better understand the hardness landscape of LPN and to construct cryptographic primitives with specific properties, a variety of LPN modifications have been studied. On the algorithmic side, several works have developed more efficient procedures for specialized variants, including Sparse LPN [FKO06, BHLM25] LSPN [Val15, CSZ25], and batch LPN with structured noise under appropriate conditions [AG11]. On the hardness side, many variants of LPN have yielded implications in a wide range of settings, for example, researchers have also conjectured intractability for additional variants of LPN—the dense–sparse LPN problem [DJ24] for cryptographic constructions; recent work on Batch LPN with mildly dependent noise [GMR24b] establishes hardness results that give reductions for several learning problems. Moreover, [YZ16] showed that LPN retains its hardness even in the presence of certain auxiliary information, by providing reductions from these augmented settings to the base LPN problem. Very recently, [BBTV25] proved a lower bound on the running time for Sparse LPN algorithms with an unbounded number of samples.

Batch LPN and Dependent Noise. The Batch LPN variant, in which a batch of k samples are produced simultaneously, and their noise vector $\mathbf{e} = (e_1, e_2 \ldots, e_k)^\top$ follows a joint distribution $\mathcal{D}$ over $\mathbb{F}_2^k$. This model naturally captures scenarios where randomness is reused across queries, or where the noise terms exhibit correlations rather than being independent. Such settings arise both in cryptographic constructions and in learning-theoretic models of weak or structured randomness [GMR24a]. Understanding the robustness of standard hardness assumptions—such as LPN and LWE—under these types of dependent-noise transformations has become increasingly important. For example, [BD20] shows the hardness of LWE with a non-uniform but high-entropy secret distribution. Similar hardness has also been conjectured in LPN [YYLG16]. Other robustness results include [BHK+21], which examined LPN in a physical noise setting, where error bits are not perfectly random but arise from a real source with potential biases or correlations.

Similar to the standard LPN problem, we denote the search version of Batch LPN as the computational problem of recovering $\mathbf{sk}$ from the distribution $\mathsf{LPN}_{n,k,\mathcal{D}}(\mathbf{sk})$ (described below) as *Search* $\mathsf{LPN}_{n,k,\mathcal{D}}$ (or simply $\mathsf{LPN}_{n,k,\mathcal{D}}$).

Definition 1.2 (Batch LPN distribution)**.** Let $n, k \in \mathbb{N}, \mathbf{sk} \in \mathbb{F}_2^n$, and let $\mathcal{D}$ be a distribution over $\mathbb{F}_2^k$. The *Batch LPN* distribution $\mathsf{LPN}_{n,k,\mathcal{D}}(\mathbf{sk})$ is defined as

$$(\mathbf{u}_i, \langle \mathbf{u}_i, \mathbf{sk} \rangle + e_i)_{i=1}^k,$$

where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ are i.i.d.uniform samples from $\mathbb{F}_2^n$, and $(e_1, \ldots, e_k) \sim \mathcal{D}$. The random variables $(\mathbf{u}_i)_{i=1}^k$ and $(e_i)_{i=1}^k$ are independent.

Arora and Ge [AG11] initiated the line of study on Batch LPN. They applied linearization techniques on a special class of distribution $\mathcal{D}$ they termed "Structured Noise", in which the support of the distribution $\mathcal{D}$ must fall in the zeros of a predefined non-zero polynomial over $\mathbb{F}_2[e_1, e_2, \ldots e_k]$. An example of such error distribution would be any $\mathcal{D}$ such that the number of errors in $e_1, e_2, \ldots e_k$ for $\mathbf{e} \sim \mathcal{D}$ is always smaller than $k/3$. While such error distribution looks similar to independent noise, their framework can achieve $n^{O(k)}$ runtime on such Batch LPN instances, which is polynomial when k is a constant.

While [AG11] can be interpreted as showing LPN to be tractable for certain noise distributions with strong dependency and within a small batch, Golowich, Moitra, and Rohatgi [GMR24a] conversely showed the hardness when the noise is generated from a block δ-Santha-Vazirani (SV) source (a weakly dependent source where each bit may have a weak dependency on the preceding noise bits).

Definition 1.3 (δ-Santha-Vazirani source)**.** A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called a δ-Santha-Vazirani (SV) source if:

$$\Pr_{\mathbf{x}\sim\mathcal{D}}[x_i = 1 \mid x_1 = x'_1, \ldots, x_{i-1} = x'_{i-1}] \in \left[\tfrac{1}{2} - \delta, \tfrac{1}{2} + \delta\right], \quad \forall i \in [k], \forall x'_1, \ldots, x'_{i-1} \in \mathbb{F}_2.$$

Their result was proven via an inductive reduction from the standard LPN of noise rate roughly $\eta = \frac{1}{2} - \Omega(2^k\delta)$, demonstrating that such weak sources still preserve the essential difficulty of $\mathsf{LPN}_{n,\eta}$. They also derive hardness-based implications for learning tasks, showing a separation between reinforcement learning and supervised learning. Another result [BLMZ19] also shows a certain hardness result for Batch LPN with $k = 2$ and a certain distribution.

A natural question that arises from this line of work is:

Where exactly is the boundary between hard and easy noise distributions in Batch LPN?

In this work, we continue the study of Batch LPN with dependent noise. We provide a more comprehensive characterization of noise distributions $\mathcal{D}$ through the lens of Boolean Fourier analysis, analyzing them from both hardness (reduction) and algorithmic perspectives.

1.1 Main Results

Our first result establishes a general reduction that captures a broad family of dependent-noise distributions. Roughly speaking, we show that any Batch-LPN instance whose noise distribution satisfies a mild Fourier-analytic condition is no easier than standard LPN in the low-noise regime. This is formalized in the Theorem below.

Theorem 1.4 (Theorem 3.1, informal)**.** *If the noise distribution $\mathcal{D}$ satisfies $\sum_{s\neq 0} |\hat{P}_{\mathcal{D}}(s)| \leq 2\varepsilon$, then solving $\mathsf{LPN}_{n,k,\mathcal{D}}$ is as hard as solving standard $\mathsf{LPN}_{n,1/2-\varepsilon}$.*

Following this reduction, we identify natural structural classes of noise distributions for which the condition is satisfied. We remark the any noise distribution $\mathcal{D}$ that is δ-biased (Definition 2.2) for $\delta = \frac{2\varepsilon}{2^k-1}$ satisfies the condition in Theorem 1.4.

We further compare our bound with the Santha-Vazirani sources framework of [GMR24a], and we observe that our condition is strictly more permissive.

Remark 1.5. Any δ-SV source is a 2δ-biased source (condition on the value of x_1, prove by induction); however, a δ-biased source need not be an $\Omega(\delta)$-SV source, e.g.:

- With probability ε, $\mathbf{x}$ is sampled uniformly from $\{00\ldots0, 11\ldots1\}$;
- With probability $1-\varepsilon$, $\mathbf{x}$ is sampled uniformly from $\mathbb{F}_2^k$.

Next, we present our reductions from the low-noise regime of the standard LPN. Recall that classical LPN exhibits a sharp algorithmic threshold: it is efficiently solvable at noise rate $\eta = \Theta(\frac{\log n}{n})$ but conjectured to be hard at $\eta = \Theta(\frac{\log^2 n}{n})$. Within this broader landscape, we consider the case of low-noise but ε-dense noise distributions, where every point in $\mathbb{F}_2^k$ has probability at least ε. We show that the corresponding Batch LPN instance retains the same hardness characteristics observed in the classical independent-noise setting. Our next result formalizes the precise range of ε for which the hardness reduction continues to hold.

Theorem 1.6 (Theorem 4.1 and Remark 4.2, informal)**.** *If the noise distribution $\mathcal{D}$ is ε-dense (meaning $P_{\mathcal{D}}(\mathbf{z}) \geq \varepsilon$ for all $\mathbf{z} \in \mathbb{F}_2^k$), then solving $\mathsf{LPN}_{n,k,\mathcal{D}}$ is as hard as solving standard $\mathsf{LPN}_{n,\eta}$ where the noise rate is roughly $\eta \approx \Omega(2^k\varepsilon/k)$.*

On the algorithmic side, we show that if even a single point in the support of the noise has a small probability, then the Batch-LPN instance becomes efficiently solvable. Thus, as in classical LPN, the boundary between hardness and computability is sharply determined by the effective noise mass.

Theorem 1.7 (Theorem 4.5, informal)**.** *Let $\mathcal{D}$ be a noise distribution such that $P_{\mathcal{D}}(\mathbf{z}) \leq \varepsilon$ for some $\mathbf{z} \in \mathbb{F}_2^k$. Then there is an algorithm for $\mathsf{LPN}_{n,k,\mathcal{D}}$ that makes $\exp\big(O\big(\varepsilon n^k 2^{2k}\big)\big)$ oracle queries and runs in time $\exp\big(O\big(\varepsilon n^k 2^{2k}\big)\big)$.*

Note that if we apply this algorithm to the standard $\mathsf{LPN}_{n,\eta}$, which can be seen as $\mathsf{LPN}_{n,k,\mathrm{Ber}(\eta)^k}$, we have $\varepsilon = \eta^k$ (by considering the probability of the all-ones error vector). Thus, if $\eta = O(1/n)$, the above algorithm runs in $n^{O_k(1)}$ time. This is tight up to a logarithmic factor, in the sense that the best known algorithm for low-noise LPN requires $\eta = O(\log n/n)$ to be efficient.

Finally, we also explore the inductive construction in [GMR24b], and provide a tighter analysis via Fourier analysis.

Theorem 1.8 (Theorem 5.1, informal)**.** *If the noise distribution $\mathcal{D}$ is a δ-SV source with bias $\delta \le O(2^{-k/2}\varepsilon)$, then solving $\mathsf{LPN}_{n,k,\mathcal{D}}$ is as hard as solving standard $\mathsf{LPN}_{n,1/2-\varepsilon}$.*

While [GMR24b] requires the condition $\delta \le O(2^{-k}\varepsilon)$, we give a quadratic improvement on the dependency on k.

2 Preliminaries

In this section, we fix notation and recall standard Fourier-analytic facts used throughout the paper. In this work, we use lowercase bold letters such as $\mathbf{u}$, $\mathbf{b}$, and $\mathbf{y}$ to denote vectors. In particular, $\mathbf{sk}$ represents the hidden vector, which is called the secret. Uppercase bold letters, such as $\mathbf{A}$ and $\mathbf{U}$, are used to denote matrices. The letters n, m, and k denote natural numbers.

We use calligraphic letters (e.g., $\mathcal{D}, \mathcal{R}$) and Greek letters (e.g., μ) to denote probability distributions. We use $P_{\mathcal{D}}(\mathbf{x}) = \Pr_{\mathbf{y}\sim\mathcal{D}}[\mathbf{y} = \mathbf{x}]$ to denote the probability mass function of some distribution $\mathcal{D}$. A discrete distribution $\mathcal{D}$ over a finite set X is called *ε-dense* if $\min_{x\in X} P_{\mathcal{D}}(x) \ge \varepsilon$.

2.1 Fourier Analysis on the Boolean Cube

We denote $\chi_{\mathbf{s}}(\mathbf{x}) = (-1)^{\langle \mathbf{s},\mathbf{x}\rangle}$ as the character function. Let $\mathcal{D}$ be a distribution over $\mathbb{F}_2^k$, the Fourier coefficient of $\mathcal{D}$ at $\mathbf{s} \in \mathbb{F}_2^k$ is defined as

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}}[\chi_{\mathbf{s}}(\mathbf{x})] = \sum_{\mathbf{x}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{x})(-1)^{\langle \mathbf{s},\mathbf{x}\rangle},$$

We have basic facts that for all $\mathbf{s} \in \mathbb{F}_2^k$, $|\widehat{P}_{\mathcal{D}}(\mathbf{s})| \le 1$ and $\widehat{P}_{\mathcal{D}}(\mathbf{0}) = 1$.

A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called ε-biased if $|\widehat{P}_{\mathcal{D}}(\mathbf{s})| \le \varepsilon$ for all $\mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$.

Definition 2.1 (δ-L_1 Fourier-bounded distribution)**.** A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called δ-L_1 Fourier-bounded if

$$\sum_{\mathbf{s}\in\mathbb{F}_2^k\setminus\{\mathbf{0}\}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| \le \delta.$$

Definition 2.2 (δ-biased distribution)**.** A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called δ-biased if

$$\forall \mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| \le \delta.$$

2.2 Affine Reduction

We use $\mathrm{GL}(k, 2)$ to denote the general linear group over $\mathbb{F}_2$, i.e., the set of all invertible $k \times k$ matrices with entries in $\mathbb{F}_2$. We will apply random affine maps of the form $x \mapsto \mathbf{A}x + b$ with $\mathbf{A} \in \mathrm{GL}(k, 2)$ and $b \in \mathbb{F}_2^k$. The Fourier transform behaves cleanly under such transformations.

Definition 2.3 ($\mathcal{R}$-Affine Reduction)**.** Let $\mathcal{R}$ be a distribution over pairs $(\mathbf{A}, \mathbf{b}) \in \mathrm{GL}(k, 2) \times \mathbb{F}_2^k$. An $\mathcal{R}$-affine reduction from a standard LPN distribution $\mathsf{LPN}_{n,\eta}$ is the following procedure:

1. Sample $(\mathbf{A}, \mathbf{b}) \sim \mathcal{R}$.

2. Draw k independent samples $(\mathbf{u}_i, y_i) \sim \mathsf{LPN}_{n,\eta}$ for $i = 1, \ldots, k$, and assemble them into matrices:

$$\mathbf{U} \coloneqq \begin{pmatrix} \mathbf{u}_1^\top \\ \mathbf{u}_2^\top \\ \vdots \\ \mathbf{u}_k^\top \end{pmatrix} \in \mathbb{F}_2^{k \times n}, \quad \mathbf{y} \coloneqq \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{pmatrix} \in \mathbb{F}_2^k.$$

3. Let $\mathbf{U}' = \mathbf{A}\mathbf{U}$ and $\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{b}$. Output the batch $(\mathbf{u}'_i, y'_i)_{i=1}^k$.

Remark 2.4 (Expression of new noise vector and sample independence).

1. The transformed batch corresponds to the new noise vector $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$.
2. Since $\mathbf{A}$ is always invertible and $\mathbf{U}$ is sampled uniformly at random, the rows of $\mathbf{U}'$ remain mutually independent and uniformly distributed over $\mathbb{F}_2^n$.

   Moreover, $\mathbf{U}'$ is independent of $\mathbf{e}'$. This is because even conditioned on fixed realizations of $\mathbf{A}, \mathbf{b}$, and $\mathbf{e}$ (which fully determine $\mathbf{e}'$), the matrix $\mathbf{U}' = \mathbf{A}\mathbf{U}$ remains uniformly distributed over $\mathbb{F}_2^{k \times n}$ due to the bijectivity of the map $\mathbf{U} \mapsto \mathbf{A}\mathbf{U}$.

Lemma 2.5 (Fourier Coefficients under Affine Reduction)**.** *Applying an $\mathcal{R}$-affine reduction to $\mathsf{LPN}_{n,\frac{1}{2}-\varepsilon}$ results in $\mathsf{LPN}_{n,k,\mathcal{D}}$, where the Fourier coefficient of the noise distribution $\mathcal{D}$ is given by*

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b})\sim\mathcal{R}} \left[(2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})} \cdot (-1)^{\langle \mathbf{s}, \mathbf{b} \rangle} \right], \quad \forall \mathbf{s} \in \mathbb{F}_2^k.$$

Proof. Since $\mathbf{A}$ is invertible, the transformed vectors $\mathbf{u}'_i$ are still uniformly distributed over $\mathbb{F}_2^n$.

Thus, it suffices to analyze the distribution of the new noise vector $\mathbf{e}' = (e'_1, \ldots, e'_k)$. Let $\mathbf{e} = (e_1, \ldots, e_k)$ be the original noise vector with independent coordinates $e_i \sim \mathrm{Ber}(\frac{1}{2} - \varepsilon)$. After applying the affine transformation $(\mathbf{A}, \mathbf{b}) \sim \mathcal{R}$, the new noise vector is $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$.

For any $\mathbf{s} \in \mathbb{F}_2^k$, the Fourier coefficient is:

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \mathop{\mathbb{E}}_{\mathbf{e}'\sim\mathcal{D}}[(-1)^{\langle \mathbf{s}, \mathbf{e}' \rangle}].$$

Conditioning on the choice of $(\mathbf{A}, \mathbf{b})$, we have:

$$\begin{aligned} \widehat{P}_{\mathcal{D}}(\mathbf{s}) &= \mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b})\sim\mathcal{R}} \mathop{\mathbb{E}}_{\mathbf{e}}[(-1)^{\langle \mathbf{s}, \mathbf{A}\mathbf{e}+\mathbf{b} \rangle}] \\ &= \mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b})\sim\mathcal{R}} \mathop{\mathbb{E}}_{\mathbf{e}}[(-1)^{\langle \mathbf{s}, \mathbf{b} \rangle} \cdot (-1)^{\langle \mathbf{A}^\top \mathbf{s}, \mathbf{e} \rangle}]. \end{aligned}$$

Taking expectation over $\mathbf{e}$, and using the independence of coordinates:

$$\mathop{\mathbb{E}}_{\mathbf{e}}[(-1)^{\langle \mathbf{A}^\top \mathbf{s}, \mathbf{e} \rangle}] = \prod_{i=1}^k \mathop{\mathbb{E}}_{e_i}[(-1)^{(\mathbf{A}^\top \mathbf{s})_i \cdot e_i}].$$

For each coordinate i, if $(\mathbf{A}^\top \mathbf{s})_i = 0$, then $(-1)^{0 \cdot e_i} = 1$. If $(\mathbf{A}^\top \mathbf{s})_i = 1$, then:

$$\mathop{\mathbb{E}}_{e_i}[(-1)^{e_i}] = (\tfrac{1}{2} - \varepsilon) \cdot (-1) + (\tfrac{1}{2} + \varepsilon) \cdot 1 = 2\varepsilon.$$

Thus, for $\mathbf{t} = \mathbf{A}^\top \mathbf{s}$, we have:

$$\mathop{\mathbb{E}}_{\mathbf{e}}[(-1)^{\langle \mathbf{t}, \mathbf{e} \rangle}] = \prod_{i=1}^k \begin{cases} 1 & \text{if } \mathbf{t}_i = 0 \\ 2\varepsilon & \text{if } \mathbf{t}_i = 1 \end{cases} = (2\varepsilon)^{\mathrm{wt}(\mathbf{t})} = (2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}.$$

Putting everything together:

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b})\sim\mathcal{R}} \left[(-1)^{\langle \mathbf{s}, \mathbf{b} \rangle} \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})} \right],$$

which completes the proof. □

We also use the following straightforward lemmas and omit their proofs.

Lemma 2.6. *For two i.i.d. random variables $X, Y \in \mathbb{F}_2$ sampled from $\mathrm{Ber}(\frac{1}{2}-\delta_1), \mathrm{Ber}(\frac{1}{2}-\delta_2)$, the distribution of $Z = X + Y$ is $\mathrm{Ber}\left(\frac{1}{2} - 2\delta_1\delta_2\right)$.*

3 Reduction from High Noise LPN to δ-Biased Distributions

Theorem 3.1. *If there is an algorithm A_1 that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ in time T with success probability at least p, then there is an algorithm A_2 that solves $\mathsf{LPN}_{n,\frac{1}{2}-\varepsilon}$ in time $T + \mathrm{poly}(2^k, n)$ with success probability at least p, provided that the noise rate $\frac{1}{2} - \varepsilon$ satisfies*

$$\sum_{\mathbf{s}\in\mathbb{F}_2^k\setminus\{\mathbf{0}\}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| \le 2\varepsilon. \tag{1}$$

Corollary 3.2. *Any noise distribution $\mathcal{D}$ that is δ biased for $\delta = \frac{2\varepsilon}{2^k-1}$ satisfies the condition in Theorem 3.1.*

Before proving Theorem 3.1, we need two lemmas.

Lemma 3.3. *For any nonzero $\mathbf{s} \in \mathbb{F}_2^k$, and any sign (either $+$ or $-$), there exists a probability distribution $\mathcal{D}$ on $\mathbb{F}_2^k$ such that,*

$$\widehat{P}_{\mathcal{D}}(\mathbf{t}) = \begin{cases} 0 & (\mathbf{t} \neq \mathbf{s}, \mathbf{t} \neq \mathbf{0}), \\ \pm 1 & (\mathbf{t} = \mathbf{s}), \\ 1 & (\mathbf{t} = \mathbf{0}). \end{cases}$$

Proof. When $\mathcal{D}$ is defined as:

$$P_{\mathcal{D}}(\mathbf{t}) := (1 \pm \chi_{\mathbf{s}}(\mathbf{t}))\, 2^{-k}.$$

One can verify that $\mathcal{D}$ is a valid distribution and has the desired Fourier coefficients. □

Lemma 3.4. *For any $\mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$ there exists $\mathbf{A} \in \mathrm{GL}(k, 2)$ such that $\mathbf{A}^\top \mathbf{s} = (1, 0, \ldots, 0)^\top$.*

Proof. Extend $\mathbf{s}$ to a basis of $\mathbb{F}_2^k$: $\{\mathbf{s}, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$. Define the matrix $\mathbf{M}$ by

$$\mathbf{M} := \begin{pmatrix} \mathbf{s} & \mathbf{v}_2 & \cdots & \mathbf{v}_k \end{pmatrix}$$

Since $\{\mathbf{s}, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ is a basis, the matrix $\mathbf{M}$ is invertible. Thus,

$$\mathbf{M}^{-1}\mathbf{M} = \mathbf{I},$$

where $\mathbf{I}$ denotes the identity matrix. Looking at the first column of this equation, we obtain

$$\mathbf{M}^{-1}\mathbf{s} = (1, 0, \ldots, 0)^\top.$$

Setting $\mathbf{A}^\top := \mathbf{M}^{-1}$ proves the claim. □

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Consider the following $\mathcal{R}$-affine reduction applied to $\mathsf{LPN}_{n,\frac{1}{2}-\varepsilon}$, where $(\mathbf{A}, \mathbf{b}) \sim \mathcal{R}$ is defined by the following procedure:

We sample $(\mathbf{A}, \mathbf{b})$ from 2^k branches, where each branch is indexed by a vector $\mathbf{s} \in \mathbb{F}_2^k$. For the branch with index $\mathbf{s} \neq \mathbf{0}$:

- This branch is chosen with probability $\frac{|\widehat{P}_{\mathcal{D}}(\mathbf{s})|}{2\varepsilon}$ (by the assumption (1), this is a valid probability).
- Assign a matrix $\mathbf{A}_{\mathbf{s}} \in \mathrm{GL}(k, 2)$ such that $\mathbf{A}_{\mathbf{s}}^\top \mathbf{s} = (1, 0, \ldots, 0)^\top$; such $\mathbf{A}_{\mathbf{s}}$ exists by Lemma 3.4.

- Sample $\mathbf{b}$ from a distribution $\mu_{\mathbf{s}}$ such that

$$\widehat{P}_{\mu_{\mathbf{s}}}(\mathbf{t}) = \begin{cases} 0 & (\mathbf{t} \neq \mathbf{s}, \mathbf{t} \neq \mathbf{0}), \\ \operatorname{sign}(\widehat{P}_{\mathcal{D}}(\mathbf{s})) & (\mathbf{t} = \mathbf{s}), \\ 1 & (\mathbf{t} = \mathbf{0}). \end{cases}$$

  Such $\mu_{\mathbf{s}}$ exists by Lemma 3.3.

- Return $(\mathbf{A}_{\mathbf{s}}, \mathbf{b})$.

Otherwise, we enter the branch with index $\mathbf{0}$ with probability $1 - \sum_{\mathbf{s} \neq \mathbf{0}} \frac{|\widehat{P}_{\mathcal{D}}(\mathbf{s})|}{2\varepsilon}$ (this is a valid probability by the assumption (1)). In this branch, we set $(\mathbf{A}, \mathbf{b}) = (\mathbf{I}, \mathrm{Unif}(\mathbb{F}_2^k))$.

The intuition of our construction is that each branch contributes to exactly one (non-zero) coordinate of the Fourier coefficient.

By Remark 2.4, the new noise vector $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$ and its distribution are independent of the samples. So it remains to verify that the resulting noise distribution $\mathcal{D}_{\text{out}}$ satisfies $\mathcal{D}_{\text{out}} = \mathcal{D}$. We can compute the Fourier coefficient of $\mathbf{e}'$ using Lemma 2.5, for every $\mathbf{x} \in \mathbb{F}_2^k$,

$$\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x}) = \mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b}) \sim \mathcal{R}} \left[(-1)^{\langle \mathbf{x}, \mathbf{b} \rangle} \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{x})} \right].$$

We consider the contribution of each branch $\mathbf{s} \in \mathbb{F}_2^k$ to the final coefficient. Let $\mathcal{R}|\mathbf{s}$ denote the distribution of $(\mathbf{A}, \mathbf{b})$ conditioned on the algorithm entering branch $\mathbf{s}$.

For the trivial case $\mathbf{x} = \mathbf{0}$, $\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{0}) = 1$ holds automatically as probabilities sum to 1. We now focus on $\mathbf{x} \neq \mathbf{0}$.

Case A: Branch $\mathbf{s} = \mathbf{0}$. Under this branch, $\mathbf{b} \sim \mathrm{Unif}(\mathbb{F}_2^k)$. For any $\mathbf{x} \neq \mathbf{0}$, we have:

$$\mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b}) \sim \mathcal{R}|\mathbf{s}} \left[(-1)^{\langle \mathbf{x}, \mathbf{b} \rangle} \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{x})} \right] = \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathrm{Unif}(\mathbb{F}_2^k)} \left[(-1)^{\langle \mathbf{x}, \mathbf{b} \rangle} \right] \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{x})} = 0.$$

Thus, this branch contributes 0 to the coefficient $\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x})$.

Case B: Branch $\mathbf{s} \neq \mathbf{0}$. For any $\mathbf{x} \neq \mathbf{0}$, we have:

$$\mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b}) \sim \mathcal{R}|\mathbf{s}} \left[(-1)^{\langle \mathbf{x}, \mathbf{b} \rangle} \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{x})} \right] = \mathop{\mathbb{E}}_{\mathbf{b} \sim \mu_{\mathbf{s}}} \left[(-1)^{\langle \mathbf{x}, \mathbf{b} \rangle} \right] \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{A}_{\mathbf{s}}^\top \mathbf{x})}.$$

Recall that $\widehat{P}_{\mu_{\mathbf{s}}}(\mathbf{x}) = 0$ if $\mathbf{x} \notin \{\mathbf{0}, \mathbf{s}\}$. Since we assume $\mathbf{x} \neq \mathbf{0}$, this term is non-zero only if $\mathbf{x} = \mathbf{s}$. Specifically:

$$\mathop{\mathbb{E}}_{(\mathbf{A},\mathbf{b}) \sim \mathcal{R}|\mathbf{s}} \left[(-1)^{\langle \mathbf{x}, \mathbf{b} \rangle} \cdot (2\varepsilon)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{x})} \right] = \begin{cases} 0 & (\mathbf{x} \neq \mathbf{s}) \\ \operatorname{sign}(\widehat{P}_{\mathcal{D}}(\mathbf{s})) \cdot (2\varepsilon)^1 & (\mathbf{x} = \mathbf{s}) \end{cases}$$

Weighting this by the probability of choosing branch $\mathbf{s}$, which is $\frac{|\widehat{P}_{\mathcal{D}}(\mathbf{s})|}{2\varepsilon}$, this branch contributes: $\begin{cases} 0 & (\mathbf{x} \neq \mathbf{s}) \\ \widehat{P}_{\mathcal{D}}(\mathbf{s}) & (\mathbf{x} = \mathbf{s}) \end{cases}$ to the coefficient $\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x})$.

Summing over all branches for a fixed $\mathbf{x} \neq \mathbf{0}$:

$$\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x}) = \underbrace{0}_{\text{Branch } \mathbf{0}} + \sum_{\mathbf{s} \neq \mathbf{0}} (\text{Contribution of Branch } \mathbf{s}).$$

In the sum, only the term where $\mathbf{s} = \mathbf{x}$ is non-zero. Therefore, $\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x}) = \widehat{P}_{\mathcal{D}}(\mathbf{x})$. Since $\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x}) = \widehat{P}_{\mathcal{D}}(\mathbf{x})$ for all $\mathbf{x}$, we conclude $\mathcal{D}_{\text{out}} = \mathcal{D}$. □

4 Reduction from Low-Noise LPN to ε-Dense Distributions

Theorem 4.1. *Suppose $\mathcal{D}$ is ε-dense over $\mathbb{F}_2^k$. If there is an algorithm A_1 that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ in time T with success probability at least p, then there is an algorithm A_2 that solves $\mathsf{LPN}_{n,\eta}$ in time $T + \mathrm{poly}(2^k, n)$ with success probability at least p, provided that the noise rate η satisfies*

$$1 - (1-\eta)^k \le (2^k - 1)\varepsilon. \tag{2}$$

Remark 4.2. Equivalently, in the low-noise regime $\eta \ll 1/k$, the condition becomes approximately $\eta = O\left(\frac{2^k}{k} \cdot \min_{\mathbf{z}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{z})\right)$. Note that $\min_{\mathbf{z}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{z})$ is always smaller than 2^{-k}.

We will need two lemmas.

Lemma 4.3 (Averaging over a random invertible $\mathbf{A}$)**.** *Let $\alpha \in [-1, 1]$ and let $\mathbf{A}$ be uniform over $\mathrm{GL}(k, 2)$. For any fixed $\mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$,*

$$\mathop{\mathbb{E}}_{\mathbf{A}}\left[\alpha^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}\right] = \frac{1}{2^k - 1} \sum_{\mathbf{t}\in\mathbb{F}_2^k\setminus\{\mathbf{0}\}} \alpha^{\mathrm{wt}(\mathbf{t})} = \frac{(1+\alpha)^k - 1}{2^k - 1}.$$

Proof. The map $\mathbf{A} \mapsto \mathbf{A}^\top \mathbf{s}$ has a uniform image over $\mathbb{F}_2^k \setminus \{\mathbf{0}\}$ for fixed nonzero $\mathbf{s}$. Hence $\mathbb{E}_{\mathbf{A}}[\alpha^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}] = \frac{1}{2^k-1}\sum_{\mathbf{t}\neq\mathbf{0}} \alpha^{\mathrm{wt}(\mathbf{t})} = \frac{1}{2^k-1}\sum_{w=1}^{k} \binom{k}{w}\alpha^w = \frac{(1+\alpha)^k-1}{2^k-1}$. □

Lemma 4.4 (Fourier scaling)**.** *Let $\mathcal{D}$ be a distribution on $\mathbb{F}_2^k$ and let*

$$P_{\mathcal{D}'}(\mathbf{z}) := a\left(P_{\mathcal{D}}(\mathbf{z}) - 2^{-k}\right) + 2^{-k} \qquad \text{for some } a \in \mathbb{R}.$$

Then $\widehat{P}_{\mathcal{D}'}(\mathbf{0}) = 1$ and for every $\mathbf{s} \neq \mathbf{0}$, $\widehat{P}_{\mathcal{D}'}(\mathbf{s}) = a \cdot \widehat{P}_{\mathcal{D}}(\mathbf{s})$. Moreover, $\mathcal{D}'$ is a valid distribution (i.e., $P_{\mathcal{D}'} \ge 0$ and sums to 1) whenever $a \le \frac{1}{1-2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})}$.

Proof. By linearity of the Fourier transform and the fact that the uniform distribution has Fourier coefficients 1 at $\mathbf{s} = \mathbf{0}$ and 0 elsewhere, the stated equations follow. For validity, note $\min_{\mathbf{z}} P_{\mathcal{D}'}(\mathbf{z}) = a\left(\min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z}) - 2^{-k}\right) + 2^{-k} \ge 0$ whenever $a \le \frac{1}{1-2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})}$. □

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Let $\eta \in (0, \frac{1}{2})$ and write $\alpha := 1 - 2\eta \in (0, 1]$. Consider the following $\mathcal{R}$-affine reduction (Definition 2.3) applied to $\mathsf{LPN}_{n,\eta}$, where $(\mathbf{A}, \mathbf{b}) \sim \mathcal{R}$ is defined as follows:

1. Sample $\mathbf{A} \leftarrow \mathrm{Unif}(\mathrm{GL}(k, 2))$.
2. Sample $\mathbf{b} \sim \mathcal{D}'$ independently of $\mathbf{A}$, from a distribution $\mathcal{D}'$ to be fixed below.

By Remark 2.4, the new noise vector $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$ and its distribution are independent of the samples. So it remains to verify that the resulting noise distribution $\mathcal{D}_{\mathrm{out}}$ satisfies $\mathcal{D}_{\mathrm{out}} = \mathcal{D}$. By Lemma 2.5 for the resulting noise distribution $\mathcal{D}_{\mathrm{out}}$ we have, for every $\mathbf{s} \in \mathbb{F}_2^k$,

$$\widehat{P}_{\mathcal{D}_{\mathrm{out}}}(\mathbf{s}) = \mathop{\mathbb{E}}_{\mathbf{A},\mathbf{b}}\left[\alpha^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}(-1)^{\langle \mathbf{s},\mathbf{b}\rangle}\right] = \left(\mathop{\mathbb{E}}_{\mathbf{A}}[\alpha^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}]\right) \cdot \left(\mathop{\mathbb{E}}_{\mathbf{b}\sim\mathcal{D}'}[(-1)^{\langle \mathbf{s},\mathbf{b}\rangle}]\right).$$

For $\mathbf{s} = \mathbf{0}$, both factors equal 1, hence $\widehat{P}_{\mathcal{D}_{\mathrm{out}}}(\mathbf{0}) = 1$. For $\mathbf{s} \neq \mathbf{0}$, by Lemma 4.3 we get

$$\mathop{\mathbb{E}}_{\mathbf{A}}\left[\alpha^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}\right] = \frac{(1+\alpha)^k - 1}{2^k - 1} = \frac{(2-2\eta)^k - 1}{2^k - 1} = \frac{2^k(1-\eta)^k - 1}{2^k - 1}.$$

Define the scaling constant

$$a := \frac{2^k - 1}{2^k(1-\eta)^k - 1}. \tag{3}$$

With this choice and by Lemma 4.4, we set $\mathcal{D}'$ so that

$$P_{\mathcal{D}'}(\mathbf{z}) \coloneqq a\left(P_{\mathcal{D}}(\mathbf{z}) - 2^{-k}\right) + 2^{-k}, \tag{4}$$

then for every $\mathbf{s} \neq \mathbf{0}$ we have

$$\widehat{P}_{\mathcal{D}'}(\mathbf{s}) = a\widehat{P}_{\mathcal{D}}(\mathbf{s}) \qquad \text{and hence} \qquad \widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{s}) = \frac{2^k(1-\eta)^k - 1}{2^k - 1} \cdot \left(a\widehat{P}_{\mathcal{D}}(\mathbf{s})\right) = \widehat{P}_{\mathcal{D}}(\mathbf{s}).$$

Together with the case $\mathbf{s} = \mathbf{0}$ this shows that $\mathcal{D}_{\text{out}} = \mathcal{D}$. Therefore, after the reduction the batch distribution is $\mathsf{LPN}_{n,k,\mathcal{D}}$ *exactly*. Running A_1 on the produced batch recovers the secret with the same success probability $\geq p$.

Validity of $\mathcal{D}'$. It remains to verify that $\mathcal{D}'$ defined in (4) is a valid distribution. By Lemma 4.4, this holds whenever $a \leq \frac{1}{1-2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})}$. Using the definition (3) of a, this condition is equivalent to

$$\frac{2^k - 1}{2^k(1-\eta)^k - 1} \leq \frac{1}{1 - 2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})} \quad \Longleftrightarrow \quad (2^k - 1)\left(1 - 2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})\right) \leq 2^k(1-\eta)^k - 1.$$

Rearranging gives $\min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z}) \geq \dfrac{1-(1-\eta)^k}{2^k - 1}$, which is the the stated hypothesis (2). Thus $\mathcal{D}'$ is valid. □

Given the hardness results for the low-noise LPN regime, we now point out that the same algorithm–hardness gap also appears in the more general setting of LPN with dependent noise. Even in the classical independent-noise case, when the noise bits are i.i.d. Bernoulli with rate η, it is known that for very low noise $\eta \approx \log n/n$ there is a simple polynomial-time decoding algorithm: by sampling a polynomial number of examples one obtains a system containing many completely noiseless equations and can then recover the secret via Gaussian elimination. At the same time, it is conjectured that LPN remains computationally hard even when the noise rate is as small as $\eta \approx \log^2 n/n$. Our dependent-noise model exhibits the same qualitative behavior, and on the algorithmic side, we analyze a sample-and-eliminate decoder combined with the linearization method in [AG11], which yields an efficient algorithm in the low-noise regime below.

Theorem 4.5. *Let $\mathcal{D}$ be a noise distribution such that $P_{\mathcal{D}}(\mathbf{z}) \leq \varepsilon$ for some $\mathbf{z} \in \mathbb{F}_2^k$. Then there is an algorithm for $\mathsf{LPN}_{n,k,\mathcal{D}}$ that makes $\exp\big(O(\varepsilon n^k 2^{2k})\big)$ oracle queries and runs in time $\exp\big(O(\varepsilon n^k 2^{2k})\big)$.*

Note that if we fix k to be a constant and take $\varepsilon = n^{-k}$, then $\varepsilon n^k 2^{2k}$ is a constant, so the above bounds yield a polynomial-time algorithm that uses only a polynomial number of samples, even in the presence of dependent noise. The proof of this theorem is straightforward and is deferred to the appendix.

5 Reduction to δ-SV source Batch LPN

In this section, we present an improved analysis of the reduction from [GMR24b]. We define the LPN noise parameter as $\eta = \frac{1}{2} - \varepsilon$, where ε is the bias. Our result can be summarized as follows.

Theorem 5.1. *Suppose $\mathcal{D}$ is a δ-SV source (Definition 1.3), where $\delta = 2^{-\frac{k-1}{2}}\varepsilon$. If there is an algorithm A_1 that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ in time T with success probability at least p, then there is an algorithm A_2 that solves $\mathsf{LPN}_{n,\eta}$ in time $T + \mathrm{poly}(2^k, n)$ with success probability at least p.*

Our result improves the dependence of δ on k from $\Theta(2^{-k}\varepsilon)$ in [GMR24b] to $\Theta(2^{-\frac{k}{2}}\varepsilon)$. For completeness, we postpone the proof of Theorem 5.1, first give an overview of their reduction, and then present our improved result.

Overview of the reduction in [GMR24b]. The reduction simulates samples $(\mathbf{u}_i, \mathbf{y}_i = \langle \mathbf{u}_i, \mathbf{sk}\rangle + e_i)_{i=1}^k$ from the batch instance $\mathsf{LPN}_{n,k,\mathcal{D}}$ that shares the same secret key $\mathbf{sk}$, by making queries to a standard $\mathsf{LPN}_{n,\eta}$ oracle.

To construct the first sample $(\mathbf{u}_1, \mathbf{y}_1)$, we need to simulate the marginal distribution of e_1. Let $\Pr(e_1 = 1) = \frac{1}{2} - \delta_1$. The algorithm queries a fresh sample $(\mathbf{u}', \mathbf{y}')$ from the $\mathsf{LPN}_{n,\eta}$ oracle. It then sets $(\mathbf{u}_1, \mathbf{y}_1) = (\mathbf{u}', \mathbf{y}' + e_1')$, where e_1' is an additional noise bit sampled independently from $\mathrm{Ber}(\frac{1}{2} - \frac{\delta_1}{2\varepsilon})$. Such a random variable e_1' can be constructed because $\delta_1 \le \delta < \varepsilon$. By Lemma 2.6, this successfully simulates the first sample.

For the i-th sample $(\mathbf{u}_i, \mathbf{y}_i)$, the algorithm first queries a fresh sample $(\mathbf{u}', \mathbf{y}')$. It then tries to find a distribution μ on $\mathbb{F}_2^i$, and sample a vector $\mathbf{F} \sim \mu$. Such that for any $\mathbf{z} \in \mathbb{F}_2^{i-1}$, the linear combination satisfies:

$$\Pr_{\mathbf{F}\sim\mu}\left(F_0 + e_1 F_1 + \cdots + e_{i-1}F_{i-1} = 1 \mid (e_j)_{j=1}^{i-1} = \mathbf{z}\right) = \frac{1}{2} - \frac{\frac{1}{2} - \Pr_{\mathbf{e}\sim\mathcal{D}}(e_i = 1 | (e_j)_{j=1}^{i-1} = \mathbf{z})}{2\varepsilon}$$

where F_0 is an extra random bit from $\mathbf{F}$. Note that μ is a fixed distribution that only depend on $\mathcal{D}$, and independent from the actual values of $(e_j)_{j=1}^{i-1}$.

If such μ and $\mathbf{F} \sim \mu$ exists, then we set $(\mathbf{u}_i, \mathbf{y}_i) = (\mathbf{u}' + \sum_{j=1}^{i-1} F_j \mathbf{u}_j, \mathbf{y}' + \sum_{j=1}^{i-1} F_j \mathbf{y}_j)$. This successfully simulates the noise of the i-th sample by Lemma 2.6. And $\mathbf{u}_i$ is also independent from all previous samples $(\mathbf{u}_j)_{j=1}^{i-1}$ as $\mathbf{u}'$ is sampled independently from the uniform distribution on $\mathbb{F}_2^n$.

The dependency of δ on k comes from the existence of μ. In [GMR24b], the authors show that such μ exists if $\mathcal{D}$ is a δ-SV source with $\delta \le 2^{-k-3}\varepsilon$ using linear algebraic arguments. Our observation is that μ can be constructed using Fourier analysis, which leads to a simpler proof and improved parameters. Formally, we have the following lemma.

Lemma 5.2 (Improved Lemma 3.2 from [GMR24b]). *Fix $k \in \mathbb{N}$, and pick any function $f : \mathbb{F}_2^k \to [0,1]$ satisfying*

$$\sum_{\mathbf{z}\in\mathbb{F}_2^k} (1 - 2f(\mathbf{z}))^2 \le 1.$$

Then there exists a distribution μ on $\mathbb{F}_2^{k+1}$ such that for all $\mathbf{z} \in \mathbb{F}_2^k$,

$$\Pr_{\mathbf{F}\sim\mu}\left(F_0 + \sum_{i=1}^k \mathbf{z}_i F_i \equiv 1 \pmod 2\right) = f(\mathbf{z}).$$

Moreover, there is an algorithm that takes f as input and samples $\mathbf{F} \sim \mu$ in time $2^{O(k)}$.

Proof. Let $g(\mathbf{z}) \coloneqq 1 - 2f(\mathbf{z})$. The condition of the lemma implies $\sum_{\mathbf{z}\in\mathbb{F}_2^k} g(\mathbf{z})^2 \le 1$.

The idea is to construct μ such that its marginals correspond to the Fourier coefficients of g. Recall that $g(\mathbf{z}) = \sum_{\mathbf{s}\in\mathbb{F}_2^k} \widehat{g}(\mathbf{s})(-1)^{\langle \mathbf{z},\mathbf{s}\rangle}$.

Let $\mathbf{F} = (F_0, F_1, \ldots, F_k) \sim \mu$. We have:

$$\begin{aligned}\Pr_{\mathbf{F}\sim\mu}\left(F_0 + \langle \mathbf{z}, \mathbf{F}_{1..k}\rangle \equiv 1 \pmod 2\right) &= \mathbb{E}_{\mathbf{F}\sim\mu}\left[\frac{1 - (-1)^{F_0 + \langle \mathbf{z},\mathbf{F}_{1..k}\rangle}}{2}\right] \\ &= \frac{1}{2}\left(1 - \mathbb{E}_{\mathbf{F}\sim\mu}\left[(-1)^{F_0}(-1)^{\langle \mathbf{z},\mathbf{F}_{1..k}\rangle}\right]\right).\end{aligned}$$

To satisfy the lemma statement, we need this probability to equal $f(\mathbf{z}) = \frac{1-g(\mathbf{z})}{2}$. This is equivalent to requiring:

$$\mathbb{E}_{\mathbf{F}\sim\mu}\left[(-1)^{F_0}(-1)^{\langle \mathbf{z},\mathbf{F}_{1..k}\rangle}\right] = g(\mathbf{z}).$$

Let us define the distribution μ supported on vectors $(b, \mathbf{s}) \in \mathbb{F}_2 \times \mathbb{F}_2^k$. The LHS can be rewritten as:

$$\sum_{\mathbf{s}\in\mathbb{F}_2^k} \left(\mu(0,\mathbf{s}) - \mu(1,\mathbf{s})\right)(-1)^{\langle \mathbf{z},\mathbf{s}\rangle}.$$

Comparing this with the Fourier expansion of $g(\mathbf{z})$, it suffices to construct μ such that for all $\mathbf{s} \in \mathbb{F}_2^k$:

$$\mu(0,\mathbf{s}) - \mu(1,\mathbf{s}) = \widehat{g}(\mathbf{s}).$$

We define the weights as follows:

$$(\mu(0,\mathbf{s}), \mu(1,\mathbf{s})) \coloneqq \begin{cases} (|\widehat{g}(\mathbf{s})|, 0) & \text{if } \widehat{g}(\mathbf{s}) > 0, \\ (0, |\widehat{g}(\mathbf{s})|) & \text{if } \widehat{g}(\mathbf{s}) \le 0. \end{cases}$$

This assignment ensures $\mu(0,\mathbf{s}) - \mu(1,\mathbf{s}) = \widehat{g}(\mathbf{s})$. It remains to show that these weights can form a valid probability distribution. The total mass required is

$$M \coloneqq \sum_{\mathbf{s}\in\mathbb{F}_2^k} (\mu(0,\mathbf{s}) + \mu(1,\mathbf{s})) = \sum_{\mathbf{s}\in\mathbb{F}_2^k} |\widehat{g}(\mathbf{s})| = \|\widehat{g}\|_1.$$

By Parseval's identity and the Cauchy–Schwarz inequality:

$$\|\widehat{g}\|_1^2 = \left(\sum_{\mathbf{s}\in\mathbb{F}_2^k} |\widehat{g}(\mathbf{s})| \cdot 1\right)^2 \le 2^k \sum_{\mathbf{s}\in\mathbb{F}_2^k} |\widehat{g}(\mathbf{s})|^2 = 2^k \cdot \frac{1}{2^k} \sum_{\mathbf{z}\in\mathbb{F}_2^k} g(\mathbf{z})^2 = \sum_{\mathbf{z}\in\mathbb{F}_2^k} g(\mathbf{z})^2 \le 1.$$

Thus, $M \le 1$. If $M < 1$, we can distribute the remaining probability mass $1 - M$ arbitrarily (e.g., equally adding to $\mu(0,\mathbf{0})$ and $\mu(1,\mathbf{0})$) without affecting the difference $\mu(0,\mathbf{s}) - \mu(1,\mathbf{s})$.

Since computing Fourier coefficients takes $2^{O(k)}$ time, we can sample from μ within the required time bounds. □

With this lemma, we are now ready to verify the main theorem of this section.

Proof of Theorem 5.1. By the previous discussion, we need to construct a distribution μ on $\mathbb{F}_2^i$ such that for any $\mathbf{z} \in \mathbb{F}_2^{i-1}$,

$$\Pr_{\mathbf{F}\sim\mu}\left(F_0 + \sum_{j=1}^{i-1} z_j F_j \equiv 1 \pmod 2\right) = \frac{1}{2} - \frac{\frac{1}{2} - \Pr_{\mathbf{e}\sim\mathcal{D}}(e_i = 1 | (e_j)_{j=1}^{i-1} = \mathbf{z})}{2\varepsilon}.$$

Defining $f : \mathbb{F}_2^{i-1} \to [0,1]$ as $f(\mathbf{z}) \coloneqq \frac{1}{2} - \frac{\frac{1}{2} - \Pr_{\mathbf{e}\sim\mathcal{D}}(e_i=1|(e_j)_{j=1}^{i-1}=\mathbf{z})}{2\varepsilon}$, we can verify that:

$$\sum_{\mathbf{z}\in\mathbb{F}_2^{i-1}} (1 - 2f(\mathbf{z}))^2 = \sum_{\mathbf{z}\in\mathbb{F}_2^{i-1}} \left(\frac{\frac{1}{2} - \Pr_{\mathbf{e}\sim\mathcal{D}}(e_i = 1 | (e_j)_{j=1}^{i-1} = \mathbf{z})}{\varepsilon}\right)^2 \le 2^{i-1} \cdot \frac{\delta^2}{\varepsilon^2},$$

where the second inequality comes from Definition 1.3. As $\delta \le 2^{-\frac{k-1}{2}}\varepsilon$ and $i \le k$, the above is bounded by 1. Thus, we can apply Lemma 5.2 to show the existence of μ. □

6 Technical Limitations and Future Directions

In our reduction framework, the only structural tool we used for manipulating the noise distribution is an affine transformation of the form $e' = Ae + b$, where (A, b) is sampled from a distribution over $\mathrm{GL}(k,2) \times \mathbb{F}_2^k$. While this approach is analytically clean and leads to explicit formulas for the Fourier coefficients of the transformed noise, it inevitably encounters a fundamental limitation: for a random full-rank matrix A and a random shift b, the Shannon entropy of the transformed noise vector $Ae + b$ is always at least the Shannon entropy of e itself. In particular, for any (A, b) sampled from a distribution over $\mathrm{GL}(k,2) \times \mathbb{F}_2^k$, we have $H(Ae + b) \ge H(e)$, with equality only when b is fixed. Thus, our current L_1 Fourier bound is already essentially optimal under affine reductions.

A second limitation of our current analysis is that it focuses on Batch LPN instances in which the noise is generated as a block distribution over $\mathbb{F}_2^k$. This setting naturally captures block SV sources and other

block-structured dependencies studied in prior work such as [GMR24a] and [AG11]. However, many forms of dependent noise arising in machine learning and cryptographic applications are not block sources, but instead involve long-range temporal correlations (e.g. Markov dependencies), higher-order dependencies across batches, or global constraints not expressible as a single distribution over $\mathbb{F}_2^k$. Our algorithm and reductions do not immediately extend to these more general dependent-noise models. Expanding the theory to these more general noise processes is a promising direction for future work.

References

[AAC+25] Omar Alrabiah, Prabhanjan Ananth, Miranda Christ, Yevgeniy Dodis, and Sam Gunn. Ideal Pseudorandom Codes. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1638–1647, Prague Czechia, June 2025. ACM.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *International Colloquium on Automata, Languages, and Programming*, pages 403–415. Springer, 2011.

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On Pseudorandom Generators with Linear Stretch in NC0. *computational complexity*, 17(1):38–69, April 2008.

[BBTV25] Kiril Bangachev, Guy Bresler, Stefan Tiegel, and Vinod Vaikuntanathan. Near-Optimal Time-Sparsity Trade-Offs for Solving Noisy Linear Equations. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1910–1920, Prague Czechia, June 2025. ACM.

[BD20] Zvika Brakerski and Nico Döttling. Hardness of LWE on General Entropic Distributions, 2020.

[BFKL94] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In Gerhard Goos, Juris Hartmanis, and Douglas R. Stinson, editors, *Advances in Cryptology — CRYPTO' 93*, volume 773, pages 278–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[BHK+21] Davide Bellizia, Clément Hoffmann, Dina Kamel, Hanlin Liu, Pierrick Méaux, François-Xavier Standaert, and Yu Yu. Learning parity with physical noise: Imperfections, reductions and fpga prototype. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 390–417, 2021.

[BHLM25] Arpon Basu, Jun-Ting Hsieh, Andrew D Lin, and Peter Manohar. Solving random planted csps below the $n^{k/2}$ threshold. *arXiv preprint arXiv:2507.10833*, 2025.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, July 2003.

[BLMZ19] James Bartusek, Tancrède Lepoint, Fermi Ma, and Mark Zhandry. New Techniques for Obfuscating Conjunctions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, volume 11478, pages 636–666. Springer International Publishing, Cham, 2019.

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-Case Hardness for LPN and Cryptographic Hashing via Code Smoothing. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, volume 11478, pages 619–635. Springer International Publishing, Cham, 2019.

[CSZ25] Xue Chen, Wenxuan Shu, and Zhaienhe Zhou. Algorithms for Sparse LPN and LSPN Against Low-noise (extended abstract). In *Proceedings of Thirty Eighth Conference on Learning Theory*, pages 1091–1093. PMLR, July 2025.

[DJ24] Quang Dao and Aayush Jain. Lossy Cryptography from Code-Based Assumptions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024*, volume 14922, pages 34–75. Springer Nature Switzerland, Cham, 2024.

[FKO06] Uriel Feige, Jeong Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 497–508, Berkeley, CA, USA, 2006. IEEE.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, August 1986.

[GMR24a] Noah Golowich, Ankur Moitra, and Dhruv Rohatgi. Exploration is harder than prediction: Cryptographically separating reinforcement learning from supervised learning. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1953–1967. IEEE, 2024.

[GMR24b] Noah Golowich, Ankur Moitra, and Dhruv Rohatgi. On learning parities with dependent noise. *arXiv preprint arXiv:2404.11325*, 2024.

[LF06] Éric Levieil and Pierre-Alain Fouque. An improved lpn algorithm. In *International conference on security and cryptography for networks*, pages 348–359. Springer, 2006.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 378–389. Springer, 2005.

[Pie12] Krzysztof Pietrzak. Cryptography from Learning Parity with Noise. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science*, volume 7147, pages 99–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Val15] Gregory Valiant. Finding Correlations in Subquadratic Time, with Applications to Learning Parities and the Closest Pair Problem. *Journal of the ACM*, 62(2):1–45, May 2015.

[YYLG16] Nan Yao, Yu Yu, Xiangxue Li, and Dawu Gu. On the robustness of learning parity with noise. In Kwok-Yan Lam, Chi-Hung Chi, and Sihan Qing, editors, *Information and Communications Security*, pages 99–106, Cham, 2016. Springer International Publishing.

[YZ16] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise lpn. In *Annual International Cryptology Conference*, pages 214–243. Springer, 2016.

[YZW+17] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Collision Resistant Hashing from Sub-exponential Learning Parity with Noise, 2017.

A Algorithms for Batch LPN

In this section, we design an algorithm for solving Batch LPN whenever the noise distribution $\mathcal{D}$ assigns sufficiently small probability to at least one point. Our analysis uses the algorithm by Arora and Ge [AG11] as a black box, which solves LPN when the noise support is contained within the zero set of a low-degree polynomial. Their result can be summarized as follows:

Theorem A.1 ([AG11], Informal)**.** *If there is a nonzero polynomial $f : \mathbb{F}_2^k \to \mathbb{F}_2$ of degree d, such that $\mathrm{supp}(\mathcal{D})$ is contained in the zero set of f, then there is an algorithm that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ in time $\mathrm{poly}(n^k)$ with $O(n^d 2^{k+d})$ samples with high probability.*

Definition A.2. Let $p(x) = \sum_{S\subseteq[n],|S|\le d} c_S \prod_{i\in S} x_i$ be a multilinear polynomial of degree d in n variables where the coefficients c_S's are in $\mathrm{GF}(2)$. The linearization of p, denoted $L(p)$, is a linear function over the variables y_S, where S ranges over subsets of $[n]$ of size at most d: $L(p) = \sum_{S\subseteq[n],|S|\le d} c_S y_S$.

We will assume there is a variable $y_\emptyset$ that is always 1, so the number of new variables is

$$N + 1 = \sum_{i=0}^{d} \binom{n}{i}.$$

Lemma A.3. *[AG11] The linear system obtained by $10N2^{m+d}$ samples always has at least one solution, and with high probability (over the oracle's random choices) all solutions to the system satisfy $y_{\{i\}} = u_i$.*

In the adversarial model studied in [AG11], the admissible noise patterns are specified by a multilinear polynomial $P(\eta)$, and the correctness of their algorithm requires an additional technical condition: whenever $P(\alpha) = P(\beta) = 0$, the sum $\alpha + \beta$ must avoid a particular vector ρ. Intuitively, this restriction is necessary because an adversary may examine the sample vectors a_i and then select a noise pattern tailored to confuse the learner.

In contrast, in the random-noise setting considered in our work, the noise is sampled independently of the queries, and hence the learner never faces an adversarial choice of η. In this case, every nontrivial support $S \subsetneq \{0,1\}^k$ can be represented as the zero set of some polynomial of degree at most k. The following lemma formalizes this observation.

Lemma A.4. *Let $k \geq 1$ and let $S \subseteq \mathbb{F}_2^k$ be a nonempty proper subset. Then there exists a nonconstant multivariate polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_k]$ of degree at most k such that, for every $x \in \mathbb{F}_2^k$,*

$$f(x) = 0 \quad \Longleftrightarrow \quad x \in S. \tag{5}$$

In the classical low-noise regime of LPN, a well-known generic strategy is to repeatedly sample fresh LPN instances until obtaining a subset of samples that happens to contain no errors. Concretely, one keeps drawing pairs (a_i, y_i), and repeatedly selects a batch of $\Theta(n)$ equations. For each batch, one treats the samples as if they were noiseless and attempts to solve for the secret via Gaussian elimination. If the chosen batch contains no flipped labels, the resulting system uniquely reveals the secret; otherwise, the attempt fails, and the algorithm simply samples another batch. Since each sample is independently uncorrupted with probability $1 - \eta$, a uniformly chosen batch of size n is fully clean with probability $(1 - \eta)^n$, implying an expected number of trials of $(1 - \eta)^{-n} \approx e^{\eta n}$.

Our approach follows a similar sampling method, but we leverage the linearization technique [AG11]. We repeatedly draw many random batches of k noisy samples and, for each batch, apply the transformation that converts the k correlated labels into a single higher-dimensional linear constraint determined by the zero-set polynomial of the noise support. Because in the random-noise setting, any non-full support admits a representing polynomial, every batch can be processed through this linearization. This allows us to solve on each randomly chosen batch and recover the secret once a batch yields a consistent lifted system. In this sense, our algorithm can be viewed as a randomized batch-wise extension of the classical low-noise strategy, replacing the requirement of "no flips" with the more flexible condition that the batch polynomial correctly captures the noise pattern.

Theorem A.5. *Let $\mathcal{D}$ be a noise distribution such that $P_{\mathcal{D}}(\mathbf{z}) \leq \varepsilon$ for some $\mathbf{z} \in \mathbb{F}_2^k$. Then there is an algorithm for $\mathsf{LPN}_{n,k,\mathcal{D}}$ that makes $\exp\big(O\big(\varepsilon n^k 2^{2k}\big)\big)$ oracle queries and runs in time $\exp\big(O\big(\varepsilon n^k 2^{2k}\big)\big)$.*