Hardness and Algorithms for Batch LPN under Dependent Noise

Xin Li[1⋆], Songtao Mao[1⋆⋆], and Zhaienhe Zhou[2⋆⋆⋆]

[1] Department of Computer Science, Johns Hopkins University
lixints@cs.jhu.edu, smao13@jhu.edu
[2] Department of Computer Science and Engineering, University of Michigan
zehzhou@umich.edu

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 $\mathcal{D}$ over $\mathbb{F}_2^k$ instead of from an i.i.d. product distribution. This model captures a broad range of correlated or structured noise patterns studied in cryptography and learning theory. Consequently, understanding which dependent-noise 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 identify broader classes of noise distributions $\mathcal{D}$ that preserve LPN hardness, extending the prior hardness results of Golowich, Moitra, and Rohatgi (FOCS 2024). We show hardness in three regimes:

1. If $\mathcal{D}$ satisfies a mild Fourier-analytic condition $\sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| \leq 2\varepsilon$, then Batch LPN is as hard as standard LPN with noise rate $1/2-\varepsilon$;
2. If $\mathcal{D}$ is $\Omega(\eta \cdot k2^{-k})$-dense (i.e., every error pattern occurs with probability at least $\Omega(\eta \cdot k2^{-k})$) for $\eta < 1/k$, then Batch LPN is as hard as standard LPN with noise rate η;
3. If $\mathcal{D}$ is a δ-Santha–Vazirani source, then Batch LPN is as hard as standard LPN with noise rate $1/2-\varepsilon$ whenever $\delta \leq O(2^{-k/2}\varepsilon)$, improving the previous $O(2^{-k}\varepsilon)$ dependence (in Golowich et al.).

On the algorithmic side, we extend Arora and Ge's (ICALP 2011) linearization attack to show that Batch LPN can be solved when the noise distribution assigns sufficiently small probability to at least one point, complementing our second hardness result. We also observe that a nontrivial Fourier coefficient with magnitude close to one yields a simple reduction to low-noise LPN, giving an algorithmic counterpart to our first hardness result.

Our reductions are based on random affine transformations and are analyzed through the lens of Fourier analysis, providing a general framework for studying dependent-noise LPN variants.

⋆ Supported in part by NSF Award CCF-2127575
⋆⋆ Supported in part by NSF Award CCF-2127575
⋆⋆⋆ Work done while an undergraduate at the University of Science and Technology of China, during a visit to Johns Hopkins University.

1 Introduction

The Learning Parity with Noise (LPN) problem asks an algorithm to solve a noisy system of linear binary equations, or equivalently, to 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 [20], including the construction of pseudorandom generators [8,2], pseudorandom functions [15], hashing [24], and pseudorandom codes [1].

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 (Standard LPN distribution). *Let $n \in \mathbb{N}$, $\eta \in [0, 1/2)$ and let $\mathbf{sk} \in \mathbb{F}_2^n$ be the hidden secret vector.*

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

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

Despite its simple algebraic structure and extensive study, LPN is widely believed to be computationally hard. The most efficient algorithm for solving standard LPN with a constant noise rate remains the classical BKW algorithm [9], which requires roughly $2^{O(n/\log n)}$ time and samples. Although subsequent works have improved BKW in terms of sample complexity [19] and practical runtime [18], these optimizations do not improve the asymptotic time complexity.

In the low-noise regime, a simple algorithm runs in $\exp(O(\eta n))$ time by repeatedly sampling sets of approximately n equations until obtaining a set containing no corrupted labels. Such an event occurs with probability $(1-\eta)^n \approx \exp(-\eta n)$. In particular, no polynomial-time algorithm is known for standard LPN at noise rate $\eta = \Theta(\log^2 n/n)$, a regime in which LPN is conjectured to remain computationally hard [11].

LPN Variants. To better understand the hardness landscape of LPN and to construct cryptographic primitives with specific properties, a variety of LPN variants have been studied. On the algorithmic side, several works have developed more efficient procedures for specialized variants, including Sparse LPN [14,6], LSPN [21,12], and batch LPN with structured noise under appropriate conditions [3]. On the hardness side, variants of LPN have yielded implications in a wide range of settings. For example, researchers have conjectured intractability for the dense–sparse LPN problem [13] for cryptographic constructions, and recent work on Batch LPN with mildly dependent noise [17] establishes hardness results that give reductions for several learning problems. Moreover, [23] 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, [4] proved a lower bound on the running time for Sparse LPN algorithms with an unbounded number of samples.

Batch LPN and Dependent Noise. In the Batch LPN variant, a batch of k samples is produced simultaneously, and the corresponding 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 shared among samples within a batch, 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 [16]. 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, [10] shows the hardness of LWE with a non-uniform but high-entropy secret distribution. Similar hardness has also been conjectured in LPN [22]. Other robustness results include [7], 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 samples drawn from $\mathsf{LPN}_{n,k,\mathcal{D}}(\mathbf{sk})$. We call this problem *Search* $\mathsf{LPN}_{n,k,\mathcal{D}}$, or simply $\mathsf{LPN}_{n,k,\mathcal{D}}$. Throughout the paper, the distribution $\mathcal{D}$ is part of the problem specification and is given explicitly to the algorithm by its probability mass function.

Definition 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 query vectors $(\mathbf{u}_i)_{i=1}^k$ and the noise vector $(e_i)_{i=1}^k$ are independent. Different batches are sampled independently from the same distribution $\mathcal{D}$; thus, dependence is allowed arbitrarily within each batch, but not across different batches.

Arora and Ge [3] 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 lie in the zeros of a predefined non-zero polynomial over $\mathbb{F}_2[e_1, e_2, \ldots e_k]$. An example of such an 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 an 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 [3] can be interpreted as showing that LPN is tractable for certain noise distributions with strong dependence and small batch size, Golowich, Moitra, and Rohatgi [17,16] conversely showed hardness when the noise is generated from a block δ-Santha–Vazirani (SV) source, a weakly dependent source where each bit may depend weakly on the preceding noise bits.

Definition 3 (δ-Santha–Vazirani source). *A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called a δ-Santha–Vazirani (SV) source if*

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

for every $i \in [k]$ and every prefix $(x'_1, \ldots, x'_{i-1}) \in \mathbb{F}_2^{i-1}$ with positive probability.

Their result was proven via an inductive reduction from standard LPN with 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 [5] also shows hardness for Batch LPN with $k = 2$ and a particular distribution.

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

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

In this work, we develop a unified Fourier-analytic framework for studying Batch LPN with dependent noise. Our results identify several broad sufficient conditions for hardness, together with complementary algorithmic guarantees. Although we focus on the binary setting, our Fourier-based hardness reduction also extends to Batch LWE over prime moduli.

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 high-noise regime. For a distribution $\mathcal{D}$ over $\mathbb{F}_2^k$, write $P_{\mathcal{D}}(z) = \Pr_{e\sim\mathcal{D}}[e = z]$ and $\widehat{P}_{\mathcal{D}}(\mathbf{s}) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[(-1)^{\langle \mathbf{s},\mathbf{x}\rangle}\right]$ for its Fourier coefficient at $\mathbf{s} \in \mathbb{F}_2^k$. We have the following theorem.

Theorem 1 (Theorem 5, informal). *If the noise distribution $\mathcal{D}$ satisfies*

$$\sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| \leq 2\varepsilon,$$

then solving $\mathsf{LPN}_{n,k,\mathcal{D}}$ is as hard as solving standard $\mathsf{LPN}_{n,1/2-\varepsilon}$.

This condition is satisfied by several natural classes of distributions. For example, any δ-biased distribution $\mathcal{D}$, in the sense of Definition 5, satisfies the condition in Theorem 1 whenever $\delta \leq \frac{2\varepsilon}{2^k-1}$.

We also compare this Fourier condition with the Santha–Vazirani framework of [16]. Every δ-SV source is 2δ-biased, but the converse is false: a small-bias distribution need not satisfy any comparable SV condition.

Remark 1. Every δ-SV source is 2δ-biased. Indeed, for any nonzero $s \in \mathbb{F}_2^k$, let $j = \max\{i : s_i = 1\}$. Then

$$\widehat{P}_{\mathcal{D}}(s) = \mathbb{E}\left[(-1)^{\sum_{i<j} s_i x_i}\mathbb{E}\left[(-1)^{x_j} \mid x_1, \ldots, x_{j-1}\right]\right].$$

By the SV condition, the inner conditional expectation has absolute value at most 2δ, and hence $|\widehat{P}_{\mathcal{D}}(s)| \leq 2\delta$. In contrast, a δ-biased source need not satisfy

an SV condition with parameter $O(\delta)$. For example, consider the mixture that samples $\mathbf{x}$ uniformly from $\{00\ldots0, 11\ldots1\}$ with probability ε, and samples $\mathbf{x}$ uniformly from $\mathbb{F}_2^k$ with probability $1-\varepsilon$. This distribution has small Fourier bias controlled by the mixture weight, but conditioning on a long prefix can create much stronger local dependence than allowed by an SV source.

The following observation gives an algorithmic counterpart to the above hardness result.

Remark 2 (Algorithm from large Fourier coefficient). Suppose $\max_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| = 1-\gamma$, where $0 \le \gamma < 1$, and let $\mathbf{s} \neq \mathbf{0}$ attain this maximum. Since $\mathcal{D}$ is given explicitly, such a coefficient can be found in $O(k2^k)$ time. XORing the equations indexed by $\{i : s_i = 1\}$ in each batch, and flipping the resulting label if $\widehat{P}_{\mathcal{D}}(\mathbf{s}) < 0$, gives standard $\mathsf{LPN}_{n,\gamma/2}$ samples with the same secret. Thus, the standard low-noise LPN algorithm recovers the secret in $\exp(O(\gamma n)) \cdot \mathrm{poly}(n, k, (1-\gamma)^{-1})$ time once $\mathbf{s}$ is known. In particular, $\gamma = 0$ yields noiseless equations, and $\gamma = O(\log n/n)$ gives polynomial running time after preprocessing. This complements Theorem 1: small Fourier mass implies hardness, while a large Fourier coefficient yields an algorithm, though the conditions concern the L_1 and L_∞ norms, respectively.

Next, we present our reductions from the low-noise regime of standard LPN. Recall that in the low-noise regime, standard LPN is efficiently solvable when $\eta = O(\log n/n)$, while it is conjectured to remain hard at noise rate $\eta = \Theta(\log^2 n/n)$. Within this broader landscape, we consider low-noise but ρ-dense noise distributions, where every point in $\mathbb{F}_2^k$ has probability at least ρ. 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 ρ for which the hardness reduction continues to hold.

Theorem 2 (Theorem 6, informal). *If the noise distribution $\mathcal{D}$ is ρ-dense, meaning*

$$P_{\mathcal{D}}(\mathbf{z}) \ge \rho \qquad \text{for every } \mathbf{z} \in \mathbb{F}_2^k,$$

then solving $\mathsf{LPN}_{n,k,\mathcal{D}}$ is as hard as solving standard $\mathsf{LPN}_{n,\eta}$ for every noise rate η satisfying roughly $\eta \lesssim 2^k\rho/k$.

On the algorithmic side, we show that a rare point in the noise distribution yields an algorithm whose running time depends exponentially on the probability mass of that point.

Theorem 3 (Theorem 7, informal). *If the noise distribution $\mathcal{D}$ satisfies*

$$P_{\mathcal{D}}(\mathbf{z}) \le \tau \qquad \text{for some } \mathbf{z} \in \mathbb{F}_2^k,$$

and suppose $\Delta = d_{\mathrm{TV}}(\mathcal{D}, \mathrm{Unif}(\mathbb{F}_2^k)) > 0$. Then there is an algorithm for $\mathsf{LPN}_{n,k,\mathcal{D}}$ that recovers the secret with high probability using

$$\exp\left(O(\tau n^k 2^{2k})\right) \cdot \mathrm{poly}\left(n^k, \frac{1}{\Delta}\right)$$

queries and time.

Note that if we apply this algorithm to standard $\mathsf{LPN}_{n,\eta}$, which can be seen as $\mathsf{LPN}_{n,k,\mathrm{Ber}(\eta)^k}$, we have $\tau = \eta^k$ by considering the probability of the all-ones error vector. Thus, if $\eta = O(1/n)$ and k is constant, 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 [17], and provide a tighter analysis via Fourier analysis.

Theorem 4 (Theorem 8, 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 [17] requires the condition $\delta \le O(2^{-k}\varepsilon)$, we give a quadratic improvement in the dependence on k.

The results above give complementary hardness and algorithmic guarantees for Batch LPN under dependent noise. The hardness results show that several broad classes of noise distributions preserve the hardness of standard LPN, either in the high-noise regime through Fourier-analytic conditions or in the low-noise regime through density conditions. The algorithmic results identify structured or sparse regimes in which Batch LPN becomes tractable. We summarize these comparisons in Table 1.

Result	Regime for $\mathcal{D}$	Guarantee
Theorem 5	$\sum_{\mathbf{s}\neq\mathbf{0}} \lvert\widehat{P}_{\mathcal{D}}(\mathbf{s})\rvert \le 2\varepsilon$	Hardness from $\mathsf{LPN}_{n,1/2-\varepsilon}$
Theorem 6	$\min_{\mathbf{z}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{z}) \gtrsim k\eta 2^{-k}$	Hardness from $\mathsf{LPN}_{n,\eta}$
Theorem 8	δ-SV source with $\delta \lesssim 2^{-k/2}\varepsilon$	Hardness from $\mathsf{LPN}_{n,1/2-\varepsilon}$
[17,16]	δ-SV source with $\delta \lesssim 2^{-k}\varepsilon$	Hardness from $\mathsf{LPN}_{n,1/2-\varepsilon}$
Remark 2	$\max_{\mathbf{s}\neq\mathbf{0}} \lvert\widehat{P}_{\mathcal{D}}(\mathbf{s})\rvert = 1-\gamma, 0 \le \gamma < 1$	Algorithm in time: $O(k2^k) + \exp(O(\gamma n)) \cdot \mathrm{poly}(n, k, (1-\gamma)^{-1})$
Theorem 7	$\min_{\mathbf{z}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{z}) \le \tau, \Delta > 0$	Algorithm in time: $\exp(O(\tau n^k 2^{2k})) \cdot \mathrm{poly}(n^k, \Delta^{-1})$
[3]	$\mathrm{supp}(\mathcal{D})$ in the zero set of a nonzero low-degree polynomial	Algorithm in time $n^{O(k)}$

Table 1. Summary of algorithmic and hardness results for Batch LPN with dependent noise. Here $\Delta = d_{\mathrm{TV}}(\mathcal{D}, \mathrm{Unif}(\mathbb{F}_2^k))$.

2 Preliminaries

In this section, we fix notation and recall standard Fourier-analytic facts used throughout the paper. All vector spaces are over $\mathbb{F}_2$ unless otherwise stated, and all additions and inner products are taken over $\mathbb{F}_2$. We use lowercase bold

letters, such as $\mathbf{u}, \mathbf{b}, \mathbf{t}, \mathbf{y}$, to denote vectors. The secret vector is denoted by $\mathbf{sk}$. Uppercase bold letters, such as $\mathbf{A}$ and $\mathbf{U}$, denote matrices. We write $\mathbf{0}$ for the all-zero vector, with the dimension clear from context. The letters n, m, and k denote natural numbers. For $k \in \mathbb{N}$, write $[k] := \{1, \ldots, k\}$. For $\mathbf{x} \in \mathbb{F}_2^k$, let $\mathrm{wt}(\mathbf{x}) := |\{i \in [k] : x_i = 1\}|$ denote its Hamming weight.

We use calligraphic letters, such as $\mathcal{D}, \mathcal{R}$, and Greek letters, such as μ, to denote probability distributions. For a distribution $\mathcal{D}$ over a finite set X, we use $P_{\mathcal{D}}(\mathbf{x}) = \Pr_{\mathbf{y} \sim \mathcal{D}}[\mathbf{y} = \mathbf{x}]$ to denote its probability mass function, and $\mathrm{supp}(\mathcal{D}) := \{x \in X : P_{\mathcal{D}}(x) > 0\}$ for its support. A discrete distribution $\mathcal{D}$ over a finite set X is called *ρ-dense* if $\min_{x \in X} P_{\mathcal{D}}(x) \geq \rho$. For a finite set X, we write $\mathrm{Unif}(X)$ for the uniform distribution over X. For $p \in [0, 1]$, $\mathrm{Ber}(p)$ denotes the distribution over $\mathbb{F}_2$ that assigns probability p to 1 and probability $1 - p$ to 0. Thus, if $X \sim \mathrm{Ber}(1/2 - \delta)$, then $\mathbb{E}[(-1)^X] = 2\delta$.

We also use the following lemma and omit its proof.

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

Fourier Analysis on the Boolean Cube For $\mathbf{s}, \mathbf{x} \in \mathbb{F}_2^k$, 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

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

Fourier inversion gives

$$P_{\mathcal{D}}(\mathbf{x}) = 2^{-k} \sum_{\mathbf{s} \in \mathbb{F}_2^k} \widehat{P}_{\mathcal{D}}(\mathbf{s})(-1)^{\langle \mathbf{s}, \mathbf{x} \rangle}.$$

In particular, the values of $\widehat{P}_{\mathcal{D}}(\mathbf{s})_{\mathbf{s} \in \mathbb{F}_2^k}$ uniquely determine the distribution $\mathcal{D}$. We have basic facts that for all $\mathbf{s} \in \mathbb{F}_2^k$, $|\widehat{P}_{\mathcal{D}}(\mathbf{s})| \leq 1$ and $\widehat{P}_{\mathcal{D}}(\mathbf{0}) = 1$.

We also use the following notations to describe a distribution with bounded Fourier coefficient.

Definition 4 (δ-L_1 Fourier-bounded distribution). *A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called δ-L_1 Fourier-bounded if*

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

Definition 5 (δ-biased distribution). *A distribution $\mathcal{D}$ over $\mathbb{F}_2^k$ is called δ-biased if*

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

Proposition 1. *Every δ-biased distribution over $\mathbb{F}_2^k$ is $(2^k - 1)\delta$-L_1 Fourier-bounded.*

3 The Affine Reduction Framework

In this section, we introduce the affine transformation framework used in our reductions. 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 $x \mapsto \mathbf{A}x + \mathbf{b}$ with $\mathbf{A} \in \mathrm{GL}(k, 2)$ and $\mathbf{b} \in \mathbb{F}_2^k$.

Definition 6 ($\mathcal{R}$-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 $\mathsf{LPN}_{n,\eta}(\mathbf{sk})$ is the following procedure:*

1. *Sample $(\mathbf{A}, \mathbf{b}) \sim \mathcal{R}$.*
2. *Draw k independent samples $(\mathbf{u}_i, y_i) \sim \mathsf{LPN}_{n,\eta}(\mathbf{sk})$ 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 $\mathbf{U}' = \mathbf{AU}$ and $\mathbf{y}' = \mathbf{Ay} + \mathbf{b}$. Output the batch $(\mathbf{u}'_i, y'_i)_{i=1}^k$.*

We remark that the new batch of vectors forms a valid Batch LPN.

Proposition 2.

1. *The transformed batch corresponds to the new noise vector $\mathbf{e}' = \mathbf{Ae} + \mathbf{b}$.*
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 $\mathbf{U}' = \mathbf{AU}$ remains uniformly distributed over $\mathbb{F}_2^{k \times n}$ due to the bijectivity of the map $\mathbf{U} \mapsto \mathbf{AU}$.*

The next lemma shows that Fourier coefficients behave cleanly under such transformations.

Lemma 2 (Fourier Coefficients under Affine Reduction). *Let $\mathcal{D}$ be the noise distribution induced by an $\mathcal{R}$-affine reduction from $\mathsf{LPN}_{n,\eta}$. Then, for every $\mathbf{s} \in \mathbb{F}_2^k$,*

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

In particular, when $\eta = 1/2 - \varepsilon$,

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

Proof. Let $\mathbf{e} = (e_1, \ldots, e_k)$ be the original noise vector with independent coordinates $e_i \sim \mathrm{Ber}(\eta)$. By Proposition 2, the transformed 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:

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

Let $\mathbf{t} := \mathbf{A}^\top \mathbf{s}$. Since the coordinates of $\mathbf{e}$ are independent,

$$\mathop{\mathbb{E}}_{\mathbf{e}}\left[(-1)^{\langle \mathbf{t}, \mathbf{e} \rangle}\right] = \prod_{i=1}^{k} \mathop{\mathbb{E}}_{e_i \sim \mathrm{Ber}(\eta)} \left[(-1)^{t_i e_i}\right].$$

If $t_i = 0$, the corresponding factor is 1. If $t_i = 1$, then:

$$\mathop{\mathbb{E}}_{e_i \sim \mathrm{Ber}(\eta)}[(-1)^{e_i}] = (1-\eta) - \eta = 1 - 2\eta.$$

Therefore,

$$\mathop{\mathbb{E}}_{\mathbf{e}}[(-1)^{\langle \mathbf{t}, \mathbf{e} \rangle}] = (1-2\eta)^{\mathrm{wt}(\mathbf{t})} = (1-2\eta)^{\mathrm{wt}(\mathbf{A}^\top \mathbf{s})}.$$

Substituting this into the previous expression yields

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

4 Reduction from High-Noise LPN to Fourier-Bounded Batch Noise

In this section, we present a reduction from standard LPN with noise rate $1/2-\varepsilon$ to Batch LPN whose noise distribution satisfies an L_1 Fourier condition using the affine framework from Section 3. The main idea is to design a distribution $\mathcal{R}$ over affine maps so that the transformed standard LPN noise has exactly the desired batch noise distribution.

Theorem 5. *Let $0 < \varepsilon \leq \frac{1}{2}$, and let $\mathcal{D}$ be a distribution over $\mathbb{F}_2^k$. Suppose*

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

If there is an algorithm A_1 that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ using at most q batch samples, running in time T, and succeeding with probability at least p, then there is an algorithm A_2 that solves $\mathsf{LPN}_{n,\frac{1}{2}-\varepsilon}$ using at most qk samples, running in time $T + q \cdot \mathrm{poly}(2^k, n)$, and succeeding with probability at least p.

Example 1 (δ-biased noise). If $\mathcal{D}$ is δ-biased with $\delta \le \frac{2\varepsilon}{2^k-1}$, then

$$\sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| \le (2^k-1)\delta \le 2\varepsilon.$$

Hence Theorem 5 applies.

Example 2 (Independent noise with non-identical biases). Suppose

$$\mathcal{D} = \otimes_{i=1}^{k} \mathrm{Ber}\left(\frac{1}{2} - \gamma_i\right),$$

where the coordinates are independent but not necessarily identically distributed. Then for every $\mathbf{s} \in \mathbb{F}_2^k$,

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \prod_{i:\mathbf{s}_i=1} 2\gamma_i.$$

Consequently,

$$\sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| = \prod_{i=1}^{k}(1+2|\gamma_i|) - 1.$$

Hence Theorem 5 applies whenever

$$\prod_{i=1}^{k}(1+2|\gamma_i|) - 1 \le 2\varepsilon.$$

Example 3. Let $\mathcal{U}_k$ denote the uniform distribution on $\mathbb{F}_2^k$, and let $\mathcal{Q}$ be an arbitrary distribution over $\mathbb{F}_2^k$. Consider the mixture $\mathcal{D} = (1-\rho)\mathcal{U}_k + \rho\mathcal{Q}$. For every nonzero $\mathbf{s} \in \mathbb{F}_2^k$, the uniform part has zero Fourier coefficient, and hence

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \rho\widehat{P}_{\mathcal{Q}}(\mathbf{s}).$$

Therefore

$$\sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| = \rho \sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{Q}}(\mathbf{s})| \le \rho(2^k-1),$$

since $|\widehat{P}_{\mathcal{Q}}(\mathbf{s})| \le 1$ for every $\mathbf{s}$. Thus Theorem 5 applies whenever $\rho \le \frac{2\varepsilon}{2^k-1}$.

Example 4 (Batch Markovian noise). Let $e_1 \sim \mathrm{Unif}(\mathbb{F}_2)$. For $i = 2, \cdots, k$, let $e_i = e_{i-1} + \xi_i$, where the increments $\xi_2, \ldots, \xi_k$ are independent and distributed as $\xi_i \sim \mathrm{Ber}\left(\frac{1}{2} - \gamma\right)$. This defines a Markov source within each batch. Let $\mathcal{D}$ be the distribution of $\mathbf{e} = (e_1, \cdots, e_k)$. For $\mathbf{s} \in \mathbb{F}_2^k$, write

$$\langle \mathbf{s}, \mathbf{e} \rangle = \left(\sum_{i=1}^{k} \mathbf{s}_i\right) e_1 + \sum_{j=2}^{k} \left(\sum_{i=j}^{k} \mathbf{s}_i\right) \xi_j.$$

Hence

$$\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \begin{cases} 0, & \text{if } \mathrm{wt}(\mathbf{s}) \text{ is odd,} \\ (2\gamma)^{m(\mathbf{s})}, & \text{if } \mathrm{wt}(\mathbf{s}) \text{ is even,} \end{cases}$$

where $m(\mathbf{s}) = \left|\left\{j \in \{2, \cdots, k\} : \sum_{i=j}^{k} \mathbf{s}_i = 1\right\}\right|$. The suffix-parity map appearing in the definition of $m(\mathbf{s})$ gives a bijection from the nonzero even-weight vectors in $\mathbb{F}_2^k$ to the nonzero vectors in $\mathbb{F}_2^{k-1}$. Therefore,

$$\sum_{\mathbf{s}\neq\mathbf{0}} |\widehat{P}_{\mathcal{D}}(\mathbf{s})| = (1 + 2|\gamma|)^{k-1} - 1.$$

Thus Theorem 5 applies whenever

$$(1 + 2|\gamma|)^{k-1} - 1 \leq 2\varepsilon.$$

The proof of Theorem 5 constructs the target distribution one Fourier coefficient at a time. We need two elementary ingredients. The first produces a distribution whose only nonzero Fourier coefficient is at a prescribed character. The second lets us map any nonzero character to a fixed coordinate by an invertible linear transformation. We defer their proofs to Section B.

Lemma 3. *Let $\mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$, and let $\sigma \in \{-1, 1\}$. Then there exists a probability distribution $\mu_{\mathbf{s},\sigma}$ over $\mathbb{F}_2^k$ such that*

$$\widehat{P}_{\mu_{\mathbf{s},\sigma}}(\mathbf{t}) = \begin{cases} 1, & \mathbf{t} = \mathbf{0}, \\ \sigma, & \mathbf{t} = \mathbf{s}, \\ 0, & \textit{otherwise}. \end{cases}$$

Lemma 4. *For any $\mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$ there exists $\mathbf{A} \in \mathrm{GL}(k, 2)$ such that $\mathbf{A}^\top \mathbf{s} = (1, 0, \ldots, 0)^\top$.*

We are now ready to prove Theorem 5.

Proof. We construct a distribution $\mathcal{R}$ over $\mathrm{GL}(k, 2) \times \mathbb{F}_2^k$ such that the $\mathcal{R}$-affine reduction maps standard LPN noise $\mathrm{Ber}(1/2 - \varepsilon)^{\otimes k}$ exactly to the target distribution $\mathcal{D}$. For each nonzero $\mathbf{s} \in \mathbb{F}_2^k$, define

$$q_{\mathbf{s}} := \frac{|\widehat{P}_{\mathcal{D}}(\mathbf{s})|}{2\varepsilon}.$$

By the hypothesis, $\sum_{\mathbf{s}\neq\mathbf{0}} q_{\mathbf{s}} \leq 1$, and we denote $q_{\mathbf{0}} := 1 - \sum_{\mathbf{s}\neq\mathbf{0}} q_{\mathbf{s}}$. The distribution $\mathcal{R}$ is defined by the following branching procedure.

For each nonzero $\mathbf{s}$ with $q_{\mathbf{s}} > 0$, the branch indexed by $\mathbf{s}$ is chosen with probability $q_{\mathbf{s}}$. In this branch, choose a matrix $\mathbf{A}_{\mathbf{s}} \in \mathrm{GL}(k, 2)$ such that

$$\mathbf{A}_{\mathbf{s}}^\top \mathbf{s} = (1, 0, \ldots, 0)^\top,$$

which exists by Lemma 4. Then sample $\mathbf{b} \sim \mu_{\mathbf{s},\sigma_{\mathbf{s}}}$, where $\sigma_{\mathbf{s}} := \mathrm{sgn}(\widehat{P}_{\mathcal{D}}(\mathbf{s}))$, and $\mu_{\mathbf{s},\sigma_{\mathbf{s}}}$ is the distribution from Lemma 3.

The remaining probability mass $q_{\mathbf{0}}$ is assigned to the branch indexed by $\mathbf{0}$. In this branch, set $\mathbf{A} = \mathbf{I}$ and sample $\mathbf{b} \sim \mathrm{Unif}(\mathbb{F}_2^k)$.

By the correctness of the affine reduction, the transformed samples form a valid Batch LPN instance with the same secret. It remains to show that the transformed noise distribution is exactly $\mathcal{D}$. Let $\mathcal{D}_{\text{out}}$ denote the distribution of the transformed noise vector. By Lemma 2, for every $\mathbf{x} \in \mathbb{F}_2^k$,

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

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 $\mathbf{x} \neq \mathbf{0}$.

First consider the branch $\mathbf{0}$. In this branch, $\mathbf{b}$ is uniform over $\mathbb{F}_2^k$, so

$$\mathop{\mathbb{E}}_{\mathbf{b}\sim\text{Unif}(\mathbb{F}_2^k)} \left[(-1)^{\langle \mathbf{x},\mathbf{b}\rangle}\right] = 0.$$

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

Next consider a nonzero branch $\mathbf{s}$. Conditioned on this branch, we have $\mathbf{A} = \mathbf{A}_\mathbf{s}$ and $\mathbf{b} \sim \mu_{\mathbf{s},\sigma_\mathbf{s}}$. Therefore,

$$\mathbb{E}\left[(-1)^{\langle \mathbf{x},\mathbf{b}\rangle}(2\varepsilon)^{\text{wt}(\mathbf{A}^\top \mathbf{x})} \middle| \text{branch } \mathbf{s}\right] = (2\varepsilon)^{\text{wt}(\mathbf{A}_\mathbf{s}^\top \mathbf{x})} \widehat{P}_{\mu_{\mathbf{s},\sigma_\mathbf{s}}}(\mathbf{x}).$$

By Lemma 3, since $\mathbf{x} \neq \mathbf{0}$, this expression is zero unless $\mathbf{x} = \mathbf{s}$. If $\mathbf{x} = \mathbf{s}$, then $\widehat{P}_{\mu_{\mathbf{s},\sigma_\mathbf{s}}}(\mathbf{s}) = \sigma_\mathbf{s}$ and $\text{wt}(\mathbf{A}_\mathbf{s}^\top \mathbf{s}) = 1$. Thus the conditional contribution of branch $\mathbf{s}$ is $\sigma_\mathbf{s} \cdot 2\varepsilon$. Multiplying by the probability $q_\mathbf{s} = |\widehat{P}_\mathcal{D}(\mathbf{s})|/(2\varepsilon)$, the total contribution of this branch is

$$q_\mathbf{s} \cdot \sigma_\mathbf{s} \cdot 2\varepsilon = \widehat{P}_\mathcal{D}(\mathbf{s}).$$

For the fixed nonzero $\mathbf{x}$, all branches except $\mathbf{s} = \mathbf{x}$ contribute zero. Therefore,

$$\widehat{P}_{\mathcal{D}_{\text{out}}}(\mathbf{x}) = \widehat{P}_\mathcal{D}(\mathbf{x}).$$

By Fourier inversion, the Fourier moments determine the distribution, and hence $\mathcal{D}_{\text{out}} = \mathcal{D}$. Thus the affine reduction simulates exact samples from $\mathsf{LPN}_{n,k,\mathcal{D}}$ using standard samples from $\mathsf{LPN}_{n,1/2-\varepsilon}$. Running $\mathcal{A}_1$ on the simulated batch samples gives an algorithm $\mathcal{A}_2$ for $\mathsf{LPN}_{n,1/2-\varepsilon}$ with the same success probability. If $\mathcal{A}_1$ uses q batch samples, then the simulation uses qk standard LPN samples and incurs $q \cdot \text{poly}(2^k, n)$ overhead. This completes the proof.

5 Reduction from Low-Noise LPN to Dense Batch Noise

The construction in Section 4 gives reductions to target distributions satisfying an L_1-Fourier condition. However, this condition does not cover all natural target distributions, even when the source LPN noise rate is very small. In this section, we give a complementary reduction from low-noise LPN to Batch LPN whose noise distribution satisfies a density condition.

The main idea is different from the high-noise reduction. Instead of choosing an affine map that contributes to one Fourier coefficient at a time, we choose

a uniformly random invertible linear map. Averaging over $\mathrm{GL}(k,2)$ scales all nontrivial Fourier coefficients by the same factor. We then compensate for this scaling by adding an independent shift sampled from a carefully chosen distribution.

Theorem 6. *Suppose $\mathcal{D}$ is a ρ-dense distribution over $\mathbb{F}_2^k$. If there is an algorithm A_1 that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ using at most q batch samples, running in time T, and succeeding with probability at least p, then there is an algorithm A_2 that solves $\mathsf{LPN}_{n,\eta}$ using at most qk samples, running in time $T + q \cdot \mathrm{poly}(2^k, n)$, and succeeding with probability at least p, provided that*

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

Remark 3. In the low-noise regime $\eta \ll 1/k$, we have

$$1 - (1-\eta)^k = k\eta + O(k^2\eta^2).$$

Thus (2) is approximately $\eta \lesssim \frac{2^k}{k} \cdot \min_{\mathbf{z}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{z})$. Since $\min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z}) \le 2^{-k}$, with equality if and only if $\mathcal{D}$ is uniform, this reduction is meaningful precisely in the low-noise regime.

The proof relies on two simple facts. Averaging over a uniformly random invertible matrix scales every nontrivial Fourier coefficient by the same factor. We then choose an auxiliary shift distribution to compensate for this scaling. The next two lemmas formalize these two steps. We defer their proofs to Section B.

Lemma 5 (Averaging over a random invertible matrix). *Let $\alpha \in [-1,1]$, and let $\mathbf{A}$ be uniformly distributed over $\mathrm{GL}(k,2)$. Then for any fixed $\mathbf{s} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$,*

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

Lemma 6 (Fourier scaling). *Let $\mathcal{D}$ be a distribution on $\mathbb{F}_2^k$. For $a \ge 0$, define*

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

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 whenever either $\mathcal{D}$ is uniform, or

$$a \le \frac{1}{1 - 2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})}.$$

We are now ready to prove Theorem 6.

Proof. Given $\eta \in (0, \frac{1}{2})$, let $\alpha \coloneqq 1 - 2\eta \in (0,1)$. Consider the following $\mathcal{R}$-affine reduction applied to $\mathsf{LPN}_{n,\eta}$, where $(\mathbf{A}, \mathbf{b}) \sim \mathcal{R}$ is defined as follows:

1. Sample $\mathbf{A} \sim \mathrm{Unif}(\mathrm{GL}(k,2))$.
2. Sample $\mathbf{b} \sim \mathcal{D}'$ independently of $\mathbf{A}$, where $\mathcal{D}'$ will be chosen below.

By Proposition 2, the transformed query matrix is independent of the new noise vector $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$. It remains to choose $\mathcal{D}'$ so that the transformed noise distribution is exactly $\mathcal{D}$. Let $\mathcal{D}_{\mathrm{out}}$ denote the distribution of the transformed noise. By Lemma 2, for every $\mathbf{s} \in \mathbb{F}_2^k$,

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

where we used the independence of $\mathbf{A}$ and $\mathbf{b}$. 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 5 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

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

With this choice and by Lemma 6, 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

$$\widehat{P}_{\mathcal{D}'}(\mathbf{s}) = a\widehat{P}_{\mathcal{D}}(\mathbf{s}).$$

Therefore,

$$\widehat{P}_{\mathcal{D}_{\mathrm{out}}}(\mathbf{s}) = \frac{2^k(1-\eta)^k - 1}{2^k - 1} \cdot a\widehat{P}_{\mathcal{D}}(\mathbf{s}) = \widehat{P}_{\mathcal{D}}(\mathbf{s})$$

for every nonzero $\mathbf{s}$. Since the equality also holds at $\mathbf{s} = \mathbf{0}$, this gives $\mathcal{D}_{\mathrm{out}} = \mathcal{D}$.

It remains to verify that $\mathcal{D}'$ defined in (4) is a valid distribution. If $\mathcal{D}$ is uniform, then $\mathcal{D}' = \mathcal{D}$, so this is immediate. Otherwise, by Lemma 6, it is enough to show that

$$a \leq \frac{1}{1 - 2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})}.$$

Using the definition (3) of a, this condition is equivalent to

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

Rearranging gives

$$\min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z}) \geq \frac{1 - (1-\eta)^k}{2^k - 1},$$

which is the stated hypothesis (2). Thus $\mathcal{D}'$ is valid.

Running A_1 on the simulated batch samples gives an algorithm A_2 for $\mathsf{LPN}_{n,\eta}$ with the same success probability. If A_1 uses q batch samples, then the simulation uses qk samples and incurs $q \cdot \mathrm{poly}(2^k, n)$ overhead.

5.1 An Algorithmic Upper Bound for Rare Noise Patterns

The reduction above shows that low-noise LPN can generate a broad class of dense Batch LPN distributions. We now state a complementary algorithmic upper bound for the opposite regime, where the batch-noise distribution assigns very small probability to at least one point.

This is analogous to the classical low-noise algorithm for LPN. Rather than requiring a fully clean set of equations, our algorithm samples a sufficiently large collection of batch samples and succeeds when none of their noise vectors equals the rare point. Conditioned on this event, all observed noise vectors lie in a proper subset of $\mathbb{F}_2^k$, and we can apply the Arora–Ge linearization algorithm [3].

Theorem 7. *Let $\mathcal{D}$ be a known distribution over $\mathbb{F}_2^k$. Suppose that $P_{\mathcal{D}}(\mathbf{z}_0) \le \tau$ for some $\mathbf{z}_0 \in \mathbb{F}_2^k$, and let*

$$\Delta := d_{\mathrm{TV}}\big(\mathcal{D}, \mathrm{Unif}(\mathbb{F}_2^k)\big).$$

Assume $\Delta > 0$. Then there is an algorithm for $\mathsf{LPN}_{n,k,\mathcal{D}}$ that, with high probability, recovers the secret using

$$\exp\big(O(\tau N_k 2^{2k})\big) \cdot \mathrm{poly}\left(N_k, \frac{1}{\Delta}\right)$$

oracle queries and time, where $N_k := \sum_{i=1}^{k} \binom{n}{i}$.

The dependence on τ reflects the probability of avoiding the rare point. In one trial, we sample enough batches for the Arora–Ge linearization algorithm to succeed, provided none of the observed noise vectors equals $\mathbf{z}_0$. This happens with probability roughly

$$\exp(-O(\tau N_k 2^{2k})).$$

Repeating this many times yields a successful trial with high probability. Finally, since $\mathcal{D}$ is known and $\Delta > 0$, fresh samples can be used to verify the correct candidate secret by distinguishing residual noise distributed as $\mathcal{D}$ from residual noise distributed uniformly.

For completeness, we give the full proof of Theorem 7 in Section A.

6 Reduction to Batch LPN with δ-SV Noise

In this section, we present an improved analysis of the reduction from [17]. We define the LPN noise parameter as $\eta = \frac{1}{2} - \varepsilon$, where ε is the bias. Our result is the following.

Theorem 8. *Suppose $\mathcal{D}$ is a δ-SV source, as in Definition 3, with*

$$\delta \le 2^{-\frac{k-1}{2}} \varepsilon.$$

If there is an algorithm A_1 that solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ using at most q batch samples, running in time T, and with success probability at least p, then there is an algorithm A_2 that solves $\mathsf{LPN}_{n,\eta}$, where $\eta = \frac{1}{2} - \varepsilon$, using at most qk samples, running in time $T + q \cdot \mathrm{poly}(2^k, n)$, and with success probability at least p.

This improves the dependence of δ on k from $\Theta(2^{-k}\varepsilon)$ in [17] to $\Theta(2^{-k/2}\varepsilon)$. For completeness, we first give an overview of their reduction, and then prove our improved result.

Overview of the reduction in [17]. The reduction simulates samples $(\mathbf{u}_i, y_i = \langle \mathbf{u}_i, \mathbf{sk}\rangle + e_i)_{i=1}^k$ from $\mathsf{LPN}_{n,k,\mathcal{D}}(\mathbf{sk})$ that shares the same secret key $\mathbf{sk}$, by making queries to a standard $\mathsf{LPN}_{n,\eta}$ oracle.

To construct the first sample $(\mathbf{u}_1, y_1)$, 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 $(\mathbf{u}', y')$ from the $\mathsf{LPN}_{n,\eta}$ oracle. It then sets $(\mathbf{u}_1, y_1) = (\mathbf{u}', y' + e_1')$, 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 $|\delta_1| \le \delta \le \varepsilon$. By Lemma 1, this successfully simulates the first sample.

For the i-th sample $(\mathbf{u}_i, y_i)$, the algorithm first queries a fresh sample $(\mathbf{u}', y')$. It then finds a distribution μ on $\mathbb{F}_2^i$ and samples $\mathbf{F} = (F_0, F_1, \ldots, F_{i-1}) \sim \mu$ such that for every $\mathbf{z} \in \mathbb{F}_2^{i-1}$,

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

Here μ depends only on $\mathcal{D}$ and is independent of the actual values of $(e_j)_{j=1}^{i-1}$. Given such a distribution μ, we set

$$\mathbf{u}_i = \mathbf{u}' + \sum_{j=1}^{i-1} F_j \mathbf{u}_j, \qquad y_i = y' + F_0 + \sum_{j=1}^{i-1} F_j y_j.$$

The resulting noise is

$$e' + F_0 + \sum_{j=1}^{i-1} F_j e_j,$$

where $e' \sim \mathrm{Ber}(1/2 - \varepsilon)$ is the fresh LPN noise. Conditioned on the previous noise prefix $(e_1, \cdots, e_{i-1}) = \mathbf{z}$, the choice of μ and Lemma 1 ensure that this new noise bit has exactly the desired conditional distribution of e_i given the prefix $\mathbf{z}$. Moreover, $\mathbf{u}_i$ is independent of all previous vectors $(\mathbf{u}_j)_{j=1}^{i-1}$ because it contains the fresh independent uniform vector $\mathbf{u}'$.

The dependence of δ on k comes from the existence of μ. In [17], 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 7 (Improved Lemma 3.2 from [17]). *Fix $r \ge 0$, and pick any function $f : \mathbb{F}_2^r \to [0,1]$ satisfying*

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

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

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

Moreover, given the truth table of f, one can sample from μ in time $2^{O(r)}$.

Proof. Let $g(\mathbf{z}) \coloneqq 1 - 2f(\mathbf{z})$. The assumption gives

$$\sum_{\mathbf{z}\in\mathbb{F}_2^r} g(\mathbf{z})^2 \leq 1.$$

We will construct μ such that its signed point-mass differences correspond to the Fourier coefficients of g. Since

$$g(\mathbf{z}) = 2^{-r} \sum_{\mathbf{s}\in\mathbb{F}_2^r} \widehat{g}(\mathbf{s})(-1)^{\langle \mathbf{z},\mathbf{s}\rangle}.$$

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

$$\begin{aligned}\Pr_{\mathbf{F}\sim\mu}\left(F_0 + \sum_{j=1}^{r} \mathbf{z}_j F_j = 1\right) &= \mathbb{E}_{\mathbf{F}\sim\mu}\left[\frac{1-(-1)^{F_0+\sum_{j=1}^r \mathbf{z}_j F_j}}{2}\right] \\ &= \frac{1}{2}\left(1 - \mathbb{E}_{\mathbf{F}\sim\mu}\left[(-1)^{F_0}(-1)^{\sum_{j=1}^r \mathbf{z}_j F_j}\right]\right).\end{aligned}$$

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:

$$\mathbb{E}_{\mathbf{F}\sim\mu}\left[(-1)^{F_0}(-1)^{\sum_{j=1}^r \mathbf{z}_j F_j}\right] = g(\mathbf{z}).$$

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

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

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

$$\mu(0,\mathbf{s}) - \mu(1,\mathbf{s}) = 2^{-r}\widehat{g}(\mathbf{s}).$$

We define the weights as follows:

$$(\mu(0,\mathbf{s}), \mu(1,\mathbf{s})) \coloneqq \begin{cases} (2^{-r}|\widehat{g}(\mathbf{s})|, 0) & \text{if } \widehat{g}(\mathbf{s}) > 0, \\ (0, 2^{-r}|\widehat{g}(\mathbf{s})|) & \text{if } \widehat{g}(\mathbf{s}) \leq 0. \end{cases}$$

This assignment ensures $\mu(0,\mathbf{s}) - \mu(1,\mathbf{s}) = 2^{-r}\widehat{g}(\mathbf{s})$. It remains to show that these weights can form a valid probability distribution. The total mass required is

$$M \coloneqq \sum_{\mathbf{s}\in\mathbb{F}_2^r} (\mu(0,\mathbf{s}) + \mu(1,\mathbf{s})) = 2^{-r} \sum_{\mathbf{s}\in\mathbb{F}_2^r} |\widehat{g}(\mathbf{s})|.$$

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

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

Thus, $M \le 1$. If $M < 1$, we add the remaining probability mass $1 - M$ equally 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 $2^{O(r)}$ time, we can sample from μ within the required time bounds.

With this lemma, we are now ready to verify Theorem 8.

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

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

For prefixes $\mathbf{z}$ with zero probability under $\mathcal{D}$, the conditional probability below may be defined arbitrarily, say as $1/2$, since those prefixes are never reached in the simulation. 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 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 7 to show the existence of μ.

Applying this construction inductively for $i = 1, \ldots, k$ therefore simulates one batch from $\mathsf{LPN}_{n,k,\mathcal{D}}$ using k fresh samples from $\mathsf{LPN}_{n,\eta}$. Repeating this simulation for q oracle query made by A_1 gives an algorithm for $\mathsf{LPN}_{n,\eta}$ with the same success probability, and the additional running time is $q \cdot \mathrm{poly}(2^k, n)$.

7 Technical Limitations and Future Directions

Our reductions are based on a particularly simple way of transforming the noise distribution: for each batch, we apply a random affine map $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$, where $(\mathbf{A}, \mathbf{b})$ 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: randomized invertible affine transformations cannot decrease the entropy of the noise source.

Lemma 8. *Let* $\mathbf{e}$ *be a random variable over* $\mathbb{F}_2^k$*, and let* $(\mathbf{A}, \mathbf{b})$ *be a random variable over* $\mathrm{GL}(k, 2) \times \mathbb{F}_2^k$*, sampled independently of* $\mathbf{e}$*. Define* $\mathbf{e}' = \mathbf{A}\mathbf{e} + \mathbf{b}$*. Then* $H(\mathbf{e}') \geq H(\mathbf{e})$*. More precisely,*

$$H(\mathbf{e}') = H(\mathbf{e}) + I(\mathbf{e}'; (\mathbf{A}, \mathbf{b})),$$

where H *denotes Shannon entropy and* I *denotes mutual information.*

Proof. Condition on a fixed affine map. Since $\mathbf{A} \in \mathrm{GL}(k, 2)$, the map $\mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{b}$ is a bijection on $\mathbb{F}_2^k$. Therefore, applying this map to $\mathbf{e}$ does not change entropy. Averaging over $(\mathbf{A}, \mathbf{b})$, and using that $(\mathbf{A}, \mathbf{b})$ is independent of $\mathbf{e}$, gives $H(\mathbf{e}'|\mathbf{A}, \mathbf{b}) = H(\mathbf{e})$. By the chain rule for entropy,

$$H(\mathbf{e}') = H(\mathbf{e}'|(\mathbf{A}, \mathbf{b})) + I(\mathbf{e}'; (\mathbf{A}, \mathbf{b})) = H(\mathbf{e}) + I(\mathbf{e}'; (\mathbf{A}, \mathbf{b})) \geq H(\mathbf{e}).$$

Equality holds precisely when $I(\mathbf{e}'; (\mathbf{A}, \mathbf{b})) = 0$.

This lemma shows an entropy barrier for affine reductions. However, the entropy barrier does not give a full characterization of the distributions reachable by affine reductions. There is also a Fourier-analytic barrier. In the high-noise setting, Lemma 2 implies that affine reductions from $\mathsf{LPN}_{n,1/2-\varepsilon}$ cannot produce target distributions with a nontrivial Fourier coefficient of absolute value larger than 2ε. This rules out, for example, distributions with strong parity constraints, such as the uniform distribution on an affine hyperplane, which has a nonzero Fourier coefficient of magnitude 1.

Our L_1-Fourier condition should be viewed as a sufficient condition but not a necessary condition. Indeed, the identity affine map already produces the independent product distribution $\mathrm{Ber}(1/2 - \varepsilon)^{\otimes k}$, whose nontrivial Fourier coefficients are $(2\varepsilon)^{\mathrm{wt}(\mathbf{s})}$. The total nontrivial Fourier mass is

$$\sum_{\mathbf{s} \neq \mathbf{0}} (2\varepsilon)^{\mathrm{wt}(\mathbf{s})} = (1 + 2\varepsilon)^k - 1,$$

which may be much larger than 2ε. Thus, a precise characterization of the image of randomized affine reductions remains open.

A second limitation of our current analysis is that it focuses on Batch LPN instances in which the noise is generated independently across batches from a fixed 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 [16] and [3]. However, many forms of dependent noise arising in machine learning and cryptographic applications are not block sources. Instead, they may involve long-range temporal correlations, such as Markov dependencies, higher-order dependencies across batches, or global constraints that cannot be expressed as a single distribution over $\mathbb{F}_2^k$. A simple example illustrates this distinction. Suppose that a single random vector $C \sim \mathrm{Unif}(\mathbb{F}_2^k)$is sampled once and then reused as the noise vector in every batch. Then each individual batch has a uniform noise marginal, and hence all nontrivial one-batch Fourier coefficients vanish. Nevertheless, the noise is perfectly correlated across time. The resulting

LPN instance is easy: the samples can be viewed as a noiseless linear system in the secret $\mathbf{sk}$ together with the k fixed offsets given by the coordinates of C. With $O(n+k)$ independent equations, Gaussian elimination recovers both the secret and these offsets with high probability. This example shows that temporal dependence can make the problem easier even when every one-batch marginal distribution is perfectly uniform. Therefore, any extension beyond the block-i.i.d. Batch LPN model must control not only the marginal distribution of each batch noise vector, but also the correlations among noise vectors at different times. Our current algorithms and reductions do not immediately extend to these more general dependent-noise models. Developing a theory for such noise processes is a promising direction for future work.

References

1. Alrabiah, O., Ananth, P., Christ, M., Dodis, Y., Gunn, S.: Ideal Pseudorandom Codes. In: Proceedings of the 57th Annual ACM Symposium on Theory of Computing. pp. 1638–1647. ACM, Prague Czechia (Jun 2025). https://doi.org/10.1145/3717823.3718309
2. Applebaum, B., Ishai, Y., Kushilevitz, E.: On Pseudorandom Generators with Linear Stretch in NC0. computational complexity **17**(1), 38–69 (Apr 2008). https://doi.org/10.1007/s00037-007-0237-6
3. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: International Colloquium on Automata, Languages, and Programming. pp. 403–415. Springer (2011)
4. Bangachev, K., Bresler, G., Tiegel, S., Vaikuntanathan, V.: Near-Optimal Time-Sparsity Trade-Offs for Solving Noisy Linear Equations. In: Proceedings of the 57th Annual ACM Symposium on Theory of Computing. pp. 1910–1920. ACM, Prague Czechia (Jun 2025). https://doi.org/10.1145/3717823.3718284
5. Bartusek, J., Lepoint, T., Ma, F., Zhandry, M.: New Techniques for Obfuscating Conjunctions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, vol. 11478, pp. 636–666. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_22
6. Basu, A., Hsieh, J.T., Lin, A.D., Manohar, P.: Solving random planted csps below the $n^{k/2}$ threshold. arXiv preprint arXiv:2507.10833 (2025)
7. Bellizia, D., Hoffmann, C., Kamel, D., Liu, H., Méaux, P., Standaert, F.X., Yu, Y.: Learning parity with physical noise: Imperfections, reductions and fpga prototype. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 390–417 (2021)
8. Blum, A., Furst, M., Kearns, M., Lipton, R.J.: Cryptographic Primitives Based on Hard Learning Problems. In: Goos, G., Hartmanis, J., Stinson, D.R. (eds.) Advances in Cryptology — CRYPTO' 93, vol. 773, pp. 278–291. Springer Berlin Heidelberg, Berlin, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_24
9. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM **50**(4), 506–519 (Jul 2003). https://doi.org/10.1145/792538.792543
10. Brakerski, Z., Döttling, N.: Hardness of LWE on General Entropic Distributions (2020)

11. Brakerski, Z., Lyubashevsky, V., Vaikuntanathan, V., Wichs, D.: Worst-Case Hardness for LPN and Cryptographic Hashing via Code Smoothing. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, vol. 11478, pp. 619–635. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_21
12. Chen, X., Shu, W., Zhou, Z.: Algorithms for Sparse LPN and LSPN Against Low-noise (extended abstract). In: Proceedings of Thirty Eighth Conference on Learning Theory. pp. 1091–1093. PMLR (Jul 2025)
13. Dao, Q., Jain, A.: Lossy Cryptography from Code-Based Assumptions. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, vol. 14922, pp. 34–75. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-68382-4_2
14. Feige, U., Kim, J., Ofek, E.: Witnesses for non-satisfiability of dense random 3CNF formulas. In: 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). pp. 497–508. IEEE, Berkeley, CA, USA (2006). https://doi.org/10.1109/FOCS.2006.78
15. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM **33**(4), 792–807 (Aug 1986). https://doi.org/10.1145/6490.6503
16. Golowich, N., Moitra, A., Rohatgi, D.: Exploration is harder than prediction: Cryptographically separating reinforcement learning from supervised learning. In: 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS). pp. 1953–1967. IEEE (2024)
17. Golowich, N., Moitra, A., Rohatgi, D.: On learning parities with dependent noise. arXiv preprint arXiv:2404.11325 (2024)
18. Levieil, É., Fouque, P.A.: An improved lpn algorithm. In: International conference on security and cryptography for networks. pp. 348–359. Springer (2006)
19. Lyubashevsky, V.: 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. pp. 378–389. Springer (2005)
20. Pietrzak, K.: Cryptography from Learning Parity with Noise. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012: Theory and Practice of Computer Science, vol. 7147, pp. 99–114. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27660-6_9
21. Valiant, G.: 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). https://doi.org/10.1145/2728167
22. Yao, N., Yu, Y., Li, X., Gu, D.: On the robustness of learning parity with noise. In: Lam, K.Y., Chi, C.H., Qing, S. (eds.) Information and Communications Security. pp. 99–106. Springer International Publishing, Cham (2016)
23. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise lpn. In: Annual International Cryptology Conference. pp. 214–243. Springer (2016)
24. Yu, Y., Zhang, J., Weng, J., Guo, C., Li, X.: Collision Resistant Hashing from Sub-exponential Learning Parity with Noise (2017)

A Algorithm for Batch LPN

In this section, we prove the algorithmic upper bound for Batch LPN when the noise distribution assigns small probability to at least one point. We first restate the theorem.

Theorem 7. *Let $\mathcal{D}$ be a known distribution over $\mathbb{F}_2^k$. Suppose that $P_{\mathcal{D}}(\mathbf{z}_0) \le \tau$ for some $\mathbf{z}_0 \in \mathbb{F}_2^k$, and let*

$$\Delta := d_{\mathrm{TV}}\big(\mathcal{D}, \mathrm{Unif}(\mathbb{F}_2^k)\big).$$

Assume $\Delta > 0$. Then there is an algorithm for $\mathsf{LPN}_{n,k,\mathcal{D}}$ that, with high probability, recovers the secret using

$$\exp\big(O(\tau N_k 2^{2k})\big) \cdot \mathrm{poly}\left(N_k, \frac{1}{\Delta}\right)$$

oracle queries and time, where $N_k := \sum_{i=1}^{k} \binom{n}{i}$.

The proof uses the linearization algorithm of Arora and Ge [3] as a black box. Their algorithm applies when the support of the noise distribution is contained in the zero set of a low-degree polynomial. In the adversarial-noise model, the adversary may choose the noise after observing the sample vectors, and the analysis requires an additional condition on the zero set of the polynomial. In our setting, the noise is sampled independently of the query vectors. We therefore use the following random-noise version of their result.

Theorem 9 (Random-noise version of Arora–Ge [3]). *Let $P : \mathbb{F}_2^k \to \mathbb{F}_2$ be a nonzero multilinear polynomial of degree d, and let the noise vector in each batch be sampled independently from a distribution $\mathcal{D}$ over $\mathbb{F}_2^k$, independently of the query vectors. Suppose*

$$\mathrm{supp}(\mathcal{D}) \subseteq \{\mathbf{e} \in \mathbb{F}_2^k : P(\mathbf{e}) = 0\}.$$

Then the Arora–Ge linearization algorithm solves $\mathsf{LPN}_{n,k,\mathcal{D}}$ using

$$O(N_d 2^{k+d}) \qquad \textit{batch samples, where} \qquad N_d = \sum_{i=1}^{d} \binom{n}{i},$$

with high probability.

We briefly recall the linearization step underlying the theorem. Given a batch sample $(\mathbf{u}_1, b_1), \ldots, (\mathbf{u}_k, b_k)$, the polynomial constraint $P(\mathbf{e}) = 0$ becomes, for a candidate secret $\mathbf{z}$,

$$P(b_1 + \langle \mathbf{u}_1, \mathbf{z} \rangle, \ldots, b_k + \langle \mathbf{u}_k, \mathbf{z} \rangle) = 0.$$

This is a polynomial equation in the coordinates of $\mathbf{z}$. The linearization step replaces each monomial in the coordinates of $\mathbf{z}$ by a new variable.

Definition 7 (Linearization [3]). *Let*

$$p(x) = \sum_{S \subseteq [n], |S| \leq d} c_S \prod_{i \in S} x_i$$

be a multilinear polynomial of degree d in n variables, where the coefficients c_S are in $\mathbb{F}_2$. The linearization of p, denoted $L(p)$, is the linear function over the variables y_S, where S ranges over subsets of $[n]$ of size at most d, defined by $L(p) = \sum_{S \subseteq [n], |S| \leq d} c_S y_S$.

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

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

Arora–Ge show that sufficiently many such linearized constraints force every solution to agree with the true secret on the degree-one variables.

Lemma 9 (Uniqueness of Solutions [3]). *The linear system obtained from $10N_d 2^{k+d}$ batch samples always has at least one solution, and with high probability over the oracle's random choices, all solutions to the system satisfy $y_{\{i\}} = \mathbf{sk}_i$ for every $i \in [n]$.*

Thus, once the observed noise support is contained in the zero set of a known degree-d polynomial, solving the resulting linear system and reporting variables $y_{\{i\}}$ recovers the secret.

We will need the following elementary observation: every nonempty proper subset of the Boolean cube can be represented as the zero set of a polynomial of degree at most k.

Lemma 10. *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 $\mathbf{x} \in \mathbb{F}_2^k$,*

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

Proof. For each $\mathbf{a} \in \mathbb{F}_2^k$, define the indicator polynomial

$$\mathbf{1}_\mathbf{a}(\mathbf{x}) = \prod_{i : \mathbf{a}_i = 1} x_i \prod_{i : \mathbf{a}_i = 0} (1 + x_i).$$

This polynomial has degree at most k, and it equals 1 exactly when $\mathbf{x} = \mathbf{a}$. Define

$$f(\mathbf{x}) = \sum_{\mathbf{a} \in \mathbb{F}_2^k \setminus S} \mathbf{1}_\mathbf{a}(\mathbf{x}).$$

Then $f(\mathbf{x}) = 0$ if and only if $\mathbf{x} \in S$. Since both S and $\mathbb{F}_2^k \setminus S$ are nonempty, the function represented by f is nonconstant.

We are now ready to prove Theorem 7.

Proof. The proof follows the same sampling idea as the classical low-noise algorithm for LPN. In the classical setting, one repeatedly samples a set of roughly n equations and tries Gaussian elimination; the attempt succeeds when the chosen set happens to contain no corrupted labels. This event happens with probability $(1-\eta)^n$, so the expected number of trials is $O((1-\eta)^{-n}) \approx O(\exp(\eta n))$.

Here, the role of a "clean" trial is that none of the sampled batch-noise vectors hits a rare point of the noise distribution. Conditioned on this event, all observed noise vectors lie in a proper subset of $\mathbb{F}_2^k$, which can be represented as the zero set of a low-degree polynomial. We then apply the Arora–Ge linearization algorithm [3] to solve for the secret.

Let

$$\mathbf{z}_0 \in \arg\min_{\mathbf{z}\in\mathbb{F}_2^k} P_{\mathcal{D}}(\mathbf{z}), \qquad p_0 = P_{\mathcal{D}}(\mathbf{z}_0).$$

Set $S = \mathbb{F}_2^k \setminus \{\mathbf{z}_0\}$. By Lemma 10, there is a nonconstant polynomial $f : \mathbb{F}_2^k \to \mathbb{F}_2$ of degree at most k such that $f(\mathbf{z}) = 0$ if and only if $\mathbf{z} \in S$.

Consider one trial of the algorithm. In this trial, we draw

$$M = CN_k 2^{2k}, \qquad N_k = \sum_{i=1}^{k} \binom{n}{i},$$

independent batch samples, where $C > 0$ is a sufficiently large absolute constant, and then run the Arora–Ge linearization algorithm with the polynomial f.

Let $\mathcal{E}$ be the event that none of the M sampled noise vectors is equal to $\mathbf{z}_0$. Since the batch noises are sampled independently, we have

$$\Pr[\mathcal{E}] = (1-p_0)^M \geq \exp(-O(p_0 M)).$$

Conditioned on $\mathcal{E}$, all observed noise vectors lie in $S = \{\mathbf{z} \in \mathbb{F}_2^k : f(\mathbf{z}) = 0\}$, and the sample vectors remain uniform and independent of the noise. Hence the samples satisfy the support condition of Theorem 9, and the Arora–Ge linearization algorithm recovers the secret with high probability conditioned on $\mathcal{E}$.

Repeating the trial $\exp(O(p_0 M))$ times ensures that, with high probability, at least one trial satisfies $\mathcal{E}$ and hence outputs the true secret. Since $p_0 \leq \tau$ and $M = O(N_k 2^{2k})$, the total running time is $\exp\left(O(\tau N_k 2^{2k})\right)$ up to polynomial factors.

Finally, we use fresh samples to verify the candidate secrets produced by the repeated trials. Let L be the number of candidates. From the repetition bound above, $L \leq \exp(O(p_0 M))$. For the true secret, the residual noise vectors are distributed according to $\mathcal{D}$; for any incorrect candidate, they are uniform over $\mathbb{F}_2^k$, because the fresh query vectors are uniform and independent. Since $\Delta = d_{\mathrm{TV}}\big(\mathcal{D}, \mathrm{Unif}(\mathbb{F}_2^k)\big) > 0$ and $\mathcal{D}$ is known, a standard empirical test distinguishes $\mathcal{D}$ from the uniform distribution using $O\left(\frac{\log L + \lambda}{\Delta^2}\right)$ fresh batch samples, with failure probability at most $e^{-\lambda}$. As $\log L = O(p_0 M) \leq O(\tau N_k 2^{2k})$, this verification overhead is absorbed in the claimed

$$\exp(O(\tau N_k 2^{2k})) \cdot \mathrm{poly}\left(N_k, \frac{1}{\Delta}\right)$$

running time.

B Omitted Proofs

B.1 Proofs for Section 4

Proof of Lemma 3. Define

$$P_{\mu_{\mathbf{s},\sigma}}(\mathbf{x}) := 2^{-k}\left(1+\sigma\chi_{\mathbf{s}}(\mathbf{x})\right).$$

Since $\chi_{\mathbf{s}}(\mathbf{x}) \in \{-1,1\}$, this function is nonnegative. Moreover,

$$\sum_{\mathbf{x}\in\mathbb{F}_2^k} P_{\mu_{\mathbf{s},\sigma}}(\mathbf{x}) = 1,$$

because $\mathbf{s} \neq \mathbf{0}$ and hence

$$\sum_{\mathbf{x}\in\mathbb{F}_2^k} \chi_{\mathbf{s}}(\mathbf{x}) = 0.$$

Thus $\mu_{\mathbf{s},\sigma}$ is a probability distribution. For any $\mathbf{t} \in \mathbb{F}_2^k$,

$$\begin{aligned}\widehat{P}_{\mu_{\mathbf{s},\sigma}}(\mathbf{t}) &= \sum_{\mathbf{x}\in\mathbb{F}_2^k} 2^{-k}\left(1+\sigma\chi_{\mathbf{s}}(\mathbf{x})\right)\chi_{\mathbf{t}}(\mathbf{x}) \\ &= 2^{-k}\sum_{\mathbf{x}\in\mathbb{F}_2^k}\chi_{\mathbf{t}}(\mathbf{x}) + \sigma 2^{-k}\sum_{\mathbf{x}\in\mathbb{F}_2^k}\chi_{\mathbf{s}+\mathbf{t}}(\mathbf{x}).\end{aligned}$$

By orthogonality of characters, the first term is 1 if $\mathbf{t} = \mathbf{0}$ and 0 otherwise, while the second term is σ if $\mathbf{t} = \mathbf{s}$ and 0 otherwise. This gives the desired Fourier coefficients.

Proof of Lemma 4. 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. The first column of this identity gives

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

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

B.2 Proofs for Section 5

Proof of Lemma 5 For fixed $\mathbf{s} \neq \mathbf{0}$, the vector $\mathbf{A}^\top\mathbf{s}$ is uniformly distributed over $\mathbb{F}_2^k \setminus \{\mathbf{0}\}$ when $\mathbf{A}$ is uniform over $\mathrm{GL}(k,2)$. Therefore,

$$\mathop{\mathbb{E}}_{\mathbf{A}}[\alpha^{\mathrm{wt}(\mathbf{A}^\top\mathbf{s})}] = \frac{1}{2^k-1}\sum_{\mathbf{t}\neq\mathbf{0}}\alpha^{\mathrm{wt}(\mathbf{t})} = \frac{1}{2^k-1}\sum_{w=1}^{k}\binom{k}{w}\alpha^w = \frac{(1+\alpha)^k-1}{2^k-1}.$$

Proof of Lemma 6 The total mass of $\mathcal{D}'$ is 1, since

$$\sum_{\mathbf{z}\in\mathbb{F}_2^k} \left(P_{\mathcal{D}}(\mathbf{z}) - 2^{-k}\right) = 0.$$

By the linearity of the Fourier transform and the fact that the uniform distribution has Fourier coefficient 1 at $\mathbf{s} = \mathbf{0}$ and 0 elsewhere, the stated Fourier identities follow.

It remains to check nonnegativity. If $\mathcal{D}$ is uniform, so $\mathcal{D}'$ is valid. Otherwise, since $a \geq 0$,

$$\min_{\mathbf{z}} P_{\mathcal{D}'}(\mathbf{z}) = a\left(\min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z}) - 2^{-k}\right) + 2^{-k}.$$

This quantity is nonnegative exactly when

$$a \leq \frac{1}{1 - 2^k \min_{\mathbf{z}} P_{\mathcal{D}}(\mathbf{z})}.$$

Thus $\mathcal{D}'$ is a valid probability distribution under the stated condition.