# Asymptotic Analysis of Ternary Sparse LWE

Byoungchan Chi[1], Nathan Cho[1], Jiseung Kim[2], and Changmin Lee[1]

[1] Korea University
bchan052415@gmail.com
ntckgm@gmail.com
changminlee@korea.ac.kr
[2] Jeonbuk National University
jiseungkim@jbnu.ac.kr

**Abstract.** We present an asymptotic analysis of the ternary variant of Sparse Learning with Errors (spLWE), a structured LWE variant proposed by Jain–Lin–Saha (CRYPTO'24) in which each equation involves only $k \ll n$ of the $n$ secret coordinates, enabling significantly more efficient computation than dense LWE. Unlike standard LWE, the small-secret regime of spLWE is not automatically reducible to its large-secret counterpart, leaving asymptotic hardness unclear, particularly when $k$ is very small.

We develop a two-pronged attack framework that depends explicitly on the sparsity parameter $k$. In the geometric regime $q > 3^k$, each sparse row reduces to a short-vector problem in a $k$-dimensional lattice, yielding complexity $2^{0.292k}$ via a sieving algorithm. In the statistical regime $q \leq 3^k$, we propose a greedy coordinate-recovery attack with running time $O(m \cdot k \cdot 3^k)$, where $m$ is the number of samples.

Heuristically, under mild assumptions, full recovery holds with high probability once the sample size is large enough; the resulting complexity is exponential only in $k$ and otherwise mild (up to polylogarithmic factors), i.e., polynomial in $n$, which makes very small $k$ vulnerable even at large dimensions.

Experiments on toy instances confirm the predicted sharp transition. Complexity comparisons with prior works indicate lower complexity on a few of their parameter sets, while identifying regimes where our method is not applicable.

**Keywords:** Sparse LWE, cryptanalysis, lattice-based cryptography, small-secret LWE, $k$-dependent attack

## 1 Introduction

Since Regev's seminal work [Reg09], Learning with Errors (LWE) has become a central hardness assumption in lattice-based cryptography, supporting a wide range of post-quantum primitives [Gen09,GPV08,LP11,BDK+18,CGGI16,FV12]. In its standard form, LWE asserts that distinguishing samples $(\mathbf{a}, b)$ with $b = \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q$ from uniform is computationally hard, where $\mathbf{a} \in \mathbb{Z}_q^n$ is public, $\mathbf{s} \in \mathbb{Z}_q^n$ is a hidden secret, and $e$ is a small error term. At the same time,

practical deployments repeatedly face the cost of large matrix operations, motivating structured variants that trade algebraic simplicity for efficiency [LPR10, CKKS17,BGV14]. In this context, Jain–Lin–Saha [JLS24] proposed *Sparse LWE* (spLWE), where the public coefficient vector is $k$-sparse ($\mathbf{a}$ has exactly $k$ nonzero coordinates), so that each equation involves only $k$ of the $n$ coefficients. This sparsity can yield substantial implementation gains (roughly $O(n/k)$ in certain representations), but it also creates attack surfaces that are absent (or much weaker) for dense LWE, such as collisions among supports and the possibility that individual sparse equations leak unusually informative local structure. Most importantly for this work, [JLS24] cautions that *small-secret* spLWE is not automatically equivalent to its large-secret counterpart (unlike in standard LWE), leaving the concrete hardness of small-secret spLWE unclear and raising nontrivial questions for parameter selection in the especially appealing *very-small-k* regime.

**Existing analysis and its limitations.** We focus on the ternary setting in which the secret is supported on $\{-1, 0, 1\}$ and the error is drawn from the discrete Gaussian distribution. This regime is a canonical instance of small-secret spLWE and is particularly sensitive to the sparsity parameter $k$: when $k$ is small, each sparse equation may carry substantial local information about only a few secret coordinates. Standard lattice-based attacks [BG14, AGVW17, Alb17, MR09] operate on the full $n$-dimensional lattice and do not exploit this per-equation sparsity, so their cost is largely dominated by $n$ and is not expressed as an explicit function of $k$. Closing this gap calls for attacks that are specifically tailored to the sparse structure, and in particular whose complexity depends explicitly on $k$. Jain–Lin–Saha [JLS24] themselves developed a cryptanalytic framework and proposed heuristic parameter choices, but left the hardness of the small-secret regime—and in particular the ternary case—as an *open problem*. A first step toward resolving this was taken by Kalam–Sarkar–Meier [KSM24], who mounted a combinatorial attack on ternary spLWE via subsystem extraction and meet-in-the-middle techniques. Their analysis highlights a threshold phenomenon governed by the relationship between $3^k$ and the modulus $q$. When $q > 3^k$, each sparse equation individually constrains the secret to a $k$-dimensional sublattice; one can then apply HKZ-reduced basis techniques to recover the secret via a short-vector argument, yielding complexity $2^{O(0.292k)}$. When $q < 3^k$, a single equation no longer suffices, and one is forced to aggregate information across multiple equations—a regime in which combinatorial or statistical methods become necessary.

While this line of work provides valuable insight, its combinatorial nature relies on complex internal trade-off parameters—such as subsystem sizes, splitting rules, and memory constraints—making it difficult to derive a *closed-form asymptotic bound* on the attack complexity purely in terms of the sparsity $k$. Consequently, the existing landscape lacks a rigorous understanding of how the security degrades asymptotically as $k$ decreases, particularly in the aggregation regime $q \leq 3^k$. This motivates our main question:

*Can we establish an attack whose running time depends explicitly and asymptotically on the sparsity parameter $k$?*

### 1.1 This work

We answer this question affirmatively by developing a two-pronged attack framework for ternary sparse LWE, in which the geometric regime $q > 3^k$ and the statistical regime $q \leq 3^k$ are each handled by a strategy tailored to its structure. In the geometric regime $q > 3^k$, each sparse row reduces to a short-vector problem in a $k$-dimensional lattice; the lattice-based attack achieves complexity $\approx 2^{0.292 \cdot k}$ via an sieving algorithm [BDGL16]. In the statistical regime $q \leq 3^k$, we develop a new greedy coordinate-recovery attack with running time $O(m \cdot k \cdot 3^k)$, where $m$ is the number of samples. A heuristic success analysis shows that full recovery holds with high probability once

$$m \geq \frac{C \cdot n \cdot 3^{k-1} \cdot \sigma}{k \cdot q} \log(3n)$$

for a universal constant $C > 0$ and $\sigma$ is a parameter related to the noise term. Substituting this into the running-time bound gives an overall complexity of

$$\widetilde{O}\left(\frac{n \cdot 3^{2k-1} \cdot \sigma}{q}\right).$$

The complexity is thus exponential in $k$ but only *polynomial in* $n$—precisely the feature that makes very small $k$ vulnerable, even at large dimensions. The threshold $q \approx 3^k$ marks a natural boundary between the two regimes, and the framework covers both sides.

**How the attack works.** The statistical attack directly exploits the fact that, although the secret $s$ has $n$ coordinates, each sparse equation involves only $k$ of them. For any target coordinate $s_i$ and candidate value $u \in \{-1, 0, 1\}$, one enumerates all $3^{k-1}$ ternary assignments to the *other* $k-1$ secret variables in that row, computes a marginal likelihood for the hypothesis $s_i = u$, and aggregates this evidence across all rows containing $s_i$. The algorithm then iteratively fixes the coordinate with the highest aggregated posterior and eliminates it from every row in which it appears. Crucially, each elimination shrinks the affected rows by one, so subsequent marginalizations become progressively cheaper, keeping the total work within the stated bound. Unlike the combinatorial approach of [KSM24], which requires tuning multiple internal trade-off parameters, our method admits a transparent complexity analysis directly in terms of $k$.

**Experimental validation.** Experiments on toy instances confirm the predicted sharp transition governed by the sample condition above. A direct comparison with [KSM24] demonstrates lower complexity across a broad range of their parameter sets. In particular, our results identify concrete regimes in which efficiency-motivated sparsity directly compromises security, providing actionable guidance for parameter selection in sparse LWE-based schemes.

**Extension beyond the ternary case.** Although we focus on the ternary setting for clarity, the same sparsity-driven phenomenon extends to more general *small-secret* distributions. In particular, if each secret coordinate lies in a bounded interval $s_i \in [-\|s\|_\infty, \|s\|_\infty]$, then a single sparse equation involves only $k$ coordinates and the per-row candidate space scales as $(2\|s\|_\infty+1)^k$ rather than $3^k$. Accordingly, the natural regime boundary $q \approx 3^k$ in the ternary case generalizes to $q \approx (2\|s\|_\infty + 1)^k$.

### 1.2 Related work

**Sparse LWE and parameter heuristics.** [JLS24] introduced Sparse LWE, where the public coefficient matrix is sampled to be $k$-sparse, and provided a systematic discussion of concrete security considerations and heuristic parameter-selection guidance in light of known attacks. A key message of [JLS24] is that sparsity can introduce phenomena that are absent or much weaker for dense LWE, such as support collisions and reductions to lower-dimensional sub-instances. Most importantly for our purposes, [JLS24] cautions that *small-secret* spLWE is not automatically equivalent to its large-secret counterpart (unlike in standard LWE), leaving the concrete hardness of small-secret spLWE as a major open problem and motivating further cryptanalysis in regimes where $k$ is very small.

**Combinatorial cryptanalysis of ternary spLWE.** [KSM24] initiated third-party cryptanalysis of the ternary variant of spLWE, where the secret is supported on $\{-1, 0, 1\}$. Their approach is combinatorial in nature, combining subsystem extraction with meet-in-the-middle techniques, and it highlights a threshold phenomenon governed by the comparison between $3^k$ and the modulus $q$. When $q > 3^k$, each sparse row individually constrains the secret to a low-dimensional sublattice, and the resulting short-vector problem can be solved via lattice reduction; in particular, the HKZ algorithm [HS07, Kan83] applies to the induced subset-sum instance and achieves complexity $2^{0.292 \cdot k}$. When $q < 3^k$, a single row no longer suffices and one must aggregate information across multiple equations. In the latter aggregation regime, combinatorial attacks tend to involve several internal trade-offs—subsystem sizes, splitting rules, and time–memory constraints—which makes it challenging to summarize the overall running time as a single, transparent asymptotic expression with explicit dependence on $k$. Our work targets precisely this aggregation regime: we seek an attack whose complexity exposes a sharp dependence on $k$ without relying on such trade-off parameters.

**General attacks on (sparse) small secret LWE.** Concrete security for LWE-based schemes is typically assessed against the best known algorithmic families, and additional structure—such as small secrets, sparsity, or other algebraic constraints—can substantially alter the attack landscape. Primal lattice attacks based on Kannan's embedding and BKZ-type reduction, together with hybridizations that incorporate meet-in-the-middle and representation techniques for small or sparse secrets, form one major family (see, e.g., [HG07, ACW19,

May21,CHHS19,HKLS22,BLLW22,NMW+24,WSM+25,KKN+25]). BKW-style combinatorial algorithms and their extensions [BKW03,ACF+15] trade samples for dimension reduction and have been adapted to various structured settings. A further line of attack consists of algebraic methods in the Arora–Ge framework [AG11]. These families are typically compared via cost-model based estimators and parameter-selection frameworks (see, e.g., [APS15, Pla18]). None of these general approaches, however, yields attack costs that depend explicitly and sharply on the sparsity parameter $k$; it is precisely this gap that our $k$-dependent analysis fills.

**Sparse LPN.** Sparse variants of the Learning Parity with Noise (LPN) problem over $\mathbb{F}_2$ were introduced by Alekhnovich [Ale03], where each sample vector has Hamming weight exactly $k$ (typically a small constant). Sparse LPN has been leveraged as a hardness assumption for various cryptographic constructions, including public-key encryption [ABW10], multi-party homomorphic secret sharing [DIJL23], and lossy trapdoor functions [DJ24].

Although sparse LPN and sparse LWE share the property that each sample involves only $k$ of the $n$ secret coordinates, the statistical structure differs fundamentally. Over $\mathbb{F}_2$, for any sparse row with at least one active coefficient, the set of residuals $\{b - \langle \mathbf{a}, \mathbf{x} \rangle \bmod 2 : \mathbf{x} \in \mathbb{F}_2^k\}$ obtained by enumerating all $2^k$ candidate assignments is perfectly balanced: exactly $2^{k-1}$ assignments map to 0 and $2^{k-1}$ map to 1. Consequently, there is no distinguished small-residual spike that distinguishes the correct assignment from wrong ones. In contrast, over $\mathbb{Z}_q$ with Gaussian error, exactly one assignment (the true secret restriction) produces a residual distributed as $e \sim D_{\mathbb{Z},\sigma}$ and concentrated near zero, while the other $3^k - 1$ wrong assignments yield residuals that behave close to uniform over $\mathbb{Z}_q$. This asymmetry is exploited by greedy coordinate-recovery methods in the spLWE setting, but it vanishes for sparse LPN over $\mathbb{F}_2$. We provide a more detailed explanation of this distinction at the end of Section 3.

## 2 Preliminaries

### 2.1 Notation

For an integer $n$, let $[n] := \{1, \ldots, n\}$. We use bold lowercase letters (e.g., $\mathbf{v}$) for vectors and bold uppercase letters (e.g., $\mathbf{A}$) for matrices. For a distribution $X$, we write $x \leftarrow X$ for sampling from $X$, and $x \xleftarrow{\$} \Omega$ for uniform sampling from a finite set $\Omega$. A function $\varepsilon(\lambda)$ is negligible if $\varepsilon(\lambda) = o(\lambda^{-c})$ for every constant $c > 0$. Throughout, for $a \in \mathbb{Z}$ we write $a \bmod q \in \left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$ for the centered representative of $a$ modulo $q$, and apply this convention component-wise to vectors.

For a vector $\mathbf{s} = (s_1, \ldots, s_n)$ and an index set $S_t \subseteq [n]$, we write $\mathbf{s}|_{S_t}$ for the restriction (subvector) of $\mathbf{s}$ to the coordinates indexed by $S_t$. Formally, if $S_t = \{i_1 < \cdots < i_\ell\}$, then

$$\mathbf{s}|_{S_t} := (s_{i_1}, \ldots, s_{i_\ell}) \in \mathbb{Z}^\ell.$$

### 2.2 Lattice background

A (full-rank) lattice $\Lambda \subseteq \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$. Equivalently, $\Lambda$ is the set of all integer combinations of a basis $\mathbf{B} \in \mathbb{R}^{n \times n}$:

$$\Lambda = \mathcal{L}(\mathbf{B}) := \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^n\}.$$

The determinant (covolume) of $\Lambda$ is $\det(\Lambda) := |\det(\mathbf{B})|$, which is independent of the choice of basis. We use the Euclidean norm $\|\cdot\|$ and inner product $\langle \cdot, \cdot \rangle$.

**Discrete Gaussian distribution on lattices.** Let $\Lambda \subseteq \mathbb{R}^d$ be a lattice and $\sigma > 0$. Define $\rho_\sigma(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/\sigma^2)$ and $\rho_\sigma(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_\sigma(\mathbf{x})$. The (centered) discrete Gaussian $D_{\Lambda,\sigma}$ is given by

$$\Pr_{X \leftarrow D_{\Lambda,\sigma}}[X = \mathbf{x}] = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\Lambda)} \qquad (\mathbf{x} \in \Lambda).$$

We use the one-dimensional distribution $D_{\mathbb{Z},\sigma}$ as the error distribution. A standard tail bound implies that $\|X\| = O(\sigma\sqrt{d})$ with overwhelming probability [GPV08].

### 2.3 LWE problem and its sparse variant

We recall the LWE distribution and the decisional LWE problem, and then introduce the sparse-row variant studied in this work.

**Definition 2.1 (LWE distribution)** *Let $n, q \geq 1$ and let $\chi$ be a distribution over $\mathbb{Z}$. For a secret $\mathbf{s} \in \mathbb{Z}_q^n$, define the distribution $\mathcal{D}_{\mathbf{s}}^{n,q,\chi}$ over pairs $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$, $e \leftarrow \chi$, and setting $b \equiv \langle \mathbf{a}, \mathbf{s} \rangle + e \pmod q$.*

**Definition 2.2 (Decisional LWE)** *The decisional* $\mathrm{LWE}(n, m, q, \chi)$ *problem is to distinguish, given $(\mathbf{A}, \mathbf{b})$, between the following two distributions:*

$$(\mathbf{A}, \mathbf{b}) \leftarrow \mathcal{D}_{\mathbf{s}}^{n,m,q,\chi} \quad \text{for } \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \qquad \text{and} \qquad (\mathbf{A}, \mathbf{u}) \text{ with } \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n},\ \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$$

*where $\mathcal{D}_{\mathbf{s}}^{n,m,q,\chi}$ denotes the joint distribution of $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ generated as follows: sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ uniformly and $\mathbf{e} \xleftarrow{\$} \chi^m$, and set $\mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q$.*

**Definition 2.3 (Search sparse LWE with ternary secret)** *Fix parameters $\sigma > 0$ and $(n, m, q, k) \in \mathbb{Z}_{\geq 1}^4$. Let $\mathbf{s} \in \{-1, 0, 1\}^n$ be an unknown secret. A* sparse LWE *instance consists of $m$ samples $(\mathbf{a}_t, b_t) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for $t \in [m]$ such that*

$$b_t \equiv \langle \mathbf{a}_t, \mathbf{s} \rangle + e_t \pmod q,$$

*where the errors are independent $e_t \leftarrow D_{\mathbb{Z},\sigma}$ and each $\mathbf{a}_t$ has exactly $k$ nonzero entries. The supports are uniform random $k$-subsets and nonzero coefficients are uniform in $\mathbb{Z}_q^*$. The* search sparse LWE *problem, denoted* $\mathrm{spLWE}(n, m, q, k, \sigma)$*, is to recover $\mathbf{s}$ given $(\mathbf{a}_t, b_t)_{t \in [m]}$.*

### 2.4 Useful lemma

We state a basic concentration bound that will be used in the proof of the main heuristic success theorem.

**Lemma 2.4 (Hoeffding bound for bounded sums)** *Let $Y_1, \ldots, Y_N$ be independent random variables with $0 \le Y_r \le 1$ almost surely, and let $S := \sum_{r=1}^{N} Y_r$ and $\mu := \mathbb{E}[S]$. Then*

$$\Pr\left[|S - \mu| \ge \frac{\mu}{2}\right] \le 2\exp\left(-\frac{\mu^2}{2N}\right).$$

## 3 Cryptanalysis of spLWE in the Statistical Regime

In this section, we are given spLWE samples $\{(\mathbf{a}_t, b_t)\}_{t\in[m]}$ drawn from the sparse LWE distribution in Definition 2.3, and we present a $k$-dependent cryptanalysis based on greedy coordinate recovery. While this section assumes $q \le 3^k$, the case for $q > 3^k$ is addressed separately in Section 4.

The remainder of this section is organized as follows. In Section 3.1, we define the row-marginal score and present the greedy elimination algorithm. In Section 3.2, we analyze the cost of a single attack run and make the dependence on the sparsity parameter $k$ explicit. In Section 3.3, we give a heuristic success-probability analysis based on a mean/variance calculation under an approximate independence assumption.

**High-level description.** At a high level, the attack exploits the sparsity of the equations: each row involves only $k \ll n$ secret coordinates. In particular, the number of contributing indices per equation is $k$, not $n$, which makes it feasible to exhaustively enumerate all assignments to the $k$ coordinates appearing in a given row.

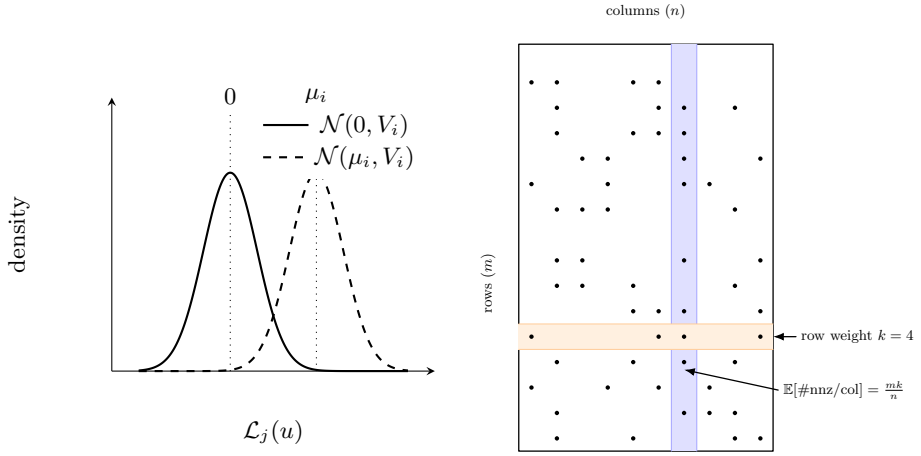
For a given row $t$ and a tentative assignment $\mathbf{x} \in \{-1, 0, 1\}^{S_t}$ to the variables indexed by $S_t$, we form the residual

$$r_t(\mathbf{x}) := \Big(b_t - \sum_{j\in S_t} a_{t,j} x_j\Big) \bmod q \;\in\; \mathbb{Z}_q.$$

If $\mathbf{x}$ matches the true restriction $\mathbf{s}|_{S_t}$, then $r_t(\mathbf{x}) = e_t$ follows the error distribution and is concentrated near 0; whereas for an incorrect assignment, the residual typically behaves close to uniform over $\mathbb{Z}_q$. This dichotomy motivates a likelihood-shaped weight that favors small residuals.

Using this weight, for each coordinate $s_i$ and candidate value $u \in \{-1, 0, 1\}$, we compute a *row-marginal likelihood* by summing over the remaining $k-1$ variables within each incident row, and then aggregate this evidence across all rows containing $s_i$. The algorithm iteratively fixes the coordinate with the most decisive score and eliminates it from all equations.

A schematic view of the row-$k$ sparsity pattern and of the resulting score distributions used in the analysis is given in Figure 1.




(a) Wrong vs. correct scores.

(b) $m \times n$ row-$k$-sparse matrix.

Fig. 1: Two schematic views used in the analysis: (a) score distributions in the greedy step, and (b) a row-$k$-sparse public matrix with expected column load $mk/n$.

### 3.1 Attack description

**Goal and outline.** We are given ternary sparse LWE samples $\{(\mathbf{a}_t, b_t)\}_{t\in[m]}$ drawn from Definition 2.3, with an unknown ternary secret $\mathbf{s} \in \{-1, 0, 1\}^n$ and discrete-Gaussian errors. Our goal is to recover $\mathbf{s}$. The attack proceeds in two conceptual steps:

1. **Row-level distinguisher** For a fixed coordinate $i$ and candidate value $u \in \{-1, 0, 1\}$, each sparse row $t$ containing $i$ yields a score measuring how consistent that row is with the hypothesis $s_i = u$, after marginalizing over the other (at most $k - 1$) unknown variables in that row.
2. **Greedy recovery** We aggregate the row-level scores into a global posterior $P_i(u)$, select the most confident coordinate-value pair, fix it, eliminate it from all affected equations, and iterate.

**Notation.** For each row $t \in [m]$, let $S_t := \mathrm{supp}(\mathbf{a}_t) \subseteq [n]$ denote the support of $\mathbf{a}_t$, so $|S_t| = k$. For each coordinate $i \in [n]$, let $T_i := \{t \in [m] : i \in S_t\}$ be the set of rows in which $s_i$ appears. During greedy elimination the sets $S_t$ shrink as coordinates are fixed, and the corresponding $T_i$ update accordingly.

**Weight function and row-marginal weights.** We define the (unnormalized) weight function

$$w(r) := \exp\left(-\pi r^2/\sigma^2\right),$$

where $r \in \mathbb{Z}_q$ is interpreted as its centered representative in $\left[-(q-1)/2, (q-1)/2\right]$.

Fix a row $t$, a coordinate $i \in S_t$, and a candidate value $u \in \{-1, 0, 1\}$ for $s_i$. We define the *row-marginal weight*

$$W_t^{(i)}(u) := \sum_{\mathbf{x}:\ S_t \setminus \{i\} \to \{-1,0,1\}} w\left(\left(b_t - a_{t,i}u - \sum_{j \in S_t \setminus \{i\}} a_{t,j}\mathbf{x}(j)\right) \bmod q\right). \quad (1)$$

Because each row contains at most $k$ variables, (1) can be evaluated by enumerating $3^{|S_t|-1} \le 3^{k-1}$ assignments.

**Local and global posteriors.** For each row $t \in T_i$, we normalize the three values $W_t^{(i)}(u)$ into a *local posterior*

$$P_t^{(i)}(u) := \frac{W_t^{(i)}(u)}{\sum_{v \in \{-1,0,1\}} W_t^{(i)}(v)} \qquad (u \in \{-1, 0, 1\}).$$

Heuristically treating the contributions of different rows in $T_i$ as approximately independent, we aggregate by multiplication and obtain the *global posterior*

$$P_i(u) := \frac{\prod_{t \in T_i} P_t^{(i)}(u)}{\sum_{v \in \{-1,0,1\}} \prod_{t \in T_i} P_t^{(i)}(v)}. \quad (2)$$

For numerical stability we work with log-scores

$$\mathcal{L}_i(u) := \sum_{t \in T_i} \log P_t^{(i)}(u), \qquad P_i(u) = \frac{\exp(\mathcal{L}_i(u))}{\sum_v \exp(\mathcal{L}_i(v))}.$$

Maximizing $P_i(u)$ is equivalent to maximizing $\mathcal{L}_i(u)$.

**Greedy coordinate recovery and elimination.** The algorithm maintains a set $\mathcal{I}$ of unfixed coordinates and a partial assignment $\hat{\mathbf{s}}$. At each iteration, it selects the most confident coordinate-value pair

$$(i^\star, u^\star) := \arg \max_{i \in \mathcal{I}, u \in \{-1,0,1\}} P_i(u),$$

sets $\hat{s}_{i^\star} = u^\star$, and eliminates $i^\star$ from all rows in $T_{i^\star}$ by updating

$$b_t \leftarrow b_t - a_{t,i^\star}\hat{s}_{i^\star} \pmod q, \qquad S_t \leftarrow S_t \setminus \{i^\star\}.$$

Only the rows $t \in T_{i^\star}$ change, so only coordinates still present in those rows require score updates. Algorithm 1 presents the full procedure and caching strategy used to update $\mathcal{L}_i(u)$ efficiently.

### 3.2 Complexity analysis

We analyze the running time of Algorithm 1. Throughout, we focus on the dominant term and treat the space cost as negligible compared to time.

**Algorithm 1** GreedyspLWE($\{(\mathbf{a}_t, b_t)\}_{t\in[m]}; q, \sigma$)

$\hat{\mathbf{s}} \leftarrow \mathbf{0} \in \mathbb{Z}^n$, $\quad \mathcal{I} \leftarrow [n]$
$S_t \leftarrow \mathrm{supp}(\mathbf{a}_t)$ for all $t \in [m]$
Compute $\log P_t^{(i)}(s_i = u)$ for all $t \in [m]$, $i \in S_t$, $u \in \{-1, 0, 1\}$
$\mathcal{L}_i(u) \leftarrow \sum_{t\in T_i} \log P_t^{(i)}(s_i = u)$ for all $i \in [n]$, $u \in \{-1, 0, 1\}$
**while** $\mathcal{I} \neq \emptyset$ **do**
    Compute $P_i(u) = \frac{\exp(\mathcal{L}_i(u))}{\sum_v \exp(\mathcal{L}_i(v))}$ from $\mathcal{L}_i(u)$ for all $i \in \mathcal{I}$
    $(i^\star, u^\star) \leftarrow \arg\max_{i\in\mathcal{I},u} P_i(u)$
    $\hat{s}_{i^\star} \leftarrow u^\star$
    **for** $t \in T_{i^\star}$ **do**
        $b_t \leftarrow b_t - a_{t,i^\star}\hat{s}_{i^\star} \pmod q$
        $S_t \leftarrow S_t \setminus \{i^\star\}$
        Recompute $\log P_t^{(j)}(s_j = u)$ for all $j \in S_t$ and $L_j(u)$, $u \in \{-1, 0, 1\}$
    **end for**
    $\mathcal{I} \leftarrow \mathcal{I} \setminus \{i^\star\}$
**end while**
**return** $\hat{\mathbf{s}}$

**Cost of updating one row.** Fix a row $t$ whose current support is $S_t$ with $\ell := |S_t| \le k$. When this row is updated, we need to recompute its local contributions to the scores of the remaining coordinates $j \in S_t$. A direct implementation can enumerate all ternary assignments $\mathbf{x} \in \{-1, 0, 1\}^{S_t}$ (there are $3^\ell$ assignments), compute the residual

$$r_t(\mathbf{x}) = \Big(b_t - \sum_{j\in S_t} a_{t,j}x_j\Big) \bmod q,$$

evaluate the weight $w(r_t(\mathbf{x}))$, and accumulate it into the appropriate buckets. Importantly, *one enumeration pass* can update all coordinates in the row at once: for each assignment $\mathbf{x}$ we add $w(r_t(\mathbf{x}))$ to the bucket corresponding to $(i, x_i)$ for every $i \in S_t$. Hence, the cost of recomputing all local row-marginals for row $t$ at support size $\ell$ is

$$O(\ell \cdot 3^\ell).$$

**Amortized bound over the greedy process.** During greedy elimination, each time a coordinate contained in row $t$ is fixed, the row support shrinks by one. Therefore each row $t$ is updated exactly $k$ times, with support sizes $\ell = k, k-1, \ldots, 1$. The total work spent on a single row is thus bounded by

$$\sum_{\ell=1}^{k} O(\ell \cdot 3^\ell) = O\left(\sum_{\ell=1}^{k} \ell \cdot 3^\ell\right) = O\left(\frac{(2k-1)3^{k+1}+3}{4}\right) = O(k \cdot 3^k),$$

since the geometric growth implies the sum is dominated by its last term.

**Total time complexity.** Summing over all $m$-rows, we state the overall running time of Algorithm 1.

**Lemma 3.1** *The Algorithm 1 terminates in total time $O(m \cdot k \cdot 3^k)$.*

All remaining operations (maintaining $\mathcal{I}$, bookkeeping of supports, and selecting the maximizer of $P_i(u)$) can be implemented in time $o(mk3^k)$ overall (e.g., via a priority queue of current confidences), and therefore do not affect the asymptotic bound in Lemma 3.1.

**Space complexity (ignored).** The algorithm stores the supports $S_t$ (and/or incidence lists) and cached local scores for pairs $(t, i)$ with $i \in S_t$, which requires $O(mk)$ words, plus an $O(q)$ lookup table for the weight function if desired. Since our focus is concrete running time, we treat the overall space usage as secondary and omit it from the main cost bound.

### 3.3 Heuristic success probability for GreedyspLWE

We give a closed-form heuristic estimate for the success probability of Algorithm 1 under an extreme-value model of the aggregated scores. Throughout, the randomness is over the spLWE instance generation (supports, coefficients, and errors).

**Target event** At iteration $i \in \{1, \ldots, n\}$, let $\mathcal{I}_i \subseteq [n]$ be the set of unfixed coordinates and $N_i := |\mathcal{I}_i| = n - i + 1$. For each $j \in \mathcal{I}_i$, the algorithm computes the aggregated posteriors $P_j(u)$ for $u \in \{-1, 0, 1\}$ (equivalently log-scores $\mathcal{L}_j(u)$), and selects

$$(j^\star, u^\star) := \arg \max_{j \in \mathcal{I}_i,\ u \in \{-1,0,1\}} \mathcal{L}_j(u).$$

We define the iteration-$i$ success event

$$E_i := \{u^\star = s_{j^\star}\}, \qquad p_i := \Pr[E_i]. \tag{3}$$

If $E_i$ holds for all $i = 1, \ldots, n$, then Algorithm 1 recovers the full secret:

$$\{\hat{\mathbf{s}} = \mathbf{s}\} \supseteq \bigcap_{i=1}^{n} E_i. \tag{4}$$

Accordingly, we estimate the overall recovery probability by the product heuristic

$$P := \Pr[\hat{\mathbf{s}} = \mathbf{s}] \approx \prod_{i=1}^{n} p_i, \tag{5}$$

treating the iteration decisions as weakly dependent.

**Heuristic assumptions and mean-field modeling.** Computing the exact success probability of Algorithm 1 is difficult, since the aggregated scores $\{\mathcal{L}_j(u)\}$ are complicated dependent functions of the random supports, the noise, and the row-wise normalizations.

We therefore model the greedy step by a sequence of heuristic assumptions and mean-field approximations that capture the dominant effect while discarding higher-order dependencies. Under this model, we will estimate the per-iteration success probabilities $p_i = \Pr[E_i]$ and the overall recovery probability via the product heuristic $P \approx \prod_{i=1}^{n} p_i$.

**Score-level heuristic assumptions.** Our analysis rests on four simplifying hypotheses about the row-marginal scores.

**Assumption 1 (Uniform residual for wrong assignments)** *Fix a row $t$ and an assignment $\mathbf{x} \in \{-1, 0, 1\}^{S_t}$. If $\mathbf{x} \neq \mathbf{s}|_{S_t}$, then the residual*

$$r_t(\mathbf{x}) := \Big(b_t - \sum_{j \in S_t} a_{t,j} x_j\Big) \bmod q \ \in \ \left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$$

*behaves as* $\mathrm{Unif}(\mathbb{Z}_q)$ *under the centered representative convention.*

**Assumption 2 (Row-wise independence)** *For a fixed coordinate $j$ and hypothesis $u \in \{-1, 0, 1\}$, the row-wise log-contributions $\{\log P_t^{(j)}(u)\}_{t \in T_j}$ are treated as independent across distinct rows $t$.*

**Assumption 3 (Gaussian modeling for aggregated *gaps*)** *At iteration $i$, for each remaining coordinate $j \in \mathcal{I}_i$ and each wrong value $u \in \{-1, 0, 1\} \setminus \{s_j\}$, define the aggregated log-gap*

$$G_{j,u} := \mathcal{L}_j(s_j) - \mathcal{L}_j(u).$$

*The $2N_i$ random variables $\{G_{j,u}\}_{j \in \mathcal{I}_i,\ u \neq s_j}$ are modeled as independent Gaussians with common mean $\mu_i$ and common variance $V_i$:*

$$G_{j,u} \sim \mathcal{N}(\mu_i, V_i).$$

**Assumption 4 (Continuous approximation for discrete Gaussians)** *We approximate $e \leftarrow D_{\mathbb{Z},\sigma}$ by the continuous density $f(x) = \frac{1}{\sigma}\exp(-\pi x^2/\sigma^2)$ when evaluating moments of $w(e)$.*

*For $U \xleftarrow{\$} \mathbb{Z}_q$ (centered representative in $[-(q-1)/2, (q-1)/2]$), we approximate*

$$\mathbb{E}[w(U)] \approx \frac{1}{q}\int_{-q/2}^{q/2} \exp(-\pi x^2/\sigma^2)dx = \frac{\sigma}{q}\mathrm{erf}\left(\frac{\sqrt{\pi}}{2}\cdot\frac{q}{\sigma}\right).$$

*In our parameter regime $q \gg \sigma$, we further simplify*

$$\mathrm{erf}\left(\frac{\sqrt{\pi}}{2}\cdot\frac{q}{\sigma}\right) \approx 1, \qquad \textit{hence} \qquad \mathbb{E}[w(U)] \approx \frac{\sigma}{q}.$$

**Mean-field modeling of the greedy process.** Assumptions 1–4 specify a probabilistic model for the score statistics. To express the resulting success estimate in closed form as a function of $(n, m, k, q, \sigma)$, we further adopt a mean-field description of how the greedy elimination changes the instance over time.

**Assumption 5 (Typical row support size)** *At iteration $i \in \{1, \ldots, n\}$, we model the (typical) support size of an updated row by*

$$\ell_i := k\Big(1 - \frac{i-1}{n}\Big). \tag{6}$$

*Equivalently, we assume that the greedy process removes roughly the same fraction $(i-1)/n$ of coordinates from a typical row by iteration $i$. This mean-field schedule ignores fluctuations across rows and the bias introduced by selecting coordinates with high confidence; it is used only to obtain a closed-form expression.*

**Assumption 6 (Typical incidence count)** *We model the number of rows containing a typical coordinate by*

$$T := \frac{mk}{n}. \tag{7}$$

*Indeed, under the random-support model where each $S_t$ is a uniformly random $k$-subset of $[n]$, for any fixed coordinate $j \in [n]$ we have*

$$\Pr[j \in S_t] = \frac{k}{n}, \qquad |T_j| = \sum_{t=1}^{m} \mathbf{1}_{\{j \in S_t\}}, \qquad \mathbb{E}[|T_j|] = \sum_{t=1}^{m} \Pr[j \in S_t] = m \cdot \frac{k}{n}.$$

*We treat $|T_j|$ as concentrated around its mean and ignore coordinate-to-coordinate variation and dependencies created during the elimination process.*

**Assumption 7 (Mean-field log-gap replacement)** *Fix a level $\ell$ and write $N := 3^{\ell-1}$. Let $S := \sum_{r=1}^{N-1} Y_r$ and $m_S := \mathbb{E}[S] = (N-1)\mu_Y$. We model the row-level log-gap by replacing the random baseline $S$ with its mean $m_S$:*

$$\mathbb{E}[\log(S + X) - \log(S + Y_0)] = \mathbb{E}[\log(m_S + X) - \log(m_S + Y_0)]. \tag{8}$$

This mean-field closure ignores row-wise fluctuations of $S$, selection-induced bias across rows, and is used only to obtain a closed-form scaling law in our analysis.

**Roadmap.** In the next step, under Assumptions 1–7, we will (heuristically) compute the score parameters $(\mu_i, V_i)$ and derive an explicit extreme-value expression for $p_i$, which will be summarized as a closed-form success condition in Theorem 3.2.

**Theorem 3.2** *Assume the score-model Assumptions 1–4 and the mean-field Assumptions 5–7. Recall $\ell_i = k\Big(1 - \frac{i-1}{n}\Big)$ and $N_i = n - i + 1$, and set $T := mk/n$. Then there exists a positive constant $C$ such that Algorithm 1 solves* spLWE$(n, m, q, k, \sigma)$ *if, for all $i \in [n]$,*

$$\frac{m \cdot k \cdot q}{n \cdot 3^{\ell_i - 1}\sigma} \geq C \cdot \log\big(3N_i\big). \tag{9}$$

*Proof.* Fix an iteration $i$ and write $N_i = n - i + 1$. Under Assumptions 5–6, we take the typical current row support size to be $\ell_i$ and the typical incidence to be $|T_j| = T$ for each $j \in \mathcal{I}_i$.

**High-level idea (from score maximization to per-row log-gaps).** At iteration $i$, the algorithm selects

$$(j^\star, u^\star) = \arg \max_{j \in \mathcal{I}_i,\ u \in \{-1,0,1\}} \mathcal{L}_j(u), \qquad \mathcal{L}_j(u) = \sum_{t \in T_j} \log P_t^{(j)}(u).$$

For any fixed coordinate $j$, the value decision is correct if and only if the correct hypothesis dominates the two wrong ones:

$$\arg \max_{u \in \{-1,0,1\}} \mathcal{L}_j(u) = s_j \iff \mathcal{L}_j(s_j) > \mathcal{L}_j(u) \quad \text{for all } u \neq s_j.$$

Therefore, to quantify how reliably the correct value dominates the wrong values, it is natural to study the (random) log-gaps $\mathcal{L}_j(s_j) - \mathcal{L}_j(u)$ for $u \neq s_j$. Since $\mathcal{L}_j(\cdot)$ aggregates row-wise evidence, each such gap decomposes as a sum of *per-row log-gaps*. This motivates introducing

$$\Delta_{t,j}(u) := \log W_t^{(j)}(s_j) - \log W_t^{(j)}(u), \qquad (u \neq s_j),$$

and analyzing how these contributions accumulate across the incident rows $t \in T_j$. This reduction turns the score comparison into a mean/variance calculation, which can then be plugged into the extreme-value model of Assumption 3. For intuition on the score separation captured by this gap viewpoint, see Figure 1(a).

**Step 1: Row-level log-gap and moment notation.** Fix a row $t$ with current support size $\ell := |S_t|$ and a coordinate $j \in S_t$. For a wrong value $u \in \{-1, 0, 1\} \setminus \{s_j\}$ define the row-level log-gap

$$\Delta_{t,j}(u) := \log W_t^{(j)}(s_j) - \log W_t^{(j)}(u).$$

Let $N := 3^{\ell-1}$. By Assumption 1, for $u = s_j$ there is exactly one marginalized assignment whose residual equals $e_t$, whereas for $u \neq s_j$ all $N$ assignments behave like uniform residuals. Accordingly, we model

$$W^{\text{corr}} = X + \sum_{r=1}^{N-1} Y_r, \qquad W^{\text{wrong}} = \sum_{r=1}^{N} Y_r,$$

where $X := w(e)$ with $e \leftarrow D_{\mathbb{Z},\sigma}$ and $Y_r := w(U_r)$ with $U_r \xleftarrow{\$} \mathbb{Z}_q$ (centered representatives), and all variables are independent. Write $\mu_X := \mathbb{E}[X]$, $\nu_X := \mathrm{V}ar(X)$ and $\mu_Y := \mathbb{E}[Y]$, $\nu_Y := \mathrm{V}ar(Y)$.

For a fixed support size $\ell$, the distribution of $\Delta_{t,j}(u)$ (for any wrong $u \neq s_j$) is modeled as depending only on $\ell$ under the random-support model. Accordingly, we introduce a generic row-level log-gap random variable $\Delta_\ell$ by

$$\Delta_\ell \overset{d}{=} \Delta_{t,j}(u) \quad \text{for any fixed } (t,j) \text{ with } |S_t| = \ell \text{ and any fixed wrong } u \neq s_j.$$

We write its mean and variance as

$$\mu_\ell := \mathbb{E}[\Delta_\ell], \qquad v_\ell := \mathrm{V}ar(\Delta_\ell).$$

**Step 2: Continuous moment evaluation under Assumption 4.** By Assumption 4, we evaluate moments of $X = w(e)$ by replacing $e \leftarrow D_{\mathbb{Z},\sigma}$ with the continuous density $f(x) = \frac{1}{\sigma} e^{-\pi x^2/\sigma^2}$. Thus

$$\mu_X := \mathbb{E}[X] = \int_{\mathbb{R}} w(x) f(x) dx = \int_{\mathbb{R}} e^{-\pi x^2/\sigma^2} \cdot \frac{1}{\sigma} e^{-\pi x^2/\sigma^2} dx = \frac{1}{\sqrt{2}}.$$

By Assumption 4 again, for $U \xleftarrow{\$} \mathbb{Z}_q$ we set

$$\mu_Y := \mathbb{E}[Y] = \frac{1}{q} \int_{-q/2}^{q/2} e^{-\pi x^2/\sigma^2} dx = \frac{\sigma}{q} \mathrm{erf}\left( \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sigma} \right),$$

and in the regime $q \gg \sigma$ we further take

$$\mathrm{erf}\left( \frac{\sqrt{\pi}}{2} \cdot \frac{q}{\sigma} \right) = 1, \qquad \text{so that} \qquad \mu_Y = \frac{\sigma}{q}.$$

Similarly, we set

$$\mathbb{E}[Y^2] \le \frac{1}{q} \int_{\mathbb{R}} e^{-2\pi x^2/\sigma^2} dx = \frac{\sigma}{q\sqrt{2}}, \qquad \nu_Y := \mathrm{V}ar(Y) \le \mathbb{E}[Y^2] = \frac{\sigma}{q\sqrt{2}}.$$

**Step 3: A variance upper bound for the row-level log-gap.** Let $S := \sum_{r=1}^{N-1} Y_r$, so that $W^{\mathrm{wrong}} = S + Y_0$ and $W^{\mathrm{corr}} = S + X$ with $Y_0 \overset{d}{=} Y$. Write

$$\Delta_\ell = \log(S + X) - \log(S + Y_0).$$

We first use the general bound (valid for all square-integrable $A, B$)

$$\mathrm{V}ar(A - B) \le 2\mathrm{V}ar(A) + 2\mathrm{V}ar(B), \tag{10}$$

which avoids tracking the covariance between $\log(S+X)$ and $\log(S+Y_0)$ induced by the shared sum $S$. Applying (10) gives

$$v_\ell := \mathrm{V}ar(\Delta_\ell) \le 2\mathrm{V}ar(\log(S + X)) + 2\mathrm{V}ar(\log(S + Y_0)).$$

We now bound $\mathrm{V}ar(\log Z)$ for a positive random variable $Z$ in terms of $\mathrm{V}ar(Z)$ and $\mathbb{E}[Z]$. Let $m := \mathbb{E}[Z]$ and define the event $\mathcal{A} := \{|Z - m| \le m/2\}$, on which $Z \ge m/2$. Since $(\log x)' = 1/x \le 2/m$ for all $x \ge m/2$, the mean value theorem implies that $\log(\cdot)$ is $(2/m)$-Lipschitz on $[m/2, \infty)$, hence on $\mathcal{A}$,

$$|\log Z - \log m| \le \frac{2}{m} |Z - m|.$$

Therefore,

$$\mathbb{E}\big[(\log Z - \log m)^2 \mathbf{1}_{\mathcal{A}}\big] \le \frac{4}{m^2}\mathbb{E}[(Z-m)^2] = \frac{4}{m^2}\mathrm{V}ar(Z).$$

For the complementary event $\mathcal{A}^c$, we use a quantitative tail bound. Recall that $Z$ is a sum of independent random variables taking values in $[0,1]$: either $Z = S + Y_0 = \sum_{r=1}^{N} Y_r$ or $Z = S + X = \sum_{r=1}^{N-1} Y_r + X$, with $0 \le Y_r \le 1$ and $0 \le X \le 1$. Let $m := \mathbb{E}[Z]$ and $\mathcal{A} := \{|Z - m| \le m/2\}$. Applying Lemma 2.4 with $t = m/2$ gives

$$\Pr[\mathcal{A}^c] = \Pr\left[|Z-m| > \frac{m}{2}\right] \le 2\exp\left(-\frac{m^2}{2N}\right). \tag{11}$$

Moreover, since $0 < Z \le N$ and $m \ge N\mu_Y$ in both cases, we have on $\mathcal{A}^c$ the crude bound $|\log Z - \log m| \le \log N + |\log m| \le 2\log N$ for all $N \ge 2$ (absorbing the dependence on $m$ into the same bound). Hence

$$\mathbb{E}\big[(\log Z - \log m)^2 \mathbf{1}_{\mathcal{A}^c}\big] \le 4(\log N)^2 \Pr[\mathcal{A}^c] \le 8(\log N)^2 \exp\left(-\frac{m^2}{2N}\right). \tag{12}$$

In particular, whenever $m^2/N$ is bounded below by an absolute constant (which is the relevant regime in our parameter ranges), the right-hand side of (12) is bounded by an absolute constant.

On the event $\mathcal{A}$ we have $Z \ge m/2$, so $\log(\cdot)$ is $(2/m)$-Lipschitz on $[m/2, \infty)$, and thus

$$|\log Z - \log m| \le \frac{2}{m}|Z - m| \qquad \text{on } \mathcal{A}.$$

Therefore,

$$\mathbb{E}\big[(\log Z - \log m)^2 \mathbf{1}_{\mathcal{A}}\big] \le \frac{4}{m^2}\mathbb{E}[(Z-m)^2] = \frac{4}{m^2}\mathrm{V}ar(Z). \tag{13}$$

Combining (12) and (13) yields an absolute constant $C_{\log} > 0$ such that

$$\mathrm{V}ar(\log Z) \le C_{\log} \cdot \frac{\mathrm{V}ar(Z)}{\mathbb{E}[Z]^2}. \tag{14}$$

Apply (14) to $Z = S + Y_0$ and $Z = S + X$. Using $\mathbb{E}[S + Y_0] = N\mu_Y$ and $\mathrm{V}ar(S + Y_0) = N\nu_Y$, we obtain

$$\mathrm{V}ar(\log(S + Y_0)) \le C_{\log} \cdot \frac{N\nu_Y}{(N\mu_Y)^2}.$$

Similarly, since $\mathbb{E}[S + X] = \mu_X + (N-1)\mu_Y \ge N\mu_Y$ and $\mathrm{V}ar(S + X) = \nu_X + (N-1)\nu_Y \le \nu_X + N\nu_Y$,

$$\mathrm{V}ar(\log(S + X)) \le C_{\log} \cdot \frac{\nu_X + N\nu_Y}{(N\mu_Y)^2}.$$

Combining these bounds yields an absolute constant $C_v > 0$ such that

$$v_\ell = \mathrm{V}ar(\Delta_\ell) \le C_v \cdot \frac{\nu_X + N\nu_Y}{(N\mu_Y)^2}. \tag{15}$$

Finally, using $0 < X \le 1$ (hence $\nu_X \le 1$) and Assumption 4 (so that $\mu_Y = \sigma/q$ and $\nu_Y \le \mathbb{E}[Y^2] = \sigma/(q\sqrt{2})$), (15) implies

$$v_\ell \le C_v \left( \frac{q^2}{N^2\sigma^2} + \frac{1}{\sqrt{2}} \cdot \frac{q}{N\sigma} \right) \le C_v \cdot \frac{q}{N\sigma}(1 + \frac{q}{N\sigma}) \le C_v' \cdot \max\left\{ \frac{q}{N\sigma}, \frac{q^2}{N^2\sigma^2} \right\}.$$

Note that $q/(N\sigma)$ typically increases as $\ell$ decreases during the greedy process. Hence we do not assume a fixed regime; instead we keep the uniform bound $v_\ell \le C_v \cdot \frac{q}{N\sigma}(1 + \frac{q}{N\sigma})$, which is valid for all $\ell$.

**Step 4: Mean lower bound for the row-level log-gap.** Recall

$$\Delta_\ell = \log(S + X) - \log(S + Y_0), \qquad S = \sum_{r=1}^{N-1} Y_r, \qquad m_S := \mathbb{E}[S] = (N-1)\mu_Y,$$

and $N = 3^{\ell-1}$. Under Assumption 7, we derive a quantitative lower bound on the main term on the right-hand side.

*Quantitative upper-tail dominance.* Fix constants $\delta \in (0,1)$ and $z_0 \in [0,1]$. Assume that $X$ dominates $Y_0$ on the upper tail in the following quantitative sense:

$$\Pr[X \ge t] \ge \Pr[Y_0 \ge t] \; + \; \delta \Pr[X \ge t] \qquad \text{for all } t \in [z_0, 1]. \tag{16}$$

Equivalently, $\Pr[Y_0 \ge t] \le (1-\delta)\Pr[X \ge t]$ for all $t \in [z_0, 1]$.

*Integral identity for log-gaps.* For any $m > 0$ and any $x, y \in (0,1]$,

$$\log(m + x) - \log(m + y) = \int_y^x \frac{du}{m+u} = \int_0^1 \frac{\mathbf{1}_{\{x \ge u\}} - \mathbf{1}_{\{y \ge u\}}}{m+u} du.$$

Taking expectation over $(X, Y_0)$ and applying Fubini yields

$$\mathbb{E}[\log(m + X) - \log(m + Y_0)] = \int_0^1 \frac{\Pr[X \ge u] - \Pr[Y_0 \ge u]}{m+u} du. \tag{17}$$

*Lower bounding the integral using* (16). Applying (17) with $m = m_S$ and restricting the integration to $u \in [z_0, 1]$[1], we obtain

$$\mathbb{E}[\log(m_S + X) - \log(m_S + Y_0)] \ge \int_{z_0}^1 \frac{\Pr[X \ge u] - \Pr[Y_0 \ge u]}{m_S + u} du.$$

[1] We keep $z_0 \in (0,1)$ symbolic in the analysis. For the final bound, $z_0$ is fixed once, and all constants are allowed to depend on this fixed choice. In practice, one can choose any $z_0$ satisfying Assumption 7.

By (16), we have $\Pr[X \geq u] - \Pr[Y_0 \geq u] \geq \delta \Pr[X \geq u]$ for all $u \in [z_0, 1]$, hence

$$\mathbb{E}[\log(m_S + X) - \log(m_S + Y_0)] \geq \delta \int_{z_0}^{1} \frac{\Pr[X \geq u]}{m_S + u} du.$$

Since $m_S + u \leq m_S + 1$ for $u \in [0, 1]$, we have $\frac{1}{m_S+u} \geq \frac{1}{m_S+1}$, and thus

$$\mathbb{E}[\log(m_S + X) - \log(m_S + Y_0)] \geq \frac{\delta}{m_S + 1} \int_{z_0}^{1} \Pr[X \geq u] du. \tag{18}$$

Relating the tail mass to $\mu_X$. Recall $\mu_X = \mathbb{E}[X] = \int_0^1 \Pr[X \geq u] du$. Define

$$c_{z_0} := \frac{\int_{z_0}^{1} \Pr[X \geq u] du}{\int_0^1 \Pr[X \geq u] du} \in [0, 1],$$

so that $\int_{z_0}^{1} \Pr[X \geq u] du = c_{z_0} \mu_X$. Moreover, since $m_S = (N-1)\mu_Y$, we have $m_S + 1 \leq N\mu_Y + 1$. Therefore, there exists a constant $c_4 > 0$ such that

$$m_S + 1 \leq c_4 N \mu_Y.$$

Combining these bounds with (18), $\mu_X = 1/\sqrt{2}$, and $\mu_Y = \sigma/q$ yields

$$\mathbb{E}[\log(m_S + X) - \log(m_S + Y_0)] \geq \frac{\delta c_{z_0}}{m_S + 1} \mu_X \geq C_4 \frac{\mu_X}{N \mu_Y} = C_4' \frac{q}{N \cdot \sigma}, \tag{19}$$

where $C_4 := \delta c_{z_0} / c_4 / \sqrt{2}$.

**Step 5: Aggregation across rows and the parameters $(\mu_i, V_i)$.** Fix an iteration $i$ and a remaining coordinate $j \in \mathcal{I}_i$. Under the mean-field Assumption 6, we take $|T_j| = T$, and under Assumption 5 we take the typical row support size to be $\ell_i$. Moreover, by Assumption 2 the row-level log-gaps contributing to the aggregated gap are treated as independent. Therefore the aggregated log-gap is modeled as a sum of $T$ i.i.d. copies of $\Delta_{\ell_i}$, and hence

$$\mu_i := T \mu_{\ell_i}, \qquad V_i := T v_{\ell_i}. \tag{20}$$

Combining (20) with the bounds on $\mu_\ell$ and $v_\ell$ obtained in Steps 3–4 yields an absolute constant $c_5 > 0$ such that

$$\frac{\mu_i^2}{V_i} = T \cdot \frac{\mu_{\ell_i}^2}{v_{\ell_i}} \geq c_5 T \cdot \frac{\frac{q}{3^{\ell_i - 1}\sigma}}{1 + \frac{q}{3^{\ell_i - 1}\sigma}}. \tag{21}$$

Since $\ell_i \leq k$, we have

$$\frac{1}{1 + \frac{q}{3^{\ell_i - 1}\sigma}} \geq \min_{2 \leq \ell \leq k} \frac{1}{1 + \frac{q}{3^{\ell - 1}\sigma}} = \alpha_k.$$[2]

[2] This bound implicitly assumes $\ell_i \geq 2$ for all $i$. Fortunately, if $\ell_i = 1$, then some updated row $t$ has support $S_t = \{j\}$ for a unique coordinate $j$. In this case the marginal-

Defining $c := c_5\alpha_k > 0$, we obtain the uniform bound

$$\frac{\mu_i^2}{V_i} \geq cT \cdot \frac{q}{3^{\ell_i - 1}\sigma}.$$

**Step 6: Iteration success via Gaussian tails for aggregated gaps.** Fix an iteration $i$ and recall $N_i := |\mathcal{I}_i|$. For each remaining coordinate $j \in \mathcal{I}_i$ and each wrong value $u \in \{-1, 0, 1\} \setminus \{s_j\}$, define the aggregated log-gap

$$G_{j,u} := \mathcal{L}_j(s_j) - \mathcal{L}_j(u).$$

If $G_{j,u} > 0$ holds for both wrong values at some coordinate $j$, then

$$\arg \max_{v \in \{-1,0,1\}} \mathcal{L}_j(v) = s_j.$$

In particular, if all gaps are positive,

$$\forall j \in \mathcal{I}_i,\ \forall u \neq s_j : \quad G_{j,u} > 0,$$

then every remaining coordinate has the correct value as its local maximizer, and hence the algorithm's global maximizer $(j^\star, u^\star) = \arg\max_{j \in \mathcal{I}_i, v} \mathcal{L}_j(v)$ necessarily satisfies $u^\star = s_{j^\star}$, i.e., $E_i$ holds. Therefore,

$$E_i^c \ \subseteq \ \bigcup_{j \in \mathcal{I}_i} \ \bigcup_{u \neq s_j} \{G_{j,u} \leq 0\}.$$

By Assumption 3, the $2N_i$ gaps $\{G_{j,u}\}_{j \in \mathcal{I}_i, u \neq s_j}$ are modeled as i.i.d. $\mathcal{N}(\mu_i, V_i)$. Let $Z \sim \mathcal{N}(0, 1)$. Then

$$\Pr[G_{j,u} \leq 0] = \Pr\left[Z \leq -\frac{\mu_i}{\sqrt{V_i}}\right] \leq \exp\left(-\frac{\mu_i^2}{2V_i}\right),$$

using the standard Gaussian tail bound $\Pr[Z \leq -x] \leq e^{-x^2/2}$ for $x \geq 0$. Applying the union bound over the $2N_i$ gaps yields

$$1 - p_i = \Pr[E_i^c] \leq 2N_i \exp\left(-\frac{\mu_i^2}{2V_i}\right), \qquad \text{hence} \qquad p_i \geq 1 - 2N_i \exp\left(-\frac{\mu_i^2}{2V_i}\right). \tag{22}$$

**Step 7: From the parameter condition to a per-iteration failure bound.** Assume (9) holds with a sufficiently large absolute constant $C > 0$. By (21), this implies

$$\frac{\mu_i^2}{2V_i} \geq C \log(3N_i).$$

---

ization in (1) is trivial and $W_t^{(j)}(u) = w((b_t - a_{t,j}u) \bmod q)$ for $u \in \{-1, 0, 1\}$. The choice $u = s_j$ yields residual $e_t$ (hence concentrated near 0), whereas $u \neq s_j$ yields a residual shifted by $a_{t,j}(s_j - u)$ modulo $q$ and thus typically receives only uniform-scale weight. Hence $s_j$ can be recovered immediately by checking the three candidates and selecting the maximizer, after which $j$ can be eliminated from the system. Therefore restricting the minimum defining $\alpha_k$ to $2 \leq \ell \leq k$ is without loss of generality.

Plugging this into the gap bound (22) gives

$$1 - p_i \le 2N_i \exp\left(-\frac{\mu_i^2}{2V_i}\right) \le 2N_i \exp\big(-C\log(3N_i)\big)$$
$$= 2N_i \cdot (3N_i)^{-C} = 2 \cdot 3^{-C} \cdot N_i^{1-C}.$$

Equivalently, for $C \ge 2$ we may absorb the leading factor $2 \cdot 3^{-C} \le 3^{-(C-1)}$ into the same polynomial tail and obtain the clean bound

$$p_i \ge 1 - (3N_i)^{-(C-1)}. \tag{23}$$

**Step 8: Overall success via the product heuristic.** Using the product heuristic (5) together with the elementary inequality $\prod_{i=1}^{n}(1-\varepsilon_i) \ge 1 - \sum_{i=1}^{n}\varepsilon_i$ for $\varepsilon_i \in [0,1]$, we obtain

$$P \;\approx\; \prod_{i=1}^{n} p_i \ge 1 - \sum_{i=1}^{n}(1-p_i) \ge 1 - \sum_{i=1}^{n}(3N_i)^{-(C-1)}.$$

Since $N_i = n - i + 1$, we can reindex the sum to get

$$P \ge 1 - \sum_{N=1}^{n}(3N)^{-(C-1)}. \tag{24}$$

In particular, if $C > 2$ then the tail sum is uniformly bounded:

$$\sum_{N=1}^{n}(3N)^{-(C-1)} \le \sum_{N=1}^{\infty}(3N)^{-(C-1)} = 3^{-(C-1)}\zeta(C-1),$$

where $\zeta(s) := \sum_{N=1}^{\infty} N^{-s}$ denotes the Riemann zeta function (finite for $s > 1$). Hence $P$ is bounded away from 0 by an absolute constant (and improves as $C$ increases). □

### 3.4 Inapplicable to Sparse LPN

Our attack critically relies on the fact that each row involves only $k$ secret coordinates. Writing a single equation as

$$b_t \equiv \sum_{j \in S_t} a_{t,j} s_j + e_t \pmod{q}, \qquad |S_t| = k,$$

we can enumerate all candidate assignments to the restricted secret $\mathbf{s}|_{S_t} \in \{-1,0,1\}^{S_t}$ and inspect the induced residuals. In the LWE/RLWE setting (modulus $q \ge 3$ and a small error), exactly one assignment—the true $\mathbf{s}|_{S_t}$—produces a residual distributed like the error and hence concentrated near 0, while the other $3^k - 1$ assignments typically yield values that behave close to uniform in

$\mathbb{Z}_q$. In other words, the enumeration produces a distribution with a single "distinguished" outcome (a clear spike near 0) amid essentially random alternatives, and this per-row distinguishability is precisely what our greedy scoring exploits and amplifies across many rows.

For sparse LPN, however, the same enumeration loses this asymmetry. Over $\mathbb{F}_2$ we have

$$b_t = \sum_{j \in S_t} a_{t,j} s_j + e_t \pmod 2, \qquad s_j \in \{0,1\}.$$

Fixing a row $t$ and ranging over all $2^k$ assignments to $\mathbf{s}|_{S_t}$, the induced values of the inner product $\sum_{j \in S_t} a_{t,j} s_j \bmod 2$ are perfectly balanced: as long as the row contains at least one active coefficient, there are exactly $2^{k-1}$ assignments mapping to 0 and $2^{k-1}$ mapping to 1. Consequently, the residuals are also balanced, yielding a uniform-looking multiset with no distinguished "small-residual" spike. Since our greedy algorithm depends on detecting and accumulating such per-row statistical spikes, this explains why it does not provide an advantage for sparse LPN.

## 4 Cryptanalysis of spLWE in the Geometric Regime

In this section we discuss the complementary regime $q > 3^k$, in which a *single* sparse equation already induces a low-dimensional lattice problem. This regime is qualitatively different from the aggregation regime analyzed in Sections 3.1–3.3, because one can attack each row essentially independently.

Recall that each sample $(\mathbf{a}_t, b_t)$ satisfies

$$b_t \equiv \langle \mathbf{a}_t, s \rangle + e_t \pmod q, \qquad e_t \leftarrow D_{\mathbb{Z},\sigma},$$

and $\mathbf{a}_t$ has exactly $k$ nonzero entries. Let $S_t := \mathsf{supp}(\mathbf{a}_t) \subseteq [n]$ be the support of the $t$-th row, so $|S_t| = k$. Define the restricted vectors

$$\bar{\mathbf{a}}_t := \mathbf{a}_t|_{S_t} \in \mathbb{Z}_q^k, \qquad \bar{\mathbf{s}}_t := s|_{S_t} \in \{-1,0,1\}^k.$$

Then the row equation reduces to a $k$-dimensional LWE(-type) relation

$$b_t - \langle \bar{\mathbf{a}}_t, \bar{\mathbf{s}}_t \rangle \equiv e_t \pmod q.$$

**A target lattice.** Consider the lattice $\Lambda_t \subseteq \mathbb{Z}^{k+2}$ generated by the basis

$$B_t := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ b_t & \bar{a}_{t,1} & \bar{a}_{t,2} & \cdots & \bar{a}_{t,k} & q \end{pmatrix} \in \mathbb{Z}^{(k+2)\times(k+2)}.$$

(Here $\bar{a}_{t,j}$ denotes the $j$-th coordinate of $\bar{\mathbf{a}}_t$.) By construction, $\det(\Lambda_t) = |\det(B_t)| = q$. It is clear that $\Lambda_t$ contains the structured vector

$$\mathbf{t}_t := (1, -\bar{\mathbf{s}}_t, e_t)^{\mathsf{T}}.$$

Since $\bar{\mathbf{s}}_t \in \{-1, 0, 1\}^k$ and $e_t \leftarrow D_{\mathbb{Z},\sigma}$, we have

$$\|\mathbf{t}_t\|^2 = 1 + \|\bar{\mathbf{s}}_t\|^2 + e_t^2 \le 1 + k + e_t^2, \qquad \text{and typically } \|\mathbf{t}_t\| \approx \sqrt{2k/3 + \sigma^2}.$$

**Why $q > 3^k$ makes $\mathbf{t}_t$ exceptionally short.** In dimension $d := k + 2$ with $\det(\Lambda_t) = q$, Minkowski's first theorem (or, heuristically, the Gaussian heuristic) suggests a shortest-vector scale on the order

$$\lambda_1(\Lambda_t) \lesssim \sqrt{d} \cdot q^{1/d} = \sqrt{k+2} \cdot q^{1/(k+2)}.$$

In the regime $q > 3^k$, the determinant term $q^{1/(k+2)}$ grows enough that $\mathbf{t}_t$ becomes a conspicuous outlier relative to typical random lattice vectors, and one expects $\mathbf{t}_t$ (or a very closely related vector) to be recovered by an SVP solver like a sieving algorithm. In other words, in this regime each sparse row yields a low-dimensional lattice instance from which we can recover the partial secret $\bar{\mathbf{s}}_t = s|_{S_t}$ together with the row error $e_t$.

**Recovering the full secret.** Running the above procedure over many rows yields multiple partial views $\{(S_t, \bar{\mathbf{s}}_t)\}$ of the secret. As soon as the union of supports covers $[n]$ (with overlaps providing consistency checks), we can reconstruct $s$ by merging these restrictions.

**Cost model.** To estimate the cost of solving SVP in dimension $d = k + 2$, we follow the cost model of [BDGL16]. In this model, the asymptotic running time (in bit operations) is approximated by

$$T_{\mathrm{SVP}}(d) \;\approx\; 2^{0.292d}.$$

Therefore, in the regime $q > 3^k$, our overall attack on spLWE reduces to solving such $(k + 2)$-dimensional SVP instances (one per informative row), and its complexity is governed by the above expression with $d = k + 2$.

## 5 Experiments

In this section we experimentally validate the success condition of Theorem 3.2. Recall that the theorem predicts recovery when

$$\frac{m \cdot k \cdot q}{n \cdot 3^{k-1} \cdot \sigma} \;\ge\; C \cdot \log(3n)$$

for a universal constant $C > 0$; equivalently, when the ratio $\Gamma := \frac{mkq}{n \cdot 3^{k-1} \cdot \sigma \cdot \ln(3n)}$ exceeds $C$. In Section 5.1 we sweep each parameter on toy instances and observe the resulting recovery rates. Our implementation is publicly available. [3]

In Section 5.2 we compare our attack with the meet-in-the-middle attack of [KSM24] on their concrete parameter sets.

[3] https://github.com/ksprsk/sparse_lwe_anon

### 5.1 Parameter sweeps on toy instances

We fix a base parameter set $(n, m, k, q, \sigma) = (200, 1600, 10, 1021, 3.19)$ and vary each of the five parameters individually while keeping the others fixed. For every parameter point we run 100 independent trials and record the full-recovery rate. Figures 2a to 2e plot the results, and Figure 3 overlays all five sweeps by plotting the recovery rate against $\Gamma$.

**What changes when sweeping each parameter.** Our success condition in Theorem 3.2 suggests that recovery is governed by the signal-to-noise ratio captured by $\Gamma = \frac{mkq}{n \cdot 3^{k-1} \cdot \sigma \cdot \ln(3n)}$. To build intuition, we briefly explain what each sweep does to the instance and why the observed trends match this prediction.

- **Sweeping $m$ (number of equations/rows).** Increasing $m$ increases the amount of independent evidence aggregated in each score, so the log-likelihood gaps concentrate more strongly around their means. Consequently, the recovery rate improves monotonically with $m$ (Figure 2a), consistent with the linear dependence $\Gamma \propto m$.

- **Sweeping $n$ (dimension/number of coordinates).** Larger $n$ makes the problem harder in two ways: (i) there are more coordinates to recover, and (ii) the algorithm faces a more severe multiple-comparisons effect because it selects the maximum over a larger index set. Both effects are reflected in Figure 2b by a clear decline in recovery as $n$ grows, in agreement with $\Gamma \propto 1/(n \ln(3n))$.

- **Sweeping $k$ (row sparsity / number of incident constraints).** The parameter $k$ has the strongest impact because it appears through the exponential penalty $3^{k-1}$ in the denominator of $\Gamma$. As $k$ increases, each coordinate is entangled with more competing hypotheses, which dilutes the per-hypothesis advantage and dramatically reduces the separation between correct and wrong scores. This explains the sharp threshold in the $k$-sweep (Figure 2c).

- **Sweeping $q$ (modulus).** A larger modulus increases the effective dynamic range of the score contributions and, in our model, scales the mean separation linearly, yielding $\Gamma \propto q$. Accordingly, the recovery rate improves with $q$ in Figure 2d. Minor non-monotonicity is expected on small toy instances due to finite-sample variance (only 100 trials per point) and the discreteness of admissible $q$ values.

- **Sweeping $\sigma$ (noise level).** Increasing $\sigma$ weakens statistical distinguishability, shrinking the normalized gaps and reducing the probability that the correct hypothesis dominates the wrong ones. Thus the recovery rate decreases as $\sigma$ grows (Figure 2e), matching $\Gamma \propto 1/\sigma$.

**Putting everything together.** Across all five sweeps, the recovery behavior is well summarized by the single predictor $\Gamma$: when we re-plot the outcomes against

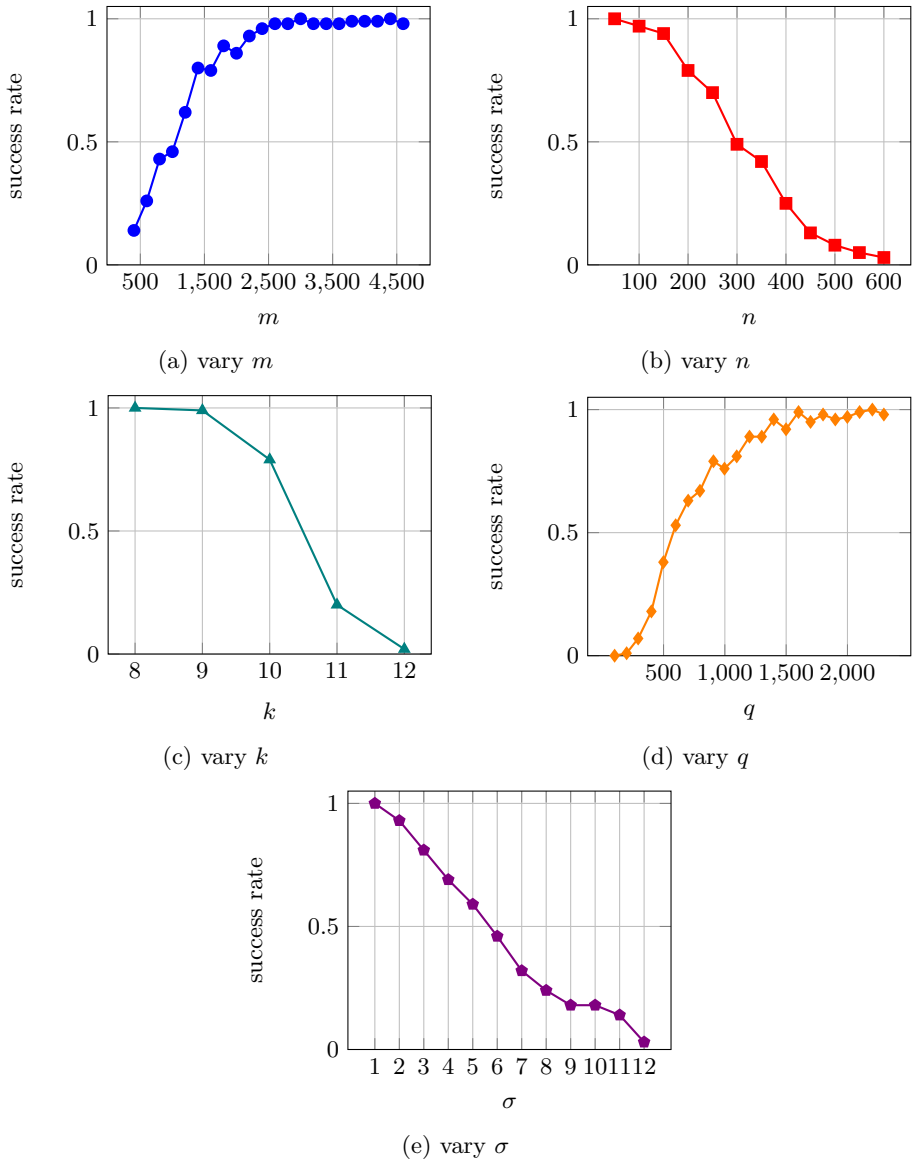



Fig. 2: Parameter sweeps on toy instances with base parameters $(n, m, k, q, \sigma) = (200, 1600, 10, 1021, 3.19)$. Each subplot varies one parameter while keeping others fixed.

$\Gamma$ and overlay all points (Figure 3), the five curves collapse into a common transition region. Empirically, the recovery rate is close to 1 for $\Gamma \gtrsim 0.3$ and

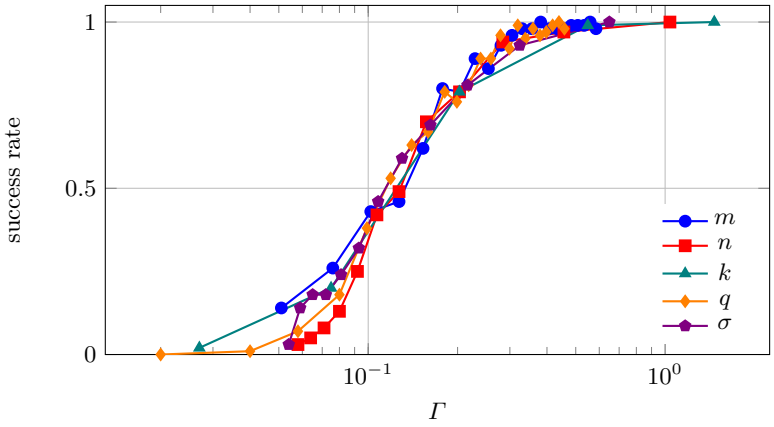



Fig. 3: Recovery rate as a function of $\Gamma$ for the five parameter sweeps.

drops to (nearly) 0 for $\Gamma \lesssim 0.08$, indicating that $\Gamma$ captures the success/failure boundary across different parameter variations. Among the individual sweeps, the $k$-sweep exhibits the sharpest transition—from 1.0 at $k = 8$ to 0.02 at $k = 12$—consistent with the exponential factor $3^{k-1}$ in the denominator of $\Gamma$.

Overall, the fact that varying different parameters yields a similar transition behavior when viewed through $\Gamma$ supports the adequacy of the heuristic modeling in Section 3.3.

### 5.2 Comparison with KSM24

We compare our attack with the MitM-based combinatorial attack of Kalam–Sarkar–Meier [KSM24], which targets the same ternary sparse-LWE problem. Since [KSM24] considers ternary errors $e_t \in \{-1, 0, 1\}$, we approximate this by setting $\sigma = 2$ in our Gaussian-error model, which closely matches the second moment of the uniform ternary distribution.

**Complexity of our attack.** The running time depends on whether a single sparse row already induces a low-dimensional lattice problem. When $q > 3^k$, each row $(\mathbf{a}_t, b_t)$ gives rise to a $d$-dimensional SVP-type instance with $d = k+2$, which we solve with lattice sieving [BDGL16, Duc18]. By the cost model in Section 4, the asymptotic running time (in bit operations) of solving SVP in dimension $d$ is approximated by

$$T_{\mathrm{SVP}}(d) \;\approx\; 2^{0.292 \cdot d}.$$

When $q < 3^k$, we instead apply our greedy procedure (Algorithm 1), whose complexity is

$$T_{\mathrm{greedy}} = O(mk \cdot 3^k) \qquad (\textit{Lemma } 3.1).$$

Accordingly, the Ours column in the tables below reports $0.292(k - d_{\text{free}}(k))$[4] bits in the regime $q > 3^k$, and $\log_2(mk \cdot 3^k)$ bits otherwise.

**Results.** Tables 1 to 3 compare the two attacks across the parameter sets of [KSM24]. Boldface entries indicate the lower complexity. The dashes (–) in the 'Ours' column indicate instances where $\Gamma$ is significantly less than 0.08, implying that our attack would not be applicable under these parameters.

Table 1: Comparison at $q = 2^{64}$, $\sigma = 2$.

| | $m = 2^{13}$ | | | | $m = 2^{17}$ | | | | $m = 2^{21}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | $n$ | KSM | Ours | $\Gamma$ | $n$ | KSM | Ours | $\Gamma$ | $n$ | KSM | Ours | $\Gamma$ |
| 20 | 1218 | 28 | **6** | – | 1425 | 28 | **6** | – | 1656 | 29 | **6** | – |
| 30 | 1143 | 35 | **9** | – | 1265 | 36 | **9** | – | 1395 | 36 | **9** | – |
| 40 | 1110 | 43 | **12** | – | 1195 | 43 | **12** | – | 1285 | 43 | **12** | – |
| 50 | 1090 | **49** | – | $10^{-3}$ | 1158 | **49** | – | 0.03 | 1234 | **49** | – | 0.40 |

Table 2: Comparison at $q = 2^{32}$, $\sigma = 2$.

| | $m = 2^{13}$ | | | | $m = 2^{17}$ | | | | $m = 2^{21}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | $n$ | KSM | Ours | $\Gamma$ | $n$ | KSM | Ours | $\Gamma$ | $n$ | KSM | Ours | $\Gamma$ |
| 20 | 1218 | 28 | **6** | – | 1425 | 28 | **6** | – | 1656 | 28 | **6** | – |
| 30 | 1143 | 94 | – | $10^{-4}$ | 1265 | 101 | – | 0.01 | 1395 | 113 | **73** | 0.17 |
| 40 | 1110 | 201 | – | $10^{-8}$ | 1195 | 218 | – | $10^{-7}$ | 1285 | 236 | – | $10^{-6}$ |
| 50 | 1090 | 316 | – | $10^{-13}$ | 1158 | 334 | – | $10^{-12}$ | 1234 | 356 | – | $10^{-11}$ |

Table 3: Comparison at $q = 2^{16}$, $\sigma = 2$.

| | $m = 2^{13}$ | | | | $m = 2^{17}$ | | | | $m = 2^{21}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | $n$ | KSM | Ours | $\Gamma$ | $n$ | KSM | Ours | $\Gamma$ | $n$ | KSM | Ours | $\Gamma$ |
| 20 | 1218 | 240 | – | $10^{-4}$ | 1425 | 285 | – | $10^{-3}$ | 1656 | 321 | – | 0.08 |
| 30 | 1143 | 479 | – | $10^{-8}$ | 1265 | 535 | – | $10^{-7}$ | 1395 | 581 | – | $10^{-6}$ |
| 40 | 1110 | 686 | – | $10^{-13}$ | 1195 | 746 | – | $10^{-12}$ | 1285 | 786 | – | $10^{-11}$ |
| 50 | 1090 | 837 | – | $10^{-18}$ | 1158 | 888 | – | $10^{-17}$ | 1234 | 945 | – | $10^{-15}$ |

[4] $d_{\text{free}}(k)$ is a dimension for free parameter in [Duc18].

**Discussion.** At $q = 2^{64}$ (Table 1), for $k \leq 40$ we have $q > 3^k$, and lattice sieving reduces the cost to 6–12 bits compared with 28–43 bits for the MitM attack of [KSM24]. At $q = 2^{32}$ (Table 2), the regime $q > 3^k$ still holds for $k = 20$, giving 6 bits vs. 28 bits. For $k \geq 30$ the greedy attack applies and remains faster in all cases (e.g., 65 vs. 94 bits at $k$=30, $m$=$2^{13}$), though $\Gamma \ll 1$ for these parameters, so full recovery is not guaranteed by our analysis. At $q = 2^{16}$ (Table 3), the greedy attack is uniformly faster (e.g., 49 vs. 240 bits at $k$=20) but $\Gamma \ll 1$ throughout.

In summary, our approach is most effective in the regime $q > 3^k$, where lattice sieving yields both lower complexity and reliable recovery. This is precisely the regime where sparsity provides the greatest efficiency benefit to the cryptosystem, and hence the regime of primary cryptanalytic interest.

## Acknowledgment

Parts of this paper were edited for grammar using ChatGPT and Gemini.

## References

ABW10. Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010.

ACF+15. Martin R Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for lwe problems. *ACM Communications in Computer Algebra*, 49(2):62–62, 2015.

ACW19. Martin R Albrecht, Benjamin R Curtis, and Thomas Wunderer. Exploring trade-offs in batch bounded distance decoding. In *International Conference on Selected Areas in Cryptography*, pages 467–491. Springer, 2019.

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.

AGVW17. Martin R Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving usvp and applications to lwe. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 297–322. Springer, 2017.

Alb17. Martin R Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 103–129. Springer, 2017.

Ale03. Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307. IEEE, 2003.

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 10–24. SIAM, 2016.

BDK+18. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

BG14. Shi Bai and Steven D Galbraith. Lattice decoding attacks on binary LWE. In *Australasian Conference on Information Security and Privacy*, pages 322–337. Springer, 2014.

BGV14. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

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

BLLW22. Lei Bi, Xianhui Lu, Junjie Luo, and Kunpeng Wang. Hybrid dual and meet-LWE attack. In *ACISP 2022*, volume 13494 of *LNCS*, pages 168–188. Springer, 2022.

CGGI16. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *international conference on the theory and application of cryptology and information security*, pages 3–33. Springer, 2016.

CHHS19. Jung Hee Cheon, Minki Hhan, Seungwan Hong, and Yongha Son. A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE. *IEEE Access*, 7:89497–89506, 2019.

CKKS17. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International conference on the theory and application of cryptology and information security*, pages 409–437. Springer, 2017.

DIJL23. Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin. Multi-party homomorphic secret sharing and sublinear mpc from sparse lpn. In *Annual International Cryptology Conference*, pages 315–348. Springer, 2023.

DJ24. Quang Dao and Aayush Jain. Lossy cryptography from code-based assumptions. In *Annual International Cryptology Conference*, pages 34–75. Springer, 2024.

Duc18. Léo Ducas. Shortest vector from lattice sieving: a few dimensions for free. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 125–145. Springer, 2018.

FV12. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Paper 2012/144, 2012.

Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th STOC*, pages 197–206, 2008.

HG07. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In *Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27*, pages 150–169. Springer, 2007.

HKLS22. Minki Hhan, Jiseung Kim, Changmin Lee, and Yongha Son. How to meet ternary lwe keys on babai's nearest plane. Cryptology ePrint Archive, Paper 2022/1473, 2022. https://eprint.iacr.org/2022/1473.

HS07. Guillaume Hanrot and Damien Stehlé. Improved analysis of kannan's shortest lattice vector algorithm. In *Annual international cryptology conference*, pages 170–186. Springer, 2007.

JLS24. Aayush Jain, Huijia Lin, and Sagnik Saha. A systematic study of sparse lwe. In *Annual International Cryptology Conference*, pages 210–245. Springer, 2024.

Kan83. Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 193–206, 1983.

KKN+25. Alexandr Karenin, Elena Kirshanova, Julian Nowakowski, Eamonn W Postlethwaite, and Fernando Virdia. Cool+ cruel= dual. *Cryptology ePrint Archive*, 2025.

KSM24. Abul Kalam, Santanu Sarkar, and Willi Meier. A combinatorial attack on ternary sparse learning with errors (slwe). *Cryptology ePrint Archive*, 2024.

LP11. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *Cryptographers' Track at the RSA Conference*, pages 319–339. Springer, 2011.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*, pages 1–23. Springer, 2010.

May21. Alexander May. How to meet ternary LWE keys. In *Annual International Cryptology Conference*, pages 701–731. Springer, 2021.

MR09. Daniele Micciancio and Oded Regev. *Lattice-based Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

NMW+24. Niklas Nolte, Mohamed Malhou, Emily Wenger, Samuel Stevens, Cathy Li, François Charton, and Kristin Lauter. The cool and the cruel: separating hard parts of lwe secrets. In *International Conference on Cryptology in Africa*, pages 428–453. Springer, 2024.

Pla18. Rachel Player. *Parameter selection in lattice-based cryptography*. PhD thesis, Royal Holloway, University of London, 2018.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

WSM+25. Emily Wenger, Eshika Saxena, Mohamed Malhou, Ellie Thieu, and Kristin Lauter. Benchmarking attacks on learning with errors. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 279–297. IEEE, 2025.