TAPAS: Datasets for Learning the Learning with Errors Problem

Eshika Saxena[*]
FAIR at Meta

Alberto Alfarano
FAIR at Meta

François Charton[†]
FAIR at Meta

Emily Wenger[†]
Duke University

Kristin Lauter[†]
FAIR at Meta

Abstract

AI-powered attacks on Learning with Errors (LWE)—an important hard math problem in post-quantum cryptography—rival or outperform "classical" attacks on LWE under certain parameter settings. Despite the promise of this approach, a dearth of accessible data limits AI practitioners' ability to study and improve these attacks. Creating LWE data for AI model training is time- and compute-intensive and requires significant domain expertise. To fill this gap and accelerate AI research on LWE attacks, we propose the TAPAS datasets, a **t**oolkit for **a**nalysis of **p**ost-quantum cryptography using **A**I **s**ystems. These datasets cover several LWE settings and can be used off-the-shelf by AI practitioners to prototype new approaches to cracking LWE. This work documents TAPAS dataset creation, establishes attack performance baselines, and lays out directions for future work.

1 Introduction

The Learning with Errors (LWE) problem is a hard math problem used in post-quantum cryptography. Put simply, the LWE problem is: given a set of samples $(\mathbf{a}, b) \in \mathbb{Z}_q$ where b is the noisy inner product of $\mathbf{a}$ with a secret vector $\mathbf{s}$, e.g. $b = \mathbf{a} \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q$, recover $\mathbf{s}$. LWE is attractive for use in post-quantum cryptography because no classical **or** quantum algorithms are known to recover $\mathbf{s}$ given many ($\mathbf{a}$, b) samples in polynomial time under certain conditions. Thus, LWE can serve as an alternative to current hard math problems like integer factorization used in cryptosystems like RSA, which can be broken by quantum computers due to Shor's algorithm [33]. Consequently, LWE is used in several US-government standardized PQC systems, such as CRYSTALS-KYBER [4], and also in homomorphic encryption applications [2], which enable computation on encrypted data.

Given the importance of LWE in securing the future internet, understanding its security is critical. Cryptanalysis of LWE and LWE-based cryptosystems is a long-studied problem in theoretical cryptography, and analysis from this community has not uncovered any notable weaknesses of LWE. However, vigorous security analysis demands examining LWE from multiple angles, including using previously-underutilized tools like AI, to ensure it is secure against a range of attacks.

AI attacks on LWE match or outperform classical attacks in certain settings. A series of recent works [35, 21, 20, 34] has explored a novel approach using AI models to recover LWE secrets. The attack setup is simple: train a model using millions of LWE samples $(\mathbf{a}, b)$ (generated from a small initial sample set) to predict b given a particular $\mathbf{a}$. If the model performs better than random on this task, it must have implicitly learned information about the secret $\mathbf{s}$. The secret can then be recovered via carefully crafted model queries.

[*]Corresponding author: eshika@meta.com
[†]Co-senior authors

Traditional LWE cryptanalysis methods treat lattice-based systems as a mathematical problem to be solved rather than data to be analyzed. The AI approach inverts this paradigm, instead searching for patterns in large quantities of LWE data. Not only is this AI approach novel, it is effective: AI models match or outperform existing "classical" attacks on lattice cryptography under certain conditions [36]. Additional work [23] suggests that further helpful cryptanalytic insights could emerge from enhancing this data-centric approach to LWE. Finally, recent work argues that such AI-based cryptanalysis approaches are not inherently limited, raising the possibility that they could scale further [32].

AI research on LWE (and cryptography) benefits both fields. The intersection of AI and cryptography is a long-recognized [28] but relatively unexplored area, offering many opportunities for meaningful research. By studying the application of AI techniques to LWE cryptanalysis, researchers can unlock new techniques and approaches to complement those of [35, 20, 21, 34]. Both the AI and cryptography communities would benefit from this.

On the cryptography side, additional work on the AI approach may lead to the discovery of better attacks on LWE. Discovering and mitigating these before LWE is widely deployed is critical for internet security. The development of more robust cryptographic protocols will have important implications for secure communication and data protection.

On the AI side, studying AI attacks on LWE will require AI researchers to tackle critical problems for enabling higher-order mathematical reasoning in AI models. Currently, AI models struggle to perform mathematical operations such as modular arithmetic, which is a fundamental part of the LWE problem [16, 24]. As AI practitioners gain exposure to important cryptography problems, further investigation into how and why AI models fail on these problems will help advance the capabilities of AI models.

Dataset availability roadblocks AI LWE research. Despite these myriad benefits to both communities, it remains difficult for AI researchers to begin studying LWE attacks. Running AI attacks on LWE requires significant *preprocessing* of LWE samples, which enables models to learn better [21]. Preprocessing can take hours to days, can be computationally expensive, and requires some expertise in lattice reduction algorithms—all of which can deter AI practitioners who may lack time, compute, and domain knowledge. Prior work has open-sourced some LWE datasets [36, 20], but these contain only a few million LWE samples, making it difficult to train bigger models or develop scaling laws. Additionally, published datasets only consider difficult LWE parameter regimes that may not prove accessible (computationally or otherwise) for AI researchers.

Our Contribution: large-scale, open-source LWE datasets. For the AI community to contribute meaningfully to the study of LWE cryptanalysis, large datasets of reduced LWE samples must exist and be publicly accessible. Such datasets would provide the AI community with an easy entry point to the important and often inaccessible research intersection between AI and cryptography. This motivates our work: *providing large-scale datasets of preprocessed Learning with Errors (LWE) data across many parameter settings* to make this research more accessible to the AI community. To this end, this paper provides the following:

- **Five new datasets of LWE samples**, designed for off-the-shelf use in AI applications, hosted on Huggingface[1] for easy access.
- **Baseline results for SALSA and Cool and Cruel attack performance on provided datasets**, setting a baseline on which future work can improve.

Paper organization. The rest of the paper proceeds as follows. §2 gives context on the Learning with Errors problem and prior cryptanalytic approaches. §3 describes the datasets we provide. §4 gives baseline attack performance on datasets. §5 discusses related datasets and benchmarks, and §7 suggests ways our datasets can be used to enhance AI attacks on LWE.

2 Background: Cryptanalysis of Learning with Errors

The Learning with Errors (LWE) problem was first proposed for cryptography by Regev [26, 27]. Since no classical or quantum algorithms are known to solve LWE in polynomial time for certain parameter settings, it has been widely adopted for use in post-quantum cryptosystems. Substantial

[1] https://huggingface.co/datasets/facebook/TAPAS

efforts have been made to discover weaknesses in LWE, to ensure it is a solid foundation for post-quantum cryptography. This section describes the basics of LWE and gives an overview of attacks against it.

LWE Basics. There are two variants of the LWE problem: Search-LWE and Decision-LWE. The Search-LWE problem is: given samples $(\mathbf{A}, \mathbf{b})$—where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is populated with uniform random entries modulo a large prime q, and $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$ is the noisy matrix-vector product of $\mathbf{A}$ and a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ with error vector $\mathbf{e}$—recover the secret vector $\mathbf{s}$. In Decision-LWE, one is tasked with deciding whether $(\mathbf{A}, \mathbf{b})$ are LWE samples or were generated uniformly at random. The secret $\mathbf{s}$ and error $\mathbf{e}$ are chosen from some probability distributions, which we denote by χ_s and χ_e. The hardness of LWE depends on n, q, and these distributions. We denote a single row of $(\mathbf{A}, \mathbf{b^t})$ as $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, and use this notation going forward to refer to a specific LWE sample.

"Classical" Attacks on LWE. Generally, classical or algebraic attacks on LWE try to find short vectors in a lattice Λ constructed from samples $(\mathbf{A}, \mathbf{b})$. Finding a short enough vector is sufficient to recover the LWE secret [25]. Most approaches leverage lattice reduction algorithms like LLL and BKZ [19, 31, 10] to *reduce* or shorten the vectors in Λ. They then run additional algorithms to recover $\mathbf{s}$ from this reduced lattice basis, e.g. [11, 13, 37, 1, 5, 8, 22, 7, 3]. See [36] for a broad overview of classical attacks on LWE.

AI Attacks on LWE. So far, two AI-powered attacks have been proposed against LWE: SALSA and its variants [35, 20, 21, 34] and Cool & Cruel [23]. The SALSA attack works as follows. Start with a set of $4n$ LWE pairs $(\mathbf{a}, b)$, where n is the length of the vector $\mathbf{a}$. Through repeated sub-sampling of these $4n$ samples, create several million new LWE matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ with corresponding $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$. Run lattice reduction on these samples (further details in §3) to produce reduced samples $(\mathbf{RA}, \mathbf{Rb} = \mathbf{RA} \cdot \mathbf{s} + \mathbf{Re})$. Then, train a model $\mathcal{M}$ to predict $\mathbf{Rb}$ from $\mathbf{RA}$. If this model ever achieves better-than-random performance on this task, it has implicitly learned $\mathbf{s}$, and can be selectively queried to recover $\mathbf{s}$.

The Cool & Cruel attack takes a different but related approach. It uses the same subsampling-then-reduce procedure from SALSA to produce $(\mathbf{RA}, \mathbf{Rb})$ pairs. However, it then observes an asymmetry in the columns of $\mathbf{RA}$ due to a quirk of the lattice reduction algorithms—the first columns of $\mathbf{RA}$ are not reduced, while the last columns are. This asymmetry admits an attack in which the entries of the secret $\mathbf{s}$ corresponding to the unreduced $\mathbf{RA}$ columns can be guessed via brute force, while the other entries of $\mathbf{s}$ can be recovered via linear regression (making this an "AI" attack) [36].

Comparing Classical and AI Attacks on LWE. Prior work [36] provided the first empirical comparisons of classical and AI attacks on LWE and found AI attacks matched or outperformed classical attacks on certain weakened LWE settings (realistic n and q sizes, sparse secrets). This work motivates our own, as it demonstrates that AI attacks are already competitive in this space.

3 TAPAS: Datasets for Learning the Learning with Errors Problem

Given the competitive performance of AI attacks on LWE and the ongoing need to assess the security of LWE-based cryptosystems, our goal is to make LWE attack development accessible to AI researchers. To this end, we provide five datasets that are ready for AI model training off the shelf. Building on the naming convention of the initial AI attacks on LWE (SALSA, SALSA Picante, etc), we refer to the provided datasets collectively as TAPAS: a **T**oolkit for **A**nalysis of **P**ost-quantum cryptography using **AI** **S**ystems. These datasets cover a wide range of LWE hardness settings, as controlled by the lattice dimension n, modulus q, and secret/error distributions χ_s and χ_e. By providing a variety of datasets with varying hardness, we hope to enable AI researchers to prototype novel approaches on scaled-down versions of the LWE problem and then also scale up to larger, more realistic datasets. This section presents details of our LWE datasets, including their parameters and development process.

3.1 Dataset Overview

Important LWE parameters. We first give an overview of important parameters for LWE, which are relevant for attacks on LWE. Every instance of LWE is defined by a lattice dimension n, a modulus

q, and secret and error distributions χ_s and χ_e, which determine problem hardness. We also briefly discuss variants of LWE like R-LWE and MLWE, although these are not used in our datasets.

- **Dimension** n is the number of entries in the random vector $\mathbf{a}$. In practical LWE implementations, n is a power of 2, to avoid powerful attacks on LWE variants when n is not a power of 2 [14].
- The **prime modulus** q defines the field where all lattice operations are carried out. All vector and lattice entries are integers modulo q.
- The coordinates of the secret $\mathbf{s}$ are chosen from a **secret distribution** χ_s. Although early implementations of LWE chose secret coordinates uniformly at random from $\mathbb{Z}_q$, computational efficiency and improved functionality motivates choosing small entries for $\mathbf{s}$. So secrets are often chosen from narrow distributions such as binary and ternary secrets, $\mathbf{s} \in \{0, 1\}$ or $\mathbf{s} \in \{-1, 0, 1\}$, recommended for use in homomorphic encryption operations [2, 6], where efficiency is key. Similarly, in the standardized CRYSTALS-KYBER system, χ_s is a binomial distribution with $\eta = 3$, which corresponds to $\mathbf{s} \in \{-3, -2, -1, 0, 1, 2, 3\}$.
- The coordinates of the error $\mathbf{e}$ are chosen from an **error distribution** χ_e. For homomorphic encryption applications $\chi_e = N(0, 3)$ (rounded to the nearest integer) [2, 6], regardless of the lattice dimension or modulus size. For CRYSTALS-KYBER, χ_e is again binomial with $\eta = 3$, so $\mathbf{e} \in \{-3, -2, -1, 0, 1, 2, 3\}$.
- LWE has several **variants**. In basic LWE, the coordinates of $\mathbf{A}$ are chosen uniformly at random from $\mathbb{Z}_q$. However, this approach is computationally inefficient, since it requires storing a full $n \times n$ matrix. To address this, Ring-LWE was proposed, in which a sample is defined as $(a(x), b(x) = a(x)s(x) + e(x))$ here $a(x), b(x)$ are polynomials in a cyclotomic ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ and n is a power of 2. RLWE is widely used in homomorphic encryption applications [2, 6] and only requires storing the n-long polynomial vector. Yet another LWE variant is Module Learning with Errors (MLWE), which builds on RLWE but works in a free R_q-Module $\mathcal{M} = R_q^k$ of rank k. An MLWE sample is a pair $(\mathbf{a}, b)$ where $\mathbf{a} = (a_1(x), a_2(x), \ldots, a_k(x)) \in \mathcal{M}$, and $b = \mathbf{a} \cdot \mathbf{s} + e \in R_q$ for some secret vector of polynomials $\mathbf{s} = (s_1(x), s_2(x), \ldots, s_k(x)) \in \mathcal{M}$, and error polynomial e chosen from a specified distribution. MLWE is used in CRYSTALS-KYBER [4] and is between LWE and RLWE in terms of computational efficiency, requiring storage of k n-long vectors. Although RLWE and MLWE are important, this work considers only LWE for simplicity.

Our datasets. In this work, we release 5 reduced LWE datasets with varying parameter settings. Table 1 gives an overview of the parameters for these datasets. The parameters are chosen to offer a wide range of problem hardness, including several parameter settings used in standardized (or proposed standardized) LWE-based cryptosystems. Access to datasets with varying parameter settings enables investigation along many different axes to see what affects attack performance, such as: n (sequence length), q (size of the modulus), and ρ (quality of reduction). In this work, we provide datasets with 10x (40 million) and 100x (400 million) the number of samples from prior work to enable further research on how the number of samples affects attack performance.

Table 1: **Overview of datasets provided in this work**. *n and q are the LWE dimension and modulus, respectively. ω is the penalty parameter used in reduction (prior experimental work found that 10 is sufficient for Gaussian error with $\sigma = 3.2$ [21]). ρ is the stddev reduction, a metric reported in [20].*

n	$\log_2 q$	ω	ρ	# samples
256	20	10	0.4284	400M
512	12	10	0.9036	40M
512	28	10	0.6740	40M
512	41	10	0.3992	40M
1024	26	10	0.8600	40M

3.2 Data Generation

To create our datasets, we leverage the data preprocessing techniques first proposed in [21] and improved in [23]. All prior work on applying AI to LWE problems found that this preprocessing step was critical to improving AI performance on the task, so we believe it is advantageous to the community to publish preprocessed datasets.

Each dataset starts with a set of $4n$ LWE samples $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{4n \times n}, \mathbb{Z}_q^{4n}$. We assume the samples are eavesdropped, a common assumption in LWE attack literature. Then, we employ the subsampling trick of [21] to create millions of new LWE samples from these: select m random indices from the $4n$ set to form $(\mathbf{A}_i, \mathbf{b}_i) \in \mathbb{Z}_q^{m \times n}, \mathbb{Z}_q^m$. This trick allows us to create up to $\binom{4n}{m}$ samples from the initial starting set—billions more samples than we will actually need.

To "preprocess" this data and create the training datasets, we then create a q-ary lattice embedding $\mathbf{\Lambda}_i$ of each subsampled $\mathbf{A}_i$ via:

$$\mathbf{\Lambda}_i = \begin{bmatrix} 0 & q \cdot \mathbf{I}_n \\ \omega \cdot \mathbf{I}_m & \mathbf{A}_i \end{bmatrix} \tag{1}$$

Lattice reduction on $\mathbf{\Lambda}_i$ finds a unimodular transformation $[\mathbf{L} \quad \mathbf{R}]$ which minimizes the norms of $[\mathbf{L} \quad \mathbf{R}]\, \mathbf{\Lambda_i} = [\omega \cdot \mathbf{R} \quad \mathbf{RA} + q \cdot \mathbf{L}]$. ω is a scaling parameter that trades-off reduction strength and the error introduced by reduction. This $\mathbf{R}$ matrix is then applied to the original $(\mathbf{A}_i, \mathbf{b}_i)$ to produce reduced samples $(\mathbf{RA}_i, \mathbf{Rb}_i)$ with smaller norms. Repeating this process many times (parallelized across many CPUs) produces a dataset of reduced LWE samples.

To run lattice reduction on Λ_i, we interleave two popular reduction algorithms, BKZ2.0 [10] and flatter [29], following the approach of [23]. After each reduction algorithm completes a step, we run the polishing algorithm of [9]. These algorithms are parameterized by a blocksize β (for BKZ2.0); a reduction strength parameter α (for flatter); algorithm switching parameters γ and s, which represent the minimum amount (γ) by which the reduction must improve over s steps of a particular algorithm, otherwise an algorithm switch is triggered; and a data writing threshold τ. An overview of our reduction approach, which describes the purpose of these parameters, is given in Algorithm 1 and 2.

Algorithm 1 Interleaved lattice reduction

InterleavedReduction($\Lambda_i, \alpha, \beta, \gamma, s, \tau$):

$\rho = \inf$; {set reduction threshold at infinity}
prior_ρ = [];
algo1 = Flatter(α) [29]; a_1 = True; {flatter goes first}
algo2 = BKZ2.0(β) [10]; a_2 = False; {BKZ2.0 goes second}
while $\rho \geq \tau$ **do**
  while a_1 **do**
    Λ_i = polish(algo1(Λ_i));
    a_1, a_2, ρ, prior_ρ = CheckForSwitch(Λ_i, ρ, prior_ρ, s, γ, a_1, a_2);
    if $\rho \leq \tau$ **then**
      break
    end if
  end while
  while a_2 **do**
    Λ_i = polish(algo2(Λ_i));
    $a_1, a_2, \rho,$ prior_ρ = CheckForSwitch(Λ_i, ρ, prior_ρ, s, γ, a_1, a_2);
    if $\rho \leq \tau$ **then**
      break
    end if
  end while
end while
return Λ_i;

Algorithm 2 Criteria for switching reduction algorithms based on reduction progress.

CheckForSwitch($\Lambda_i, \rho,$ prior_ρ, s, γ, a_1, a_2):

stall = False; {Assume we aren't stuck.}
if len(prior_ρ) $> s + 1$ **then**
  decreases = [prior_$\rho[i-1]$ - prior_$\rho[i]$
  for i in range$(-s, 0)$]
  if mean(decreases) $< \gamma$ **then**
    stall = True;
  end if
end if
if stall **then**
  a_1 =!$a1$
  a_2 =!$a2$
end if
prior_ρ.append(ρ);
ρ = ComputeReduction(Λ_i);
return $a_1, a_2, \rho,$ prior_ρ;

3.3 Implementation Details

Lattice reduction is implemented in Python, utilizing the `fpylll` and `flatter` libraries for the BKZ2.0 and flatter implementations[2]. We run lattice reduction on CPUs only (2.1GHz Intel Xeon Gold CPUs with 750 GB of RAM). Table 2 specifies the reduction parameter values used. These were selected after substantial empirical evaluation of different parameter values. We found in general that, for each matrix, the amount of reduction tended to "flatline" after an initial period of significant improvement (see Figure 1), so we optimized parameters to achieve maximal reduction in reasonable time before performance flatlined.

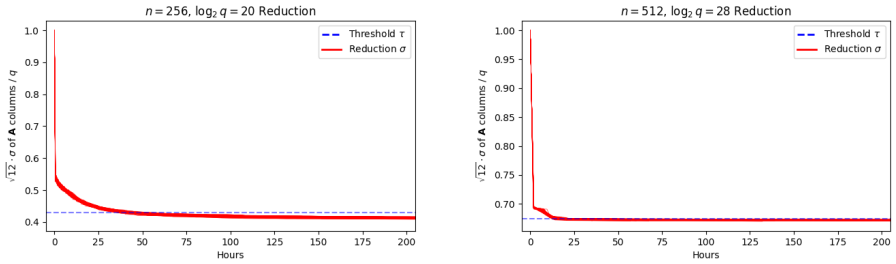

Figure 1: **Reduction over time for $N = 256, \log q = 20$ (left) and $N = 512, \log q = 28$ (right).** *Threshold τ denoted by the dashed blue line. Each red line denotes a separate reduction experiment.*

Table 2: **Reduction parameters.** *The writing threshold τ is customized for each n/q setting based on extensive experiments. Most other parameters are fixed across all reduction experiments.*

Parameter	α	β	γ	s	τ	ω
Setting	0.04	30 if $n \leq 256$ else 18	-0.001	3	Varies by n/q	10

4 Baseline Results

4.1 Data Quality Analysis

We present different statistics for each of the datasets in Table 3. We measure the reduction factor ρ as the ratio of the means of the standard deviations of the rows of $\mathbf{RA}$ and $\mathbf{A}$. A smaller ρ implies that $\mathbf{RA}$ is more reduced. We preprocess each matrix $\mathbf{A}$ until ρ stops decreasing. For each dataset, we report ρ to 4 decimal places. In addition, per [23], the reduction algorithms produce a cliff shape in the standard deviations of the columns of $\mathbf{RA}$ (see Figure 2). Columns with standard deviations greater than $\frac{q}{\sqrt{12}}$ form the "cruel" region in the cliff because secret bits in those column indices are more difficult to recover. In Table 3, we also report the size of the cliff produced by the lattice reduction algorithms.

Table 3: **Statistics on the datasets provided by this work**. *n and q are the LWE dimension and modulus, respectively. m is the number of random indices we select at a time for subsampling. ρ is the stddev reduction, a metric reported in [20], while # cruel bits is the number of unreduced bits, used by [23]. # samples/matrix is slightly less than $m + n$ as we remove the rows with all zeroes. All statistics are calculated on a representative subset of 5000 examples.*

n	$\log_2 q$	m	ρ	# cruel bits	# samples/matrix
256	20	224	0.4284	37	438
512	12	448	0.9036	411	841
512	28	448	0.6740	225	939
512	41	448	0.3992	69	938
1024	26	1624	0.8600	750	2626

[2]https://github.com/facebookresearch/LWE-benchmarking/tree/main/src/generate

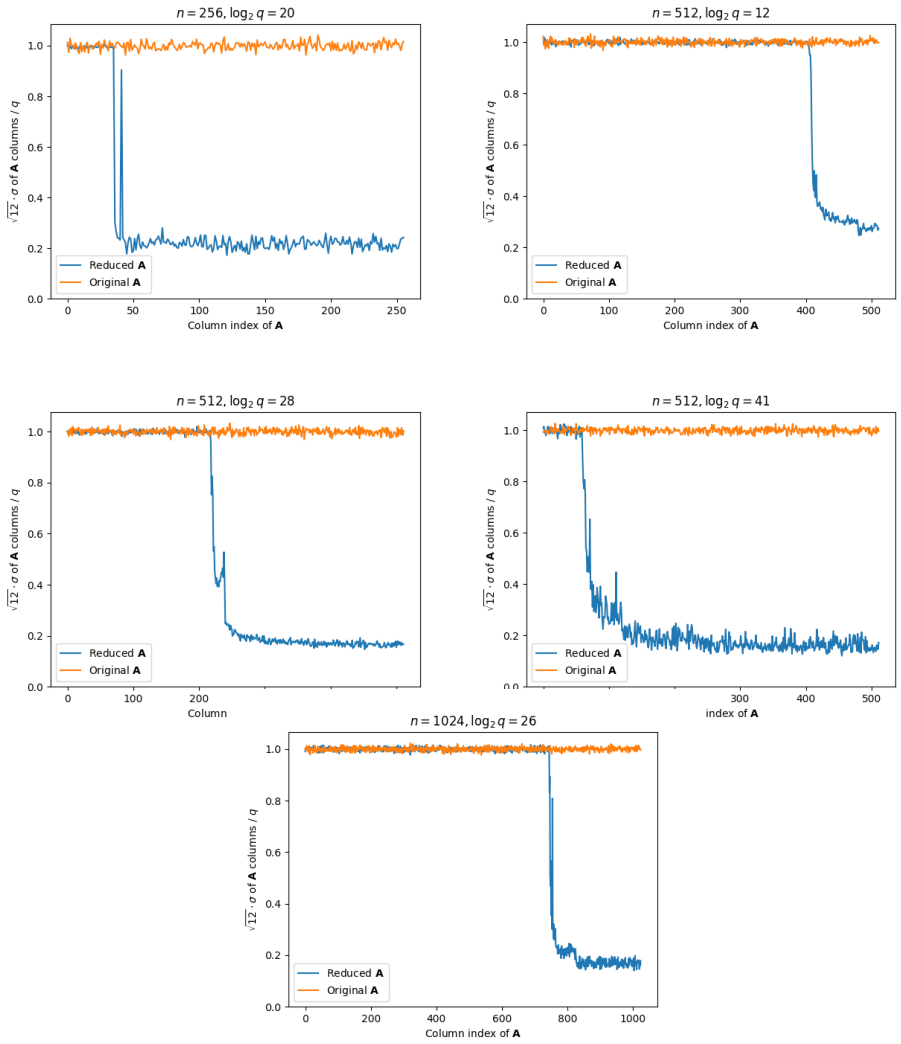

Figure 2: **Cliff shape in our five reduced datasets.** *Datasets that are more reduced have fewer columns of* $\mathbf{A}$ *with a normalized standard deviation of 1.0 (shorter cliff).*

4.2 Computational Cost

In Table 4, we report the number of CPU hours required to process one matrix in each setting. The reduction time depends on the τ threshold and the size of n and m. There is also a tradeoff between BKZ block size and CPU hours and the reduction achieved. With experimentation, we find a block size that balances both. We see that the $n = 256$ dataset has the longest CPU hours per matrix, likely because it has the most aggressive reduction threshold τ relative to the problem difficulty.

Table 4: **Computational cost of generating datasets provided by this work**. *n and q are the LWE dimension and modulus, respectively. # hours/matrix is the average hours needed to reduce a matrix to the given τ. Total CPU hours is hours to reduce a matrix on a single CPU times the number of matrices needed for 400M ($n = 256$) or 40M ($n = 512, 1024$) samples.*

n	$\log_2 q$	# hours/matrix	Total CPU Hours
256	20	44.31	40,486,047
512	12	7.34	349,090
512	28	16.88	664,971
512	41	15.2	647,721
1024	26	20.94	318,958

4.3 Benchmark Results

We run the SALSA and Cool and Cruel attacks—the two existing AI attacks on LWE—to establish baseline performance on the datasets. Both attacks are described briefly in §2, but we refer the interested reader to [36, 23, 34] for additional details. Here we give an overview of our implementation of each attack.

SALSA. We follow the setup of [36], using their open source code[3]. We train encoder-only transformers with 4 layers, embedding dimension 512, and 8 attention heads. We leverage the Adam [18] optimizer with a target learning rate of $l = 1e-4$, 8000 warmup steps, and weight decay of $1e-3$. We adopt the angular embedding technique of [34], which translates input elements of $\mathbf{a} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$ into coordinates on the unit circle. Finally, we use a training batch size of 64 and run the distinguisher algorithm (again, adopted from [36]) on 128 samples. We train the model on a single NVIDIA Titan X GPU for up to 3 days or 1000 epochs.

Cool and Cruel. Again, we follow the setup and use the open source code of [36], which improves upon the initial Cool and Cruel attack of [23]. In particular, we use the linear regression approach innovated by [36] to recover the cool bits, after using the simple brute force approach to recover cruel bits. We use $100,000$ data points for the brute force recovery and the rest for linear regression on the cool bits. We run each attack on a single NVIDIA Titan X GPU.

Results. Drawing on prior results [36], we generate binary secrets with varying Hamming weight and 3 or 4 cruel bits for each dataset, since these are known to be in the recoverable range for existing AI attacks. Then we run both attacks on these secrets. Out of all these experiments, we report the highest Hamming weight secret recovered for each setting in Table 5. These experiments sanity check the quality of the data by showing that sparse secrets can be recovered, and they establish baseline attack performance.

Table 5: **Benchmark results of ML attacks on the datasets provided by this work**. *n and q are the LWE dimension and modulus, respectively. h is the hamming weight of the secret (the number of non-zero elements in the secret vector of size n).*

n	$\log_2 q$	max h recovered (binary)		max h recovered (ternary)	
		SALSA	CC	SALSA	CC
256	20	**33**	22	20	**22**
512	12	**6**	**6**	4	**7**
512	28	9	**12**	11	**13**
512	41	**63**	60	**45**	37
1024	26	5	**6**	4	**5**

5 Related Work

Numerous works have proposed benchmarks for AI performance on mathematical reasoning tasks, which are somewhat related to our work. These include GM8SK [12], a grade school math reasoning benchmark; MATH [17], which includes high-school level math problems; and FrontierMath [15], an even more complex dataset of mathematical reasoning problems. These—and many other—math datasets are used to benchmark mathematical reasoning capabilities of large language models.

TAPAS is distinct from these generic math datasets in several ways. Instead of assessing generic mathematical capabilities, TAPAS datasets determine if models have learned specific mathematical capabilities—namely modular addition and multiplication—necessary for solving LWE. As a result of understanding these math concepts, models trained on our datasets also solve a useful problem: recovering LWE secrets. The dual nature of our datasets sets them apart.

Prior work on AI attacks for LWE [36, 20] released some reduced datasets for training AI models on the LWE problem. However, these datasets only have 4M examples, limiting attack performance. We release much larger datasets to aid investigation of how data scaling affects the model's reasoning

[3] https://github.com/facebookresearch/LWE-benchmarking

capabilities. We also provide data for different parameter settings compared to prior work to enable exploration into how attack performance changes with the LWE parameters.

6 Discussion and Conclusion

This work presents TAPAS, a new collection of datasets to enable further study of AI-powered attacks on the Learning with Errors problem. These datasets are 10x bigger than those provided by prior work, containing at least 40 million reduced examples per LWE setting. For one setting ($n = 256$, $\log_2 q = 20$), we provide 400 million reduced LWE examples. Our benchmark experiments provide a baseline for current state-of-the-art AI-powered attack performance on these datasets, to serve as a starting point for community-wide efforts to improve on these attacks.

We believe these datasets hold immense value for the AI community, beyond the obvious benefits to those assessing the security of post-quantum cryptosystems. They provide an easy on-ramp for AI practitioners to experiment with a new problem domain—using AI models to recover secrets from LWE problems. To outperform current state-of-the-art attacks, model trainers must address complex challenges well-known in the AI community, such as models' difficulty in solving modular arithmetic. Creative solutions to these challenges will enable progress on LWE cryptanalysis specifically, while also advancing the broader field of machine learning.

Limitations. Prior work observed a correlation between the amount by which LWE data is reduced and the complexity (as measured by Hamming weight) of recoverable secrets—more reduced data enables recovery of higher Hamming weight secrets [21]. While the datasets provided in this work are much *larger* and more *diverse* (e.g. more parameter settings) than those previously made available, they are still on par with the reduction quality of prior datasets. Improvements in reduction quality require innovation in lattice reduction techniques, a line of inquiry outside the scope of this paper. Future work could explore whether simply training AI models on more data could mediate the effect of truncated reduction quality, as more learnable patterns may be discerned in larger amounts of data. Furthermore, our datasets could also enable the study of scaling laws for how the amount of LWE data used in AI training affects secret recovery. In addition, generating these datasets required significant computational resources. To reduce this computational burden, future work can explore: data augmentation, synthetic data generation, and efficiency improvements. By making dataset generation more efficient, we can create larger training datasets that enhance AI attacks.

7 Future Work Leveraging Our Datasets

We hope that the datasets provided in this paper catalyze future AI-powered analysis of LWE security. Here are some suggestions for how AI researchers could leverage our datasets to advance the science of AI-enabled cryptanalysis of LWE (and beyond).

Current work on AI cryptanalysis of LWE [36] has only explored the use of transformers in attacking LWE. Future work should consider other model architectures and/or training paradigms (such as reinforcement learning) to improve attacks. Perhaps a fusion approach, in which the pattern mining capabilities of AI models support traditional cryptanalysis techniques, would unlock progress.

Additionally, by treating LWE samples as data rather than as algebraic entities, we can explore statistical correlations in the data and other interesting properties that might yield cryptanalytic insights. [23] already demonstrated this potential by crafting a novel attack based on the *shape* of reduced data that had not previously been observed. Therefore, we encourage other statistical investigation of large-scale LWE datasets to surface other statistical signals.

In this work, we provide much larger training sets than previously reported. This should allow researchers to use larger models, train them for longer, and hopefully compute scaling laws for LWE attacks. Establishing scalings has proven beneficial in many applications of AI.

Publishing reduction matrices (together with the reduced LWE samples allows researchers to work on different secrets. This might allow models to be trained on some secrets and fine-tuned on others, a dramatic improvement to the efficiency of the current SALSA attack (which otherwise has to be rerun, preprocessing included, for every secret).

The reformulation of LWE as a pure arithmetic task (learning modular addition) will help integrate new paradigms from AI for Math, such as results on grokking [16] or modular arithmetic [30].

Finally, existing AI attacks formulate the problem of recovering the LWE secret in the same way: train a model $\mathcal{M}$ to predict b given $\mathbf{a}$ with a fixed secret. However, this is not the only way the problem could be formalized. Researchers could use our datasets to investigate other cryptanalytic tasks of interest, like decision-LWE or distinguishability, by training models to distinguish between the provided (A, b) samples and random samples. Models could also be trained on many different LWE problems with different secrets, or perhaps trained on ($\mathbf{a}$, b) pairs and asked to predict the secret $\mathbf{s}$. Such reformulations of the problem have not been explored and could possibly yield better generalization.

Acknowledgments and Disclosure of Funding

The authors would like to thank Mohamed Malhou for his valuable feedback and insights.

References

[1] Report on the Security of LWE: Improved Dual Lattice Attack., 2023. https://zenodo.org/record/6412487.

[2] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, et al. Homomorphic encryption standard. In *Protecting Privacy through Homomorphic Encryption*. Springer, 2021. https://eprint.iacr.org/2019/939.

[3] Martin R Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In *Proc. of EUROCRYPT*, 2019.

[4] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber (version 3.02) – Submission to round 3 of the NIST post-quantum project. 2021. Available at https://pq-crystals.org/.

[5] Lei Bi, Xianhui Lu, Junjie Luo, Kunpeng Wang, and Zhenfei Zhang. Hybrid dual attack on lwe with arbitrary secrets. Cryptology ePrint Archive, Paper 2021/152, 2021. https://eprint.iacr.org/2021/152.

[6] Jean-Philippe Bossuat, Rosario Cammarota, Jung Hee Cheon, Ilaria Chillotti, et al. Security guidelines for implementing homomorphic encryption. *Cryptology ePrint Archive*, 2024.

[7] Johannes Buchmann, Richard Lindner, and Markus Rückert. Explicit hard instances of the shortest vector problem. In *Post-Quantum Cryptography: Second International Workshop*. Springer, 2008.

[8] Kevin Carrier, Yixin Shen, and Jean-Pierre Tillich. Faster dual lattice attacks by using coding theory. *Cryptology ePrint Archive*, 2022.

[9] François Charton, Kristin Lauter, Cathy Li, and Mark Tygert. An efficient algorithm for integer lattice reduction. *arXiv preprint arXiv:2303.02226*, 2023.

[10] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *Proc. of ASIACRYPT 2011*, 2011.

[11] 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*, 2019.

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[13] Léo Ducas, Marc Stevens, and Wessel van Woerden. Advanced lattice sieving on gpus, with tensor cores. Cryptology ePrint Archive, Paper 2021/141, 2021. https://eprint.iacr.org/2021/141.

[14] Yara Elias, Kristin E. Lauter, Ekin Ozman, and Katherine E. Stange. Provably weak instances of ring-lwe. In *Proc. of CRYPTO*, 2015.

[15] Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*, 2024.

[16] Andrey Gromov. Grokking modular arithmetic, 2023. https://arxiv.org/pdf/2301.02679.pdf.

[17] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

[19] H.W. jr. Lenstra, A.K. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[20] Cathy Li, Emily Wenger, Zeyuan Allen-Zhu, Francois Charton, and Kristin Lauter. SALSA VERDE: a machine learning attack on Learning With Errors with sparse small secrets. In *Proc. of NeurIPS*, 2023.

[21] Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin Lauter. Salsa Picante: A Machine Learning Attack on LWE with Binary Secrets. In *Proc. of ACM CCS*, 2023.

[22] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, page 1468–1480, 2010.

[23] 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. *Proc. of AFRICACRYPT*, 2024.

[24] Theodoros Palamas. Investigating the ability of neural networks to learn simple modular arithmetic. 2017.

[25] Chris Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract. In *Proc. of the Forty-First Annual ACM Symposium on Theory of Computing*, 2009. https://eprint.iacr.org/2008/481.

[26] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *Proc. of STOC*, 2005.

[27] Oded Regev. The learning with errors problem (invited survey). In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 191–204, 2010.

[28] Ronald L Rivest. Cryptography and machine learning. In *International Conference on the Theory and Application of Cryptology*, pages 427–439. Springer, 1991.

[29] Keegan Ryan and Nadia Heninger. Fast practical lattice reduction through iterated compression. *Cryptology ePrint Archive*, 2023.

[30] Eshika Saxena, Alberto Alfarano, Emily Wenger, and Kristin E. Lauter. Making hard problems easier with custom data distributions and loss regularization: A case study in modular arithmetic. In *Forty-second International Conference on Machine Learning*, 2025.

[31] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 1987.

[32] Avital Shafran, Eran Malach, Thomas Ristenpart, Gil Segev, and Stefano Tessaro. Is ML-based cryptanalysis inherently limited? simulating cryptographic adversaries via gradient-based methods. Cryptology ePrint Archive, Paper 2024/1126, 2024.

[33] Peter W Shor et al. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. los alamos physics preprint archive, 1995.

[34] Samuel Stevens, Emily Wenger, Cathy Yuanchen Li, Niklas Nolte, Eshika Saxena, Francois Charton, and Kristin Lauter. Salsa fresca: Angular embeddings and pre-training for ml attacks on learning with errors. https://eprint.iacr.org/2024/150.

[35] Emily Wenger, Mingjie Chen, François Charton, and Kristin E Lauter. Salsa: Attacking lattice cryptography with transformers. *Proc. of NeurIPS*, 2022.

[36] Emily Wenger, Eshika Saxena, Mohamed Malhou, Ellie Thieu, and Kristin Lauter. Benchmarking attacks on learning with errors. *Proc. of Oakland S& P*, 2025.

[37] Wenwen Xia, Leizhang Wang, Dawu Gu, Baocang Wang, et al. Improved Progressive BKZ with Lattice Sieving and a Two-Step Mode for Solving uSVP. *Cryptology ePrint Archive*, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, in the abstract we describe the datasets and results we provide in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: There are no theoretical results in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provide the details on the experiments in Sections 3 and 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we provide the data and code with documentation.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we provide the details on the experiments in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In reporting the benchmark performance of AI attacks on our provided datasets, we provide the number of successful attacks out of number of attempted attacks for each setting. Due to the binary nature of our attack (e.g. success or failure), this is the appropriate level of statistical rigor.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provide details on the compute needed in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Yes, the research conducted in the paper conforms to the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss potential impacts of this work in sections 1, 6, and 7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The datasets we release are synthetically generated integers, so they don't pose any such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited and referenced authors whose work on which we build.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide datasets with details and documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.