




Almost pairwise independence and resilience to deep learning attacks

Rustem Takhanov  

Nazarbayev University, Astana, Kazakhstan

Abstract. Almost pairwise independence (API) is a quantitative property of a class of functions that is desirable in many cryptographic applications. This property is satisfied by Learning with errors (LWE)-mappings and by special Substitution-Permutation Networks (SPN). API block ciphers are known to be resilient to differential and linear cryptanalysis attacks. Recently, security of protocols against neural network-based attacks became a major trend in cryptographic studies. Therefore, it is relevant to study the hardness of learning a target function from an API class of functions by gradient-based methods.

We propose a theoretical analysis based on the study of the variance of the gradient of a general machine learning objective with respect to a random choice of target function from a class. We prove an upper bound and verify that, indeed, such a variance is extremely small for API classes of functions. This implies the resilience of actual LWE-based primitives against deep learning attacks, and to some extent, the security of SPNs. The hardness of learning reveals itself in the form of the barren plateau phenomenon during the training process, or in other words, in a low information content of the gradient about the target function. Yet, we emphasize that our bounds hold for the case of a regular parameterization of a neural network and the gradient may become informative if a class is mildly pairwise independent and a parameterization is non-regular. We demonstrate our theory in experiments on the learnability of LWE mappings.

Keywords: pairwise independence · decorrelation theory · Learning With Errors (LWE) · Substitution-Permutations Networks (SPN) · barren plateau phenomenon · information content of the gradient · hardness of learning

1 Introduction

The gradient-based learning is a paradigm that proved to be highly successful in such diverse areas as language modeling [Ope22], protein folding prediction [JEP⁺21], game playing [SHM⁺16], quantum chemistry [PSMF20] etc. Cryptography is a field that is tightly connected with machine learning (ML), yet ML methods rarely lead to success in this area. This is due to a fundamental difference in goals between these two areas, the goal of cryptography being to design primitives that are hard for learning methods by construction [KV94]. Nonetheless, recently neural network (NN)-based approaches have attracted some attention from cryptographers. This is aligned with a general rise of interest towards using gradient-based methods to tackle problems of combinatorial nature [LCK18, KvHW19, SLB⁺19]. In our paper we find that it is unlikely that such an approach will succeed in a typical cryptographical application. We study the nature of difficulties that a gradient-based method faces when it learns a target function that is sampled from a set of functions that satisfies an almost pairwise independence assumption.

E-mail: rustem.takhanov@nu.edu.kz (Rustem Takhanov)



Pairwise independence is a natural property of collections of functions, \mathcal{H} , between two finite domains, \mathcal{X} and \mathcal{Y} . This property states that for any two distinct elements $x, y \in \mathcal{X}$, and a random function $h \in \mathcal{H}$, an image $[h(x), h(y)]$ is a random variable uniformly distributed on \mathcal{Y}^2 . The notion was introduced in cryptography [CW79, WC81] and has found applications in message authentication [BHK⁺99] and derandomization [LW06]. In a soft version of its definition, we only require that the total variation distance between a distribution of the random variable $[h(x), h(y)]$ and a purely uniform distribution, with an additional averaging over x , is $\mathcal{O}(\epsilon)$ for some negligible parameter ϵ . We call such classes of functions *almost pairwise independent*. Exact definitions can be found in Section 2.

Gradient-based learning is a general term that encompasses all learning algorithms based on the minimization of a certain objective function using access to the approximate gradient of the function at points of interest. The latter formulation includes well-known deep learning optimization methods, such as Stochastic Gradient Descent (SGD), RMSProp, Nesterov Momentum, Adam, etc. A framework that captures such methods was suggested in [Sha18] and it allows to describe the phenomenon of a *low information content of the gradient*. It was noted in [SSS17] that when learning a class of functions containing many nearly uncorrelated or almost orthogonal functions with respect to the data distribution, minimizing the mean squared error loss results in a gradient that exhibits negligible correlation with the target function according to which the dataset was sampled.

Let us briefly describe how this phenomenon can appear in a typical situation. Suppose that the elements of a certain function class (often referred to as a hypothesis set) are parameterized by a parameter k . We assume that the parameter k is chosen randomly, and this choice uniquely defines the target function that we aim to learn. Then, the variance of the gradient of a loss at a given point with respect to a random choice of k measures how sensitive the gradient is to the choice of the target function. If the target function itself depends on k in a highly sensitive way, but the variance is extremely small, then an outcome of a gradient-based optimization with a high probability does not depend on k , and therefore, it is unlikely that it will successfully learn the target. This phenomenon can be rigorously established by proving an upper bound on the variance of the gradient. An archetypical example is a learning problem for the class of orthogonal target functions $\{\sin(kx)\}_{k=1}^K$ defined on the interval $[0, 2\pi]$, where the parameter k is chosen uniformly from the set $\{1, 2, \dots, K\}$ (which has a simple meaning — frequency of the target wave function). If we attempt to approximate the target function $\sin(kx)$ using a neural network $p(\mathbf{w}, x)$ with mean squared loss, then the variance (with respect to the choice of k) of the objective’s gradient behaves like $\mathcal{O}\left(\frac{\int_0^{2\pi} \|\nabla_{\mathbf{w}} p(\mathbf{w}, x)\|^2 dx}{K}\right)$, which vanishes as K increases.

In other words, gradient descent is unable to learn a high-frequency wave if the frequency range is too broad, which is a well-known fact in deep learning research [RBA⁺19]. We see that such an upper bound includes a factor $\int_0^{2\pi} \|\nabla_{\mathbf{w}} p(\mathbf{w}, x)\|^2 dx$ that measures the regularity of the model, i.e., the function set used to fit the data (e.g., a neural network). This factor limits the generality of the framework because it does not rule out the possibility that less regular models might still be able to learn the target.

Notably, an upper bound on the variance with a similar structure can be proven for the training of the class of parities [Sha18], tensor networks [LYDD22] and quantum circuits [MBS⁺18]. In the research community focused on physical applications, this phenomenon is referred to as the “barren plateau”. Practically, the dynamics of the training process in the “barren plateau” case either involve overfitting or exhibit random motion on a flat objective landscape. “Barren plateaus” frequently occur in learning tasks with synthetic datasets (i.e., datasets that are not collected naturally but are generated from some mathematical expression), particularly when the target function involves modular multiplication or a high-frequency function [TTP⁺24].

We use the latter framework in our analysis. The goal of this paper is to establish

an upper bound for cases where the hypothesis set is almost pairwise independent. To demonstrate the strength of our general bound, we consider two special cases of particular importance to cryptography: *Learning with Errors mappings* and *Substitution-Permutation Networks*. The bound can be interpreted as a negative result, indicating that the listed classes are not learnable by any gradient-based algorithm (regardless of the neural network architecture) when training examples are drawn from a uniform distribution over inputs. On the positive side, the bound suggests that a successful gradient-based attack would require preprocessing, aimed at generating a new training set with a highly non-uniform input distribution, as was done in a recent attack [LSW⁺23].

The target hash function $h \in \mathcal{H}$ that is used to generate the dataset $\{(x_i, h(x_i))\}_{i=1}^m$, plays the role of the key parameter k described above. We prove upper bounds on the variance of the gradient (with respect to the randomness in the choice of h) of an objective defined as the expectation of $L(p(\mathbf{w}, X), h(X))$, where L is a loss function, $p(\mathbf{w}, \cdot) : \mathcal{X} \rightarrow \mathbb{R}$ is a parameterized family of functions (e.g., a neural network), and the expectation is taken over X sampled uniformly from \mathcal{X} . In our general bound we prove

the variance w.r.t. h of the objective's gradient is negligibly small,

provided that our neural network $\{p(\mathbf{w}, \cdot)\}$ and the loss function L are regular, and the measure ϵ of pairwise independence of \mathcal{H} is negligibly small (a precise formulation can be found in Theorem 2). An important novelty of our bounds is that they hold not only for special losses (like it was previously proved for the parity problem [SSS17] or periodic functions [Sha18]) but for any loss function L and a regular parameterization of $p(\mathbf{w}, \cdot)$.

As was already mentioned, important examples of almost pairwise independent classes include the Learning with errors mappings and Substitution-Permutation Networks. The **learning with errors** problem (LWE) has an instance (A, \mathbf{b}) where $A \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{b} \in \mathbb{Z}_q^m$. It is assumed that A is generated uniformly from $\mathbb{Z}_q^{m \times n}$ and $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, where \mathbf{s} is a secret key generated uniformly from \mathbb{Z}_q^n and $\mathbf{e} \in \mathbb{Z}_q^m$ is a noise vector whose entries are generated independently according to some fixed distribution χ (usually, χ is a discretized gaussian distribution with a zero mean). The goal of an LWE task is to recover the secret \mathbf{s} from (A, \mathbf{b}) . If $A^\top = [\mathbf{a}_1, \dots, \mathbf{a}_m]$ and $\mathbf{b} = [b_i]_{i=1}^m$, an instance of LWE can be written as a set of pairs $\mathcal{T} = \{(\mathbf{a}_i, b_i)\}_{i=1}^m$ such that $b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i$, $e_i \sim \chi$. The set \mathcal{T} can be understood as a training set for another learning task. Thus, we come to a slightly weaker version of the LWE problem in which the goal is to approximate the function $\mathbf{x} \rightarrow \langle \mathbf{s}, \mathbf{x} \rangle$ on the whole of its domain \mathbb{Z}_q^n , given the training set \mathcal{T} . It is straightforward to reduce LWE to the problem of finding the shortest vector in a lattice (exact-SVP). A famous polynomial quantum reduction of approximate versions of SVP to LWE [Reg05], together with a polynomial classical reduction [BLP⁺13], imply that any polynomial time algorithm for LWE would have extraordinary consequences. Currently, most algorithms for LWE with a polynomial number of samples have an asymptotic running time $2^{\mathcal{O}(n)}$ [HKM18]. Although there is not much hope that a gradient-based approach can solve LWE, it is of practical importance to estimate the maximum size of the problem's instance that can be potentially handled this way [CMLea22, DNGW23]. It is quite straightforward to see that the set of mappings $\mathbf{x} \rightarrow \langle \mathbf{s}, \mathbf{x} \rangle$ (which we call LWE mappings), parameterized by secrets, is an almost pairwise independent class of functions.

Substitution-Permutation Network (SPN) is a form of a block cipher. An SPN is defined as a family of encryption/decryption mappings parameterized by keys. Since the most common modern encryption standard, AES, is a special case of an SPN, it is crucial to understand the potential of NN-based attacks on this type of cipher. Recently it was shown that for some natural choice of parameters, an SPN is an almost pairwise independent class of functions [LTV21]. Thus, the general theory that we build naturally encompasses SPNs as a special case.

Organization. In Section 2 we precisely define the notion of an almost pairwise independent class of functions. Section 3 is dedicated to a description of the general

framework, introduced by Shamir [Sha18], for the gradient-based optimization. Our upper bounds on the variance of the gradient are formulated in Section 4. Subsection 6 specifically deals with the LWE case and subsection 6.1 deals with the case of SPNs. In Section 7 we describe computational experiments with the learnability of LWE mappings and discuss their results in the context of our bounds. Proofs of theorems can be found in the part of the paper that follows the experimental section.

Notations. For any finite multiset S , $|S|$ denotes its cardinality counting multiplicities of elements. $X \sim S$ denotes the fact that the random variable X is sampled from S with probability $\mathbb{P}[X = s] = \frac{m_S(s)}{|S|}$, $s \in S$, where $m_S(s)$ is a multiplicity of s in S . Throughout the paper, q denotes a prime number, n is natural, and $\mathbb{Z}_q = \{0, \dots, q-1\}$ is equipped with an addition, denoted by $+$, and a multiplication, denoted by \cdot (both modulo q). For $\mathbf{x} = [x_i]_1^n, \mathbf{y} = [y_i]_1^n \in \mathbb{Z}_q^n$, $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes $x_1 \cdot y_1 + \dots + x_n \cdot y_n \in \mathbb{Z}_q$. Sometimes, given functions $f, g : \mathbb{Z}_q^n \rightarrow \mathbb{C}$, we will denote the inner product $\sum_{\mathbf{x} \in \mathbb{Z}_q^n} f(\mathbf{x})^\dagger g(\mathbf{x})$ also by $\langle f, g \rangle$. The normalized version, i.e. $\mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [f(\mathbf{x})^\dagger g(\mathbf{x})]$, is denoted by $\langle f, g \rangle_{\mathbf{x}}$. Correspondingly, $\|f\|_{\mathbf{x}} = \sqrt{\langle f, f \rangle_{\mathbf{x}}}$. For a function $f : \mathbb{Z}_q \rightarrow \mathbb{C}$, \hat{f} denotes the discrete Fourier transform of f , i.e. $\hat{f}(\omega) = \sum_{x \in \mathbb{Z}_q} f(x) e^{-\frac{2\pi x \omega i}{q}}$. For a real $r \in \mathbb{R}$, $\{r\}$ denotes its fractional part, and $\lceil r \rceil$ denotes the smallest integer that is greater or equal to r . For $S \subseteq D$, $\mathbf{1}_S : D \rightarrow \{0, 1\}$ denotes an indicator function of a set S (the domain D will be clear from a context). Given $f : U \rightarrow \mathbb{R}$ and $g : U \rightarrow \mathbb{R}_+$, we write $f \lesssim g$ if there exist a universal constant $\alpha \in \mathbb{R}_+$ such that we have $|f(x)| \leq \alpha g(x), x \in U$. For $x, y \in \mathbb{R}$, $x \vee y$ and $x \wedge y$ denote $\max(x, y)$ and $\min(x, y)$ correspondingly.

1.1 Related work

Convex optimization and SQ-theory. If an objective function is the expectation of a random convex function (i.e. of the form $\mathbb{E}_w[f(x, w)]$), then it was shown by [FGV17] that, provided some technical requirements, such a learning algorithm belongs to the class of the so-called statistical query (SQ) algorithms [Koa93]. According to the theory of SQ learning [BFJ⁺94], given any concept class \mathcal{C} (i.e. any class of $\{-1, 1\}$ -valued functions), a key parameter that defines the hardness of learning \mathcal{C} by an SQ-algorithm is the so-called statistical query dimension of \mathcal{C} , which is the maximum number of “nearly uncorrelated” (relative to a data distribution) functions in \mathcal{C} . Based on this idea it was proved by [Yan01] that for a uniform distribution over \mathbb{Z}_q^n and any function $\psi : \mathbb{Z}_q \rightarrow \{-1, 1\}$ such that $\mathbb{E}_{x \sim \mathbb{Z}_q} [\psi(x)] \in [-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$, an SQ-algorithm that learns the concept class $\mathcal{C} = \{c : \mathbb{Z}_q^n \rightarrow \{-1, 1\} \mid c(\mathbf{x}) = \psi(\langle \mathbf{a}, \mathbf{x} \rangle), \mathbf{a} \in \mathbb{Z}_q^n\}$ substantially better than the random guess, requires the running time $\mathcal{O}(q^{\frac{n-1}{2}})$. This result, together with the previously mentioned findings, implies that any meaningful concept class derived from LWE mappings, i.e., mappings $\mathbf{x} \rightarrow \langle \mathbf{a}, \mathbf{x} \rangle$, is hard to learn using gradient-based convex optimization algorithms. Since most modern deep learning algorithms optimize non-convex objectives, we cannot directly apply the latter fact to attacks on LWE that we are interested in.

Decorrelation theory of block ciphers. Almost pairwise independence and t -wise independence were identified as desirable properties in the context of security of block ciphers. It was shown that almost pairwise independence implies resilience to both (truncated) differential and (multidimensional) linear cryptanalysis attacks [BBV15]. Analogously, nearly t -wise independence implies resilience to differential attacks of order $\log_2(t)$. The definition of almost pairwise independence given by Vaudenay is equivalent to ours, though in [Vau03] other metrics (different from the total variation distance) measuring the deviation from the uniform distribution are also considered. Liu et al [LTV21] showed that SPNs, under certain conditions, are almost pairwise independent, the result that we use in our applications. Other examples of constructions of nearly t -wise independent permutations include [HMMR05, AGM03, FPY15, KNR09].

NN-based approaches to attack LWE. A concrete way to use neural networks for side-channel attacks on the Learning with rounding-based cryptographic schemes was demonstrated in [NDJ23]. Also, a recursive learning method was applied to train neural networks for recovering message bits in CRYSTALS-Kyber [DNGW23], which is an LWE-based set of cryptographic primitives [BDK⁺18]. Direct attacks on LWE include SALSA [WCCL22], PICANTE [LSW⁺23] and SALSA VERDE [LWAZ⁺23]. The key idea of the latter three papers is first to preprocess an instance of LWE using a reduction to SVP and lattice reduction techniques (such as BKZ [CN11]) to obtain a new instance of LWE with a smaller coordinate variance. Afterward, a new set of input-output pairs is fed to a gradient-based training algorithm with a transformer architecture. Overall, attacks on other ciphers based on approximation of encryption or decryption functions by some deep learning architectures is quite a popular topic of research [AKJM21, CY21, TTJ23]. Deep learning-based side-channel attacks are another popular topic in recent research.

NN-based approaches to attack block ciphers. Experimental works on learning an encryption/decryption mapping for various block ciphers include [BK20, Ala12, KEI⁺22, ITYY21]. While NN-based approaches have succeeded in tackling round-reduced DES or classical ciphers, results even for one round AES have been negative.

2 Almost pairwise independent families of hash functions

Let \mathcal{X} and \mathcal{Y} be two finite sets. A finite parameterized family $\{h_k\}_{k \in \mathcal{K}} \subseteq \mathcal{Y}^{\mathcal{X}}$ is called pairwise independent if for any distinct $x, x' \in \mathcal{X}$ and any $y, y' \in \mathcal{Y}$ we have $\mathbb{P}_{k \sim \mathcal{K}}[h_k(x) = y, h_k(x') = y'] = |\mathcal{Y}|^{-2}$. Such families are actively used in message authentication and universal hashing [CW79, WC81]. Since it is possible that $h_k = h_{k'}$ for some pair of distinct keys $k, k' \in \mathcal{K}$, the family $\{h_k\}_{k \in \mathcal{K}}$ can be also treated as a multiset. A classical example of a pairwise independent family is the set $\mathcal{H} = \{h_{a,b} : \text{GF}(p^n) \rightarrow \text{GF}(p^n) \mid h_{a,b}(x) = ax + b, a, b \in \text{GF}(p^n)\}$, where $\text{GF}(p^n)$ is the Galois field with p^n elements. Other examples can be found in [Sho05].

For a general hypothesis class \mathcal{H} , that is a multiset of functions from $\mathcal{Y}^{\mathcal{X}}$, let us introduce

$$\varepsilon(x, x') = \sum_{y, y' \in \mathcal{Y}} |\mathbb{P}_{h \sim \mathcal{H}}[h(x) = y, h(x') = y'] - |\mathcal{Y}|^{-2}|, \quad (1)$$

if $x \neq x'$ and $\varepsilon(x, x) = \sum_{y \in \mathcal{Y}} |\mathbb{P}_{h \sim \mathcal{H}}[h(x) = y] - |\mathcal{Y}|^{-1}|$. If \mathcal{H} is pairwise independent, then $\varepsilon(x, x') = 0$ for distinct $x, x' \in \mathcal{X}$. Thus, $\varepsilon(x, x')$ measures the deviation of our hypothesis set from the pairwise independence. Note that $\varepsilon(x, x')$ is double the total variance distance between random variables $[h(x), h(x')]$ for $h \sim \mathcal{H}$ and $[Y, Y']$ for $Y, Y' \sim^{\text{iid}} \mathcal{Y}$ (where i.i.d. means independent and identically distributed).

We will measure the pairwise independence of \mathcal{H} by the following parameter

$$\epsilon = \max_x \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2}. \quad (2)$$

Remark 1. The situation where $\varepsilon(x, x')$ is small is quite common in cryptographic applications. Examples of such classes include the LWE mapping and SPNs. The latter two cases are carefully treated in subsections 6 and 6.1. In the theory of cryptographic hash functions the family \mathcal{H} is called δ -variationally universal, if $\mathbb{P}_{h \sim \mathcal{H}}[h(x) = y] = |\mathcal{Y}|^{-1}, x \in \mathcal{X}, y \in \mathcal{Y}$ and for any distinct $x, x' \in \mathcal{X}$ we have $\varepsilon(x, x') \leq \frac{2\delta}{|\mathcal{Y}|}$. Some properties and constructions of such families can be found in [KR06, Sho05].

Note that $\tilde{\varepsilon} = \max_{x, x'} \varepsilon(x, x')$ coincides with the ∞ -distance between the random function from \mathcal{H} and the so-called perfect cipher (i.e. the uniformly random mapping

from \mathcal{X} to \mathcal{Y}) in the decorrelation theory of [Vau03]. Vaudenay showed that successful differential and linear cryptanalysis attacks on a block cipher \mathcal{H} should have complexities at least proportional to $\frac{1}{\varepsilon}$ and $\frac{1}{\varepsilon^{1/3}}$ respectively. Thus, such attacks will fail for a negligibly small value of ε . This is in line with our result, i.e. the hardness of such ciphers against attacks based on gradient methods. Though the parameter that we use, $\epsilon = \max_x \mathbb{E}_{X' \sim \mathcal{X}}[\varepsilon(x, X')]^{1/2}$, is smaller than ε . \square

3 General optimization framework

Let \mathcal{H} be a multiset. Let $h : \mathcal{X} \rightarrow \mathcal{Y} \in \mathcal{H}$ be a secret mapping. We are given access to samples $(x, h(x))$ where x is generated uniformly randomly from \mathcal{X} . Our goal is to predict an output of the mapping $h(x)$ on other inputs. Ideally, given an input x , we are interested in recovering the whole output $h(x)$. Sometimes we could be interested in predicting some properties of the output, e.g. $t(h(x))$ where $t : \mathcal{Y} \rightarrow \mathbb{R}$ is a fixed function from \mathcal{Y} to some finite subset of \mathbb{R} . For example, if $\mathcal{Y} \subset \mathbb{Z}$ and our goal is to learn a parity bit of $h(x)$, then $t(x) = (-1)^x$.

To approximate the mapping $x \rightarrow h(x)$ (or, $x \rightarrow t(h(x))$), suppose that we fixed a family of functions from \mathcal{X} to \mathbb{R} parameterized by a weight vector $\mathbf{w} \in O \subseteq \mathbb{R}^{N_{\text{par}}}$ (N_{par} denotes the number of parameters), i.e. $\{p(\mathbf{w}, \cdot) : \mathcal{X} \rightarrow \mathbb{R} \mid \mathbf{w} \in O\}$. We assume that p is continuous and differentiable w.r.t. \mathbf{w} in almost all points of an open set O . We require O to be open in order to differentiate p with respect to \mathbf{w} without concerning ourselves with the definition of the derivative on the boundary of O . In practice, such a family is usually defined as a neural network architecture with an input x encoded as a binary vector.

Our task is to minimize over \mathbf{w} the following objective

$$C_h(\mathbf{w}) = \mathbb{E}_{x \sim \mathcal{X}} [L(p(\mathbf{w}, x), h(x))], \quad (3)$$

where $L : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a fixed loss function. We make only the most general assumptions on the form of L such as continuity and the existence of a partial derivative w.r.t. the first variable in almost all points, i.e. $\frac{\partial L(p, y)}{\partial p}$. Without this assumption, we would not be able to define the gradient of $C_h(\mathbf{w})$, which is why it is not a restricting requirement. Note that $\nabla_{\mathbf{w}} C_h(\mathbf{w}) = \mathbb{E}_{x \sim \mathcal{X}} [\frac{\partial L(p(\mathbf{w}, x), h(x))}{\partial p} \nabla_{\mathbf{w}} p(\mathbf{w}, x)]$, i.e. the size of the gradient vector is controlled by size of the vector $\nabla_{\mathbf{w}} p(\mathbf{w}, x)$ and the scalar $\frac{\partial L}{\partial p}(p(\mathbf{w}, x), h(x))$.

We assume that the minimization of the cost (3) is to be solved by a gradient-based method. By the latter we understand any algorithm that iteratively computes points $\mathbf{w}_1, \mathbf{w}_2, \dots \in O$ in such a way that \mathbf{w}_{t+1} depends on, possibly, all previously computed gradient approximations and an approximation of the gradient $\nabla_{\mathbf{w}} C_h(\mathbf{w}_t)$. An approximation of $\nabla_{\mathbf{w}} C_h(\mathbf{w}_t)$, denoted by \mathbf{g}_t , is requested from an oracle \mathfrak{D} . Due to the stochastic nature of \mathbf{g}_t , we assume that $\|\mathbf{g}_t - \nabla_{\mathbf{w}} C_h(\mathbf{w}_t)\| < \delta$ and nothing more can be guaranteed beyond that accuracy δ . That is why the oracle is called the δ -accurate gradient oracle. This makes $\delta > 0$ an important parameter of this optimization framework. In practice \mathbf{g}_t is computed from the dataset, i.e. a set of random pairs $\{(x_i, h(x_i))\}$ to which we have access.

The formalism for gradient-based algorithms described above was given in [Sha18]. Shamir applied this formalism to demonstrate the inability of such algorithms to learn functions of the form $\mathbf{x} \rightarrow \psi(\mathbf{w}^\top \mathbf{x})$ defined on \mathbb{R}^n where ψ is 1-periodic. Our goal is to apply this formalism to the function $x \rightarrow h(x)$ defined on \mathcal{X} , therefore, we formulate adapted Theorem 4 from [Sha18] in the following way.

Theorem 1 ([Sha18]). *Let $\delta > 0$ be such that $\text{Var}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})] \leq \delta^3$ for any $\mathbf{w} \in O$. Then there exists a δ -accurate gradient oracle such that for any algorithm as above and any $p \in (0, 1)$, with probability $1 - p$ over the uniform choice of the function h , the algorithm's output after at most $\frac{p}{\delta}$ iterations will be independent of h .*

For completeness, let us give a proof of this Theorem.

Proof. Let us denote $\mathbb{E}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})]$ by \mathbf{g} . For a given point $\mathbf{w} \in \mathcal{O}$, let us consider a gradient oracle that outputs \mathbf{g} if $\|\nabla C_h(\mathbf{w}) - \mathbf{g}\| \leq \delta$ and $\nabla C_h(\mathbf{w})$ if otherwise. By Chebyshev's inequality, we have

$$\mathbb{P}_{h \sim \mathcal{H}}[\|\nabla C_h(\mathbf{w}) - \mathbf{g}\| > \delta] < \frac{\text{Var}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})]^3}{\delta^2} \leq \delta.$$

Therefore, if at a current iteration t the point \mathbf{w}_t is chosen independently from the key \mathbf{s} , then the next point \mathbf{w}_{t+1} will be chosen independently from \mathbf{k} with probability at least $1 - \delta$. Therefore, after T iterations, with probability at least $1 - T\delta$, all points $\mathbf{w}_1, \dots, \mathbf{w}_T$ will not depend on \mathbf{k} , which completes the proof. \square

For an API class \mathcal{H} , for any fixed $x \in \mathcal{X}$, the value of $h(x)$ depends in a very sensitive way on the parameter h . This makes an output of the gradient-based learning process that is independent of h very undesirable. Therefore, if an accuracy $\delta > \text{Var}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})]^{\frac{1}{3}}$, then we need the number of iterations at the scale of $\mathcal{O}(\text{Var}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})]^{-\frac{1}{3}})$ to succeed in our task. But if $\text{Var}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})]$ is negligibly small (as we will show is the case), e.g. at the scale of 10^{-60} , then our algorithm cannot succeed in principle.

Let us now describe our upper bounds on $\text{Var}_{h \sim \mathcal{H}}[\nabla C_h(\mathbf{w})]$ and conditions under which they hold.

4 Upper bounds on the variance (main result)

For the loss function L , we denote

$$r_{\mathbf{w}}(x, y) = \frac{\partial L(p(\mathbf{w}, x), y)}{\partial p} - \mathbb{E}_{Y \sim \mathcal{Y}}\left[\frac{\partial L(p(\mathbf{w}, x), Y)}{\partial p}\right].$$

Let us also introduce quantities measuring a typical deviation of $\frac{\partial L(p(\mathbf{w}, x), Y)}{\partial p}$ from its mean. Let

$$D_x = \text{Var}_{Y \sim \mathcal{Y}}\left[\frac{\partial L(p(\mathbf{w}, x), Y)}{\partial p}\right] \tag{4}$$

and

$$M_x = \max_{y \in \mathcal{Y}} |r_{\mathbf{w}}(x, y)|. \tag{5}$$

In the following theorem we give a general upper bound on the variance of the loss function for a random choice of a hypothesis h . For $\mathbf{w} = [w_i]_{i=1}^{N_{\text{par}}}$ we denote $\frac{\partial f(\mathbf{w}, z)}{\partial w_i}$ by $\partial_{w_i} f(\mathbf{w}, z)$. Note that the total variance of the gradient is a sum of variances of the objective's partial derivatives w.r.t. every parameter.

Theorem 2 (Main). *Let $\mathcal{O} \subseteq \mathbb{R}^{N_{\text{par}}}$ and $p : \mathcal{O} \times \mathcal{X} \rightarrow \mathbb{R}$ be a mapping¹ such that $\mathbb{E}_{X \sim \mathcal{X}}[(\partial_{w_i} p(\mathbf{w}, X))^2]$ is bounded uniformly over $\mathbf{w} \in \mathcal{O}$. Then, for any $i \in \{1, \dots, N_{\text{par}}\}$, we have*

$$\begin{aligned} \text{Var}_{h \sim \mathcal{H}}[\partial_{w_i} \mathbb{E}_{X \sim \mathcal{X}} L(p(\mathbf{w}, X), h(X))] &\lesssim \mathbb{E}_{X \sim \mathcal{X}}[(\partial_{w_i} p(\mathbf{w}, X))^2] \times \\ &(\mathbb{E}_{X \sim \mathcal{X}}[M_X^4])^{1/2} \cdot \epsilon + \gamma \wedge (|\mathcal{Y}| \cdot (\mathbb{E}_{X \sim \mathcal{X}}[D_X^2])^{1/2} \cdot \epsilon + \gamma), \end{aligned} \tag{6}$$

¹E.g., a neural network. The parameter N_{par} is the number of trained parameters of the set of functions $\{p(\mathbf{w}, \cdot)\}_{\mathbf{w} \in \mathcal{O}}$.

where

$$\epsilon = \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2}$$

and

$$\gamma = \frac{\mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2}}{|\mathcal{X}|^{1/2}}.$$

Remark 2. If the loss function L satisfies the Lipschitz condition, i.e. $|L(p, y) - L(p', y)| \leq c|p - p'|$, then $M_x \lesssim 1$, $D_x \lesssim 1$, and $\text{Var}_{h \sim \mathcal{H}} [\nabla C_h(\mathbf{w})]$ is

$$\lesssim \mathbb{E}_{X \sim \mathcal{X}} [(\partial_{w_i} p(\mathbf{w}, X))^2] \times \left(\max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \frac{1}{|\mathcal{X}|^{1/2}} \right).$$

Let us assume that $|\mathcal{X}|^{-1/2}$ is small. For API families \mathcal{H} (as we have in some cryptographic applications), $\epsilon = \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2}$ is also small. If the latter values are negligibly small, e.g. smaller than 10^{-100} , then the gradient is uninformative unless the factor $\mathbb{E}_{X \sim \mathcal{X}} [(\partial_{w_i} p(\mathbf{w}, X))^2]$ exceeds by many orders of magnitude the largest float number supported by modern hardware. Even if we used specialized software that supports such large values, a stochastic gradient that approximates the real gradient with the needed precision would require an enormous batch size. In any case, an NN whose derivative w.r.t. w_i blows up to such magnitudes is definitely beyond the current paradigm of deep learning.

A more tractable case is when $\epsilon = \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2}$ is moderately small, e.g. like 10^{-10} . This suggests that to make the gradient more informative, one needs to define an NN architecture in such a way that the average $\partial_{w_i} p(\mathbf{w}, \mathbf{x})$ is allowed to become large. Following this idea, we experimented with non-lipschitz activation functions. Results are given in Section 7. As these experiments demonstrate, to a certain extent, this strategy leads to a higher learning capability, although then we run into an exploding gradient problem (where $\nabla_{\mathbf{w}} p(\mathbf{w}, \mathbf{x})$ explodes, but the total gradient $\nabla C_h(\mathbf{w})$ remains small) and various computational instabilities during the training process. \square

To demonstrate the applicability of the previous theorem, let us now consider specific cases of hypothesis sets. We will thoroughly study the case of the LWE hypothesis set, which is defined by

$$\mathcal{H} = \{h_{\mathbf{k}} : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q \mid \mathbf{k} \in \mathbb{Z}_q^n, h_{\mathbf{k}}(\mathbf{x}) = \langle \mathbf{k}, \mathbf{x} \rangle\}, \quad (7)$$

where $q \geq 2$ is a prime number and $\langle \mathbf{k}, \mathbf{x} \rangle = k_1 x_1 + \dots + k_n x_n \bmod q$. After we give some estimates on $\varepsilon(x, y)$ for the LWE hypothesis set, the following statement is a direct consequence of Theorem 2.

Corollary 1. *Let $\mathcal{X} = \mathbb{Z}_q^n$, $\mathcal{Y} = \mathbb{Z}_q$. Let $O \subseteq \mathbb{R}^{N_{\text{par}}}$ and $p : O \times \mathcal{X} \rightarrow \mathbb{R}$ satisfy conditions of Theorem 2. For the hypothesis set defined in (7), we have*

$$\text{Var}_{h \sim \mathcal{H}} [\partial_{w_i} \mathbb{E}_{X \sim \mathcal{X}} L(p(\mathbf{w}, X), h(X))] \lesssim \mathbb{E}_{X \sim \mathcal{X}} [(\partial_{w_i} p(\mathbf{w}, X))^2] \times (\mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} \wedge q \mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2}) q^{-\frac{n-1}{2}}. \quad (8)$$

Remark 3. In a typical post-quantum cryptographic protocol that uses the LWE hypothesis set we have $\log_2 q \approx 10$ and $n \approx 544$ [CKLS18]. Then, the factor $q^{-\frac{n-1}{2}} \sim 10^{-817.3}$ in the RHS of the inequality (8) is an extremely small value. This guarantees that to make the gradient informative, one has to choose an NN architecture and a loss function in such a way that either the average $(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2$ or the average $\text{Var}_{Y \sim \mathbb{Z}_q} [\frac{\partial L(p(\mathbf{w}, \mathbf{x}), Y)}{\partial p}]^2$ blows up. Due to arguments from the previous remark, such a learning process is infeasible. \square

Due to its simplicity, let us first give a proof of Corollary 1, assuming that Theorem 2 is true.

Proof of Corollary 1. Recall that $\mathcal{X} = \mathbb{Z}_q^n$, $\mathcal{Y} = \mathbb{Z}_q$ and

$$\mathcal{H} = \{h_{\mathbf{k}} : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q \mid \mathbf{k} \in \mathbb{Z}_q^n, h_{\mathbf{k}}(\mathbf{x}) = \langle \mathbf{k}, \mathbf{x} \rangle\}.$$

By construction, we have $\frac{|\{\mathbf{k} \in \mathbb{Z}_q^n \mid \langle \mathbf{k}, \mathbf{x} \rangle = y, \langle \mathbf{k}, \mathbf{x}' \rangle = y'\}|}{q^n} = q^{-2}$ and $\varepsilon(\mathbf{x}, \mathbf{x}') = 0$ for linearly independent \mathbf{x}, \mathbf{x}' . If $\mathbf{x}' = \lambda \mathbf{x}$ and $\lambda \notin \{0, 1\}$, $\mathbf{x} \neq \mathbf{0}$, then $\frac{|\{\mathbf{k} \in \mathbb{Z}_q^n \mid \langle \mathbf{k}, \mathbf{x} \rangle = y, \lambda \langle \mathbf{k}, \mathbf{x} \rangle = y'\}|}{q^n}$ equals $q^{-1}[y' = \lambda y]$, i.e. $\varepsilon(\mathbf{x}, \mathbf{x}') = 2(1 - q^{-1})$. If exactly one of \mathbf{x}, \mathbf{x}' is zero, then $\varepsilon(\mathbf{x}, \mathbf{x}') = 2(1 - q^{-1})$. To summarise, we have

$$\varepsilon(\mathbf{x}, \mathbf{x}') = \begin{cases} 0, & \text{if } \text{rank}([\mathbf{x}, \mathbf{x}']) = 2; \\ 2(1 - q^{-1}), & \text{if } \text{rank}([\mathbf{x}, \mathbf{x}']) = 1, \mathbf{x} \neq \mathbf{x}'; \\ 0, & \text{if } \mathbf{x} = \mathbf{x}' \neq \mathbf{0}; \\ 2(1 - q^{-1}), & \text{if } \mathbf{x} = \mathbf{x}' = \mathbf{0}. \end{cases}$$

Using the latter equation, we bound

$$\max_{x \in \mathcal{X}} \mathbb{E}_{X' \in \mathcal{X}} [\varepsilon(x, X')^2] \leq 4(1 - q^{-1})^2 q^{-(n-1)} \leq 4q^{-(n-1)}.$$

Thus, using Theorem 2 we obtain a major bound on the variance,

$$\begin{aligned} \text{Var}_{h \sim \mathcal{H}} [\partial_{w_i} \mathbb{E}_{X \sim \mathcal{X}} L(p(\mathbf{w}, X), h(X))] &\lesssim \mathbb{E}_{X \sim \mathcal{X}} [(\partial_{w_i} p(\mathbf{w}, X))^2] \times \\ &(\mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} \wedge q \mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2}) q^{-\frac{n-1}{2}}. \end{aligned}$$

□

5 Proof of Theorem 2

Recall that $r_{\mathbf{w}}(x, y) = \frac{\partial L(p(\mathbf{w}, x), y)}{\partial p} - \mathbb{E}_{Y \sim \mathcal{Y}} [\frac{\partial L(p(\mathbf{w}, x), Y)}{\partial p}]$. We have

$$\begin{aligned} &\text{Var}_{h \sim \mathcal{H}} [\partial_{w_i} \mathbb{E}_{X \sim \mathcal{X}} L(p(\mathbf{w}, X), h(X))] = \\ &\text{Var}_{h \sim \mathcal{H}} [\mathbb{E}_{X \sim \mathcal{X}} [\frac{\partial L(p(\mathbf{w}, X), h(X))}{\partial p} \partial_{w_i} p(\mathbf{w}, X)]] = \\ &\text{Var}_{h \sim \mathcal{H}} [\mathbb{E}_{X \sim \mathcal{X}} [r_{\mathbf{w}}(X, h(X)) \partial_{w_i} p(\mathbf{w}, X)]] \leq \mathbb{E}_{h \sim \mathcal{H}} [(\partial_{w_i} p(\mathbf{w}, x), r_{\mathbf{w}}(x, h(x)))_x^2]. \end{aligned}$$

Further we will use the following result which is a generalization of the classical Bessel's inequality.

Proposition 1 (Boas-Bellman inequality [Dra04]). *If x, y_1, \dots, y_d are elements of an inner product space $(H; \langle \cdot, \cdot \rangle)$, then the following inequality*

$$\sum_{i=1}^d |(x, y_i)|^2 \leq (x, x) \left[\max_{1 \leq i \leq d} (y_i, y_i) + \left(\sum_{1 \leq i \neq j \leq d} |(y_i, y_j)|^2 \right)^{1/2} \right]$$

holds.

Using the Boas-Bellman inequality we obtain

$$\begin{aligned} |\mathcal{H}|^{-1} \sum_{h \in \mathcal{H}} \langle r_{\mathbf{w}}(x, h(x)), \partial_{w_i} p(\mathbf{w}, x) \rangle_x^2 &\leq |\mathcal{H}|^{-1} \|\partial_{w_i} p(\mathbf{w}, x)\|_x^2 \left[\max_{h \in \mathcal{H}} \|r_{\mathbf{w}}(x, h(x))\|_x^2 + \right. \\ &\left. \left(\sum_{h_1 \neq h_2 \in \mathcal{H}} \langle r_{\mathbf{w}}(x, h_1(x)), r_{\mathbf{w}}(x, h_2(x)) \rangle_x^2 \right)^{1/2} \right]. \end{aligned} \quad (9)$$

The second term inside the latter square brackets usually dominates the first one. Our key tool for bounding both terms is the following lemma (whose proof can be found in the next subsection).

Lemma 1. *Let the function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ be such that $\sum_{y \in \mathcal{Y}} f(x, y) = 0$ for any $x \in \mathcal{X}$ and $g_h(x) = f(x, h(x))$. Then, we have*

$$\begin{aligned} \sqrt{\sum_{h_1 \in \mathcal{H}} \sum_{h_2 \in \mathcal{H}} \langle g_{h_1}, g_{h_2} \rangle^2} &\leq |\mathcal{H}| |\mathcal{X}| (\mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \gamma) \wedge \\ &|\mathcal{H}| |\mathcal{X}| |\mathcal{Y}| (\mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \gamma). \end{aligned} \quad (10)$$

where $M_x = \max_{y \in \mathcal{Y}} |f(x, y)|$, $D_x = \text{Var}_{Y \sim \mathcal{Y}}(f(x, Y))$ and

$$\gamma = \frac{\mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2}}{|\mathcal{X}|^{1/2}}. \quad (11)$$

From Lemma 1, after setting f as $r_{\mathbf{w}}$, we obtain

$$\begin{aligned} &\left(\sum_{h_1, h_2} \langle r_{\mathbf{w}}(x, h_1(x)), r_{\mathbf{w}}(x, h_2(x)) \rangle^2 \right)^{1/2} \leq \\ &|\mathcal{H}| |\mathcal{X}| (\mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \gamma) \wedge \\ &|\mathcal{H}| |\mathcal{X}| |\mathcal{Y}| (\mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \gamma), \end{aligned}$$

where γ is defined as in equation (11).

Also, note that

$$\max_{h \in \mathcal{H}} \|r_{\mathbf{w}}(x, h(x))\|_x^2 \leq \left(\sum_{h_1, h_2} \langle r_{\mathbf{w}}(x, h_1(x)), r_{\mathbf{w}}(x, h_2(x)) \rangle_x^2 \right)^{1/2}.$$

After we plug in the latter two inequalities into the bound (9), we obtain the following fact:

$$\begin{aligned} \text{Var}_{h \sim \mathcal{H}} [\partial_{w_i} \mathbb{E}_{x \sim \mathcal{X}} L(p(\mathbf{w}, x), h(x))] &\lesssim \|\partial_{w_i} p(\mathbf{w}, x)\|_x^2 \times \\ &(\mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \gamma) \wedge \\ &|\mathcal{Y}| (\mathbb{E}_{X \sim \mathcal{X}} [D_X^2]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}} [\varepsilon(x, X')^2]^{1/2} + \gamma), \end{aligned}$$

which is the statement of our theorem.

5.1 Proof of Lemma 1

To complete the proof of theorem 2 we need to prove Lemma 1. The following lemma is instrumental in that proof. Let $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ be some function. For $h \in \mathcal{H}$, a function $g_h : \mathcal{X} \rightarrow \mathbb{R}$ is defined by $g_h(x) = f(x, h(x))$.

Lemma 2. *Let $\Phi = [f(x, h(x))]_{(h,x) \in \mathcal{H} \times \mathcal{X}}$ and $\mathbf{F} = \Phi^\top \Phi$. Then,*

$$\sum_{h_1 \in \mathcal{H}} \sum_{h_2 \in \mathcal{H}} \langle g_{h_1}, g_{h_2} \rangle^2 = \|\Phi\|_4^4 = \|\mathbf{F}\|_F^2,$$

where $\|\cdot\|_p$ denotes Schatten p -norm.

Proof. Let us denote the vector $[f(x, h(x))]_{x \in \mathcal{X}}$ by \mathbf{f}_h . A direct calculation gives us

$$\begin{aligned} \sum_{h_2 \in \mathcal{H}} \langle g_{h_1}, g_{h_2} \rangle^2 &= \sum_{h_2 \in \mathcal{H}} \left(\sum_{x \in \mathcal{X}} f(x, h_1(x)) f(x, h_2(x)) \right)^2 = \\ &= \left\| \left[\sum_{x \in \mathcal{X}} f(x, h_2(x)) f(x, h_1(x)) \right]_{h_2 \in \mathcal{H}} \right\|^2 = \|\Phi \mathbf{f}_{h_1}\|^2. \end{aligned}$$

Since the row with index h of Φ equals \mathbf{f}_h^\top , we conclude $\sum_{h \in \mathcal{H}} \mathbf{f}_h \mathbf{f}_h^\top = \Phi^\top \Phi = \mathbf{F}$. Let us now sum over $h_1 \in \mathcal{H}$ and use circular shift property of trace:

$$\sum_{h_1 \in \mathcal{H}} \|\Phi \mathbf{f}_{h_1}\|^2 = \sum_{h_1 \in \mathcal{H}} \text{Tr}(\Phi \mathbf{f}_{h_1} \mathbf{f}_{h_1}^\top \Phi^\top) = \text{Tr}(\Phi \mathbf{F} \Phi^\top) = \text{Tr}(\mathbf{F} \Phi^\top \Phi),$$

Therefore,

$$\sum_{h \in \mathcal{H}} \|\Phi \mathbf{f}_h\|^2 = \text{Tr}(\mathbf{F}^2) = \|\mathbf{F}^2\|_1 = \|\mathbf{F}\|_2^2 = \|\Phi\|_4^4 = \|\mathbf{F}\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. \square

Further, we will use the notations Φ , \mathbf{F} and \mathbf{f}_h from the latter lemma. Also, we assume that $\sum_{y \in \mathcal{Y}} f(x, y) = 0$ for any $x \in \mathcal{X}$.

Proof of Lemma 1. Using $\sum_{y \in \mathcal{Y}} f(x, y) = 0$, one can represent entries of \mathbf{F} in the following way:

$$\begin{aligned} \mathbf{F}_{x, x'} &= \sum_{h \in \mathcal{H}} f(x, h(x)) f(x', h(x')) = \\ &= |\mathcal{H}| \sum_{y, y' \in \mathcal{Y}} f(x, y) f(x', y') \mathbb{E}_{h \sim \mathcal{H}}[h(x) = y, h(x') = y'] = \\ &= |\mathcal{H}| \sum_{y, y' \in \mathcal{Y}} f(x, y) f(x', y') (\mathbb{E}_{h \sim \mathcal{H}}[h(x) = y, h(x') = y'] - |\mathcal{Y}|^{-2}), \end{aligned} \tag{12}$$

if $x \neq x'$, and

$$\begin{aligned} \mathbf{F}_{x, x} &= \sum_{h \in \mathcal{H}} f(x, h(x))^2 = |\mathcal{H}| \sum_{y \in \mathcal{Y}} f(x, y)^2 \mathbb{E}_{h \sim \mathcal{H}}[h(x) = y] = \\ &= |\mathcal{H}| \sum_{y \in \mathcal{Y}} f(x, y)^2 (\mathbb{E}_{h \sim \mathcal{H}}[h(x) = y] - |\mathcal{Y}|^{-1}) + |\mathcal{H}| D_x, \end{aligned}$$

where $D_x = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} f(x, y)^2$.

Let \mathbf{G} be a $|\mathcal{X}| \times |\mathcal{X}|$ diagonal matrix with the diagonal elements $\{|\mathcal{H}| D_x\}_{x \in \mathcal{X}}$. From $\sum_{y, y'} |\mathbb{E}_{h \sim \mathcal{H}}[h(x) = y, h(x') = y'] - |\mathcal{Y}|^{-2}| = \varepsilon(x, x')$ and Hölder's inequality we obtain

$$|(\mathbf{F} - \mathbf{G})_{x, x'}| \leq |\mathcal{H}| \cdot \varepsilon(x, x') \max_{y \in \mathcal{Y}} |f(x, y)| \max_{y' \in \mathcal{Y}} |f(x', y')|,$$

for $x \neq x'$, and

$$|(\mathbf{F} - \mathbf{G})_{x, x}| \leq |\mathcal{H}| \cdot \varepsilon(x, x) \left(\max_{y \in \mathcal{Y}} |f(x, y)| \right)^2.$$

Recall that $M_x = \max_{y \in \mathcal{Y}} |f(x, y)|$. Thus, we have

$$\|\mathbf{F} - \mathbf{G}\|_F^2 \leq |\mathcal{H}|^2 \sum_{x, x' \in \mathcal{X}} \varepsilon(x, x')^2 M_x^2 M_{x'}^2.$$

For the latter quadratic form we have

$$\sum_{x, x' \in \mathcal{X}} \varepsilon(x, x')^2 M_x^2 M_{x'}^2 \leq |\mathcal{X}| \cdot \|\varepsilon(x, x')^2\|_{x, x' \in \mathcal{X}} \cdot \mathbb{E}_{X \sim \mathcal{X}}[M_X^4].$$

Since the matrix $[\varepsilon(x, x')^2]_{x, x' \in \mathcal{X}}$ has only non-negative entries, its norm equals the Perron–Frobenius eigenvalue, and it satisfies the following inequality

$$\|\varepsilon(x, x')^2\|_{x, x' \in \mathcal{X}} \leq \max_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \varepsilon(x, x')^2.$$

This gives us

$$\|\mathbf{F} - \mathbf{G}\|_F^2 \leq |\mathcal{H}|^2 \cdot |\mathcal{X}| \cdot \mathbb{E}_{X \sim \mathcal{X}}[M_X^4] \cdot \max_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \varepsilon(x, x')^2.$$

The latter, together with $\|\mathbf{G}\|_F = |\mathcal{H}| \cdot |\mathcal{X}|^{1/2} \mathbb{E}_{X \sim \mathcal{X}}[D_X^2]^{1/2}$ and the triangle inequality, implies

$$\|\mathbf{F}\|_F \leq |\mathcal{H}| \cdot |\mathcal{X}| \cdot \mathbb{E}_{X \sim \mathcal{X}}[M_X^4]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}}[\varepsilon(x, X')^2]^{1/2} + |\mathcal{H}| \cdot |\mathcal{X}|^{1/2} \mathbb{E}_{X \sim \mathcal{X}}[D_X^2]^{1/2}.$$

Using Lemma 2 we directly obtain the first inequality of Lemma 1.

Let us now show the second inequality of Lemma 1. Using the fact that the operator norm of the matrix $[\mathbb{E}_{h \sim \mathcal{H}}[h(x) = y, h(x') = y'] - |\mathcal{Y}|^{-2}]_{y, y'}$ is bounded by the entry-wise 1-norm, i.e. $\varepsilon(x, x')$, the last expression in the equation (12) can be bounded by

$$|\mathcal{H}| \cdot |\mathcal{Y}| \cdot \varepsilon(x, x') \sqrt{D_x} \sqrt{D_{x'}},$$

for $x \neq x'$. Let \mathbf{H} be a diagonal matrix such that $\mathbf{H}_{xx} = \mathbf{F}_{xx}$. Then,

$$\begin{aligned} \|\mathbf{F} - \mathbf{H}\|_F^2 &\leq |\mathcal{H}|^2 \cdot |\mathcal{Y}|^2 \sum_{x, x' \in \mathcal{X}} \varepsilon(x, x')^2 D_x D_{x'} \leq \\ &|\mathcal{H}|^2 \cdot |\mathcal{X}| \cdot |\mathcal{Y}|^2 \cdot \|\varepsilon(x, x')^2\|_{x, x' \in \mathcal{X}} \cdot \mathbb{E}_{X \sim \mathcal{X}}[D_X^2] \leq \\ &|\mathcal{H}|^2 \cdot |\mathcal{X}| \cdot |\mathcal{Y}|^2 \cdot \mathbb{E}_{X \sim \mathcal{X}}[D_X^2] \cdot \max_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} \varepsilon(x, x')^2. \end{aligned}$$

Also, using $|\mathbf{F}_{xx}| \leq |\mathcal{H}| \cdot |\mathcal{Y}| \cdot D_x$ we conclude

$$\|\mathbf{H}\|_F^2 = \sum_{x \in \mathcal{X}} |\mathbf{F}_{xx}|^2 \leq |\mathcal{H}|^2 |\mathcal{Y}|^2 |\mathcal{X}| \mathbb{E}_{X \sim \mathcal{X}}[D_X^2].$$

Therefore,

$$\begin{aligned} \|\mathbf{F}\|_F &\leq |\mathcal{H}| \cdot |\mathcal{X}| \cdot |\mathcal{Y}| \cdot \mathbb{E}_{X \sim \mathcal{X}}[D_X^2]^{1/2} \max_{x \in \mathcal{X}} \mathbb{E}_{X' \sim \mathcal{X}}[\varepsilon(x, X')^2]^{1/2} + \\ &|\mathcal{H}| |\mathcal{X}|^{1/2} |\mathcal{Y}| \mathbb{E}_{X \sim \mathcal{X}}[D_X^2]^{1/2}. \end{aligned}$$

This completes the proof. \square

6 Stronger bounds for LWE

The LWE hypothesis set (7) has some additional structure, which allows to improve our general bound, given in Theorem 2. Moreover, since in applications of LWE, an error term is added to the output of a hypothesis, we will consider the following objective

$$C_{\mathbf{s}}(\mathbf{w}) = \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n, e \sim \chi} [L(p(\mathbf{w}, \mathbf{x}), h_{\mathbf{s}}(\mathbf{x}) + e)], \quad (13)$$

where $L : \mathbb{R} \times \mathbb{Z}_q \rightarrow \mathbb{R}$ is a fixed loss function and χ is any distribution over \mathbb{Z}_q .

Theorem 3. Let $O \subseteq \mathbb{R}^{N_{\text{par}}}$ and $p : O \times \mathbb{Z}_q^n \rightarrow \mathbb{R}$ be a mapping (a neural network) such that $\mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2]$ is bounded uniformly over $\mathbf{w} \in O$ and χ is an arbitrary distribution over \mathbb{Z}_q . We also assume that $M_0^4 \lesssim \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2] q^{2n+1}$. Then, we have

$$\begin{aligned} \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n, e \sim \chi} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle + e)] &\lesssim \\ \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2] \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2]^{1/2} q^{-\frac{n-1}{2}}. \end{aligned} \quad (14)$$

Proof. Let us first handle the case when noise e is absent, i.e. $\mathbb{P}_{x \sim \chi}[x = 0] = 1$. Again, recall that $r_{\mathbf{w}}(\mathbf{x}, y) = \frac{\partial L(p(\mathbf{w}, \mathbf{x}), y)}{\partial p} - \mathbb{E}_{Y' \sim \mathbb{Z}_q} [\frac{\partial L(p(\mathbf{w}, \mathbf{x}), Y')}{\partial p}]$. Further, our proof is identical to the proof of Theorem 2 until the application of the Boas-Bellman inequality:

$$\begin{aligned} \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle)] &\leq \\ q^{-n} \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 &\left[\max_{\mathbf{a} \in \mathbb{Z}_q^n} \|r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle)\|_{\mathbf{x}}^2 + \left(\sum_{\mathbf{a} \neq \mathbf{b}} \langle r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle), r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{b}, \mathbf{x} \rangle) \rangle_{\mathbf{x}} \right)^2 \right]^{1/2}. \end{aligned}$$

To bound the second term we need the following lemma.

Lemma 3. Let the function $f : \mathbb{Z}_q^{n+1} \rightarrow \mathbb{R}$ be such that $\sum_{y \in \mathbb{Z}_q} f(\mathbf{x}, y) = 0$ and $g_{\mathbf{a}}(\mathbf{x}) = f(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle)$. Then, we have

$$\sum_{\mathbf{a} \in \mathbb{Z}_q^n} \sum_{\mathbf{b} \in \mathbb{Z}_q^n} \langle g_{\mathbf{a}}, g_{\mathbf{b}} \rangle^2 \leq \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2] q^{3n+1} + q^{2n} f(\mathbf{0}, 0)^4,$$

where $D_{\mathbf{x}} = \text{Var}_{y \sim \mathbb{Z}_q} [f(\mathbf{x}, y)]$.

Proof of Lemma 3. Let $\Phi_n = [f(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle)]_{(\mathbf{a}, \mathbf{x}) \in \mathbb{Z}_q^{2n}}$. For $\mathbf{a} \in \mathbb{Z}_q^n$, let us define a function $g_{\mathbf{a}} : \mathbb{Z}_q^n \rightarrow \mathbb{R}$ by $g_{\mathbf{a}}(\mathbf{x}) = f(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle)$. Then, from Lemma 2 we have

$$\sum_{\mathbf{a} \in \mathbb{Z}_q^n} \sum_{\mathbf{b} \in \mathbb{Z}_q^n} \langle g_{\mathbf{a}}, g_{\mathbf{b}} \rangle^2 = \|\Phi_n\|_4^4 = \|\mathbf{F}\|_F^2,$$

where $\mathbf{F} = \Phi^\top \Phi$.

Non-diagonal elements of \mathbf{F} such that $\forall \lambda \in \mathbb{Z}_q \lambda \mathbf{y}' \neq \mathbf{y}, \mathbf{y}' \neq \lambda \mathbf{y}$, are zeros due to $\sum_{z \in \mathbb{Z}_q} f(\mathbf{y}, z) = 0$ and

$$\mathbf{F}_{\mathbf{y}, \mathbf{y}'} = \sum_{\mathbf{a} \in \mathbb{Z}_q^n} f(\mathbf{y}, \langle \mathbf{a}, \mathbf{y} \rangle) f(\mathbf{y}', \langle \mathbf{a}, \mathbf{y}' \rangle) = \sum_{y \in \mathbb{Z}_q} f(\mathbf{y}, y) \sum_{\mathbf{a} \in \mathbb{Z}_q^n : \langle \mathbf{a}, \mathbf{y} \rangle = y} f(\mathbf{y}', \langle \mathbf{a}, \mathbf{y}' \rangle) = 0.$$

The set of nonzero elements of \mathbb{Z}_q^n can be divided into equivalence classes w.r.t. the equivalence relation $\mathbf{y} \sim \mathbf{y}' \Leftrightarrow \exists \lambda \in \mathbb{Z}_q^* \mathbf{y}' = \lambda \mathbf{y}$. For any two \mathbf{y}, \mathbf{y}' from the same equivalence class c , we have

$$|\mathbf{F}_{\mathbf{y}, \mathbf{y}'}| = |q^{n-1} \sum_{y \in \mathbb{Z}_q} f(\mathbf{y}, y) f(\lambda \mathbf{y}, \lambda y)| \leq q^n \sqrt{D_{\mathbf{y}} D_{\mathbf{y}'}}$$

due to the Cauchy-Schwarz inequality. Therefore, $\sum_{\mathbf{y}, \mathbf{y}' \in c} |\mathbf{F}_{\mathbf{y}, \mathbf{y}'}|^2 \leq q^{2n} (\sum_{\mathbf{y} \in c} D_{\mathbf{y}})^2$.

Residual non-diagonal elements of \mathbf{F} are in the first row or the first column and they are equal to

$$\mathbf{F}_{\mathbf{0}, \mathbf{y}'} = \sum_{\mathbf{a} \in \mathbb{Z}_q^n} f(\mathbf{0}, 0) f(\mathbf{y}', \langle \mathbf{a}, \mathbf{y}' \rangle) = 0,$$

if $\mathbf{y}' \neq \mathbf{0}$ and $\mathbf{F}_{\mathbf{0}, \mathbf{0}} = q^n f(\mathbf{0}, 0)^2$.

Finally, we have

$$\begin{aligned} \|\mathbf{F}\|_F^2 &\leq \sum_c q^{2n} \left(\sum_{\mathbf{y} \in c} D_{\mathbf{y}} \right)^2 + q^{2n} f(\mathbf{0}, 0)^4 \leq \sum_c q^{2n} |c| \sum_{\mathbf{y} \in c} D_{\mathbf{y}}^2 + q^{2n} f(\mathbf{0}, 0)^4 \leq \\ & q^{2n+1} q^n \mathbb{E}_{\mathbf{y} \sim \mathbb{Z}_q^n} [D_{\mathbf{y}}^2] + q^{2n} f(\mathbf{0}, 0)^4. \end{aligned}$$

□

From Lemma 3, after setting f as $r_{\mathbf{w}}$, we obtain

$$\sum_{\mathbf{a}, \mathbf{b}} \langle r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle), r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{b}, \mathbf{x} \rangle) \rangle^2 \leq \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2] q^{3n+1} + M_{\mathbf{0}}^4 q^n \lesssim \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2] q^{3n+1}.$$

The latter follows from the assumption that $M_{\mathbf{0}}^4 \lesssim \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2] q^{2n+1}$. Also, note that

$$\max_{\mathbf{a} \in \mathbb{Z}_q^n} \|r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle)\|_{\mathbf{x}}^2 \leq \left(\sum_{\mathbf{a}, \mathbf{b}} \langle r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{a}, \mathbf{x} \rangle), r_{\mathbf{w}}(\mathbf{x}, \langle \mathbf{b}, \mathbf{x} \rangle) \rangle^2 \right)^{1/2} \lesssim \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2]^{1/2} q^{\frac{n+1}{2}}.$$

After we plug in the latter two inequalities into the previous bound, we obtain the needed fact for the zero noise case:

$$\text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} L(p(\mathbf{w}, \mathbf{x}), h_{\mathbf{a}}(\mathbf{x}))] \lesssim \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [D_{\mathbf{x}}^2]^{1/2} q^{-\frac{n-1}{2}}.$$

Let us now address the case when noise $e \sim \chi$ is added to the target function $\langle \mathbf{a}, \mathbf{x} \rangle$. Indeed, let the optimized objective be

$$\mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n, e \sim \chi} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle + e).$$

If we define $\tilde{L}(p, y) = \mathbb{E}_{e \sim \chi} L(p, y + e)$, then the objective obtains the previous form $\mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} \tilde{L}(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle)$. If the factor in the RHS of (14) is moderate for the old loss L , the new loss function \tilde{L} inherits this property, due to

$$\begin{aligned} \text{Var}_{y \sim \mathbb{Z}_q} \left[\frac{\partial \tilde{L}(p(\mathbf{w}, \mathbf{x}), y)}{\partial p} \right] &= \text{Var}_{y \sim \mathbb{Z}_q} \left[\mathbb{E}_{e \sim \chi} \frac{\partial L(p(\mathbf{w}, \mathbf{x}), y + e)}{\partial p} \right] \leq \\ \mathbb{E}_{e \sim \chi} \text{Var}_{y \sim \mathbb{Z}_q} \left[\frac{\partial L(p(\mathbf{w}, \mathbf{x}), y + e)}{\partial p} \right] &= \text{Var}_{y \sim \mathbb{Z}_q} \left[\frac{\partial L(p(\mathbf{w}, \mathbf{x}), y)}{\partial p} \right]. \end{aligned}$$

This completes the proof. □

Though the bound of Theorem 3 is better than the bound of Theorem 1, it is still far from being optimal. Indeed, for $n = 1$, the LWE hypothesis set consists of modular multiplications $\{x \rightarrow ax \bmod q\}_{a \in \mathbb{Z}_q}$. For this case, the RHS of the inequality (14) does not guarantee any concentration even for large q . A slightly better bound can be given if we specify the form of the loss function. For simplicity of notations, we will omit an analysis of the non-zero noise case and will assume that $e = 0$.

Theorem 4. *Let a function $t : \mathbb{Z}_q \rightarrow \{c_1, \dots, c_k\} \subseteq \mathbb{R}$ be such that $L(s, y) = l(s, t(y))$ and $|\frac{\partial l(s, c_i)}{\partial s}| \leq c$ for any $s \in \mathbb{R}, i \in \{1, \dots, k\}$. Then, we have*

$$\begin{aligned} \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle)] &\leq \\ 2c^2 \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2] &\left(\sum_{j=1}^k \mathbb{E}_{z \sim \mathbb{Z}_q^*} [|\widehat{\mathbb{1}_{t(x)=c_j}}(z)|]^2 \right) q^{-\frac{n}{2}}. \end{aligned}$$

Remark 4. Suppose that we are interested not in the whole output of $\mathbf{x} \rightarrow \langle \mathbf{s}, \mathbf{x} \rangle$ but only in one bit of information about the output. In this case it is natural to define the loss by $L(p, y) = l(p, t(y))$ where $l : \mathbb{R} \times \{0, 1\} \rightarrow \mathbb{R}$ is c -Lipschitz w.r.t. the first variable and $t : \mathbb{Z}_q \rightarrow \{0, 1\}$. Then $|\frac{\partial L(p, y)}{\partial p}| \leq c$ and the bound of Theorem 3 gives that the variance of the gradient is $\mathcal{O}(\mathbb{E}_{\mathbf{x} \in \mathbb{Z}_q^n} [(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2] q^{-\frac{n-1}{2}})$. Alternatively, Theorem 4 gives an upper bound of

$$\mathcal{O}(\mathbb{E}_{\mathbf{x} \in \mathbb{Z}_q^n} [(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2] (\mathbb{E}_{y \in \mathbb{Z}_p^*} [\widehat{t}(y)]^2) q^{-\frac{n}{2}}),$$

which is slightly better for special cases of t . For example, let $t(x) = \mathbb{1}_{S_r}$ where S_r is a set of elements in \mathbb{Z}_q whose binary representation has 1 at the r th position from the end, $1 \leq r \leq \lceil \log_2 q \rceil$. This definition of the loss function is equivalent to learning the concept class $\{\mathbf{x} \rightarrow \mathbb{1}_{S_r}(\langle \mathbf{s}, \mathbf{x} \rangle)\}_{\mathbf{s} \in \mathbb{Z}_q^n}$. In Theorem 5 of Section A one can find the proof of the bound $\mathbb{E}_{y \in \mathbb{Z}_q^*} [\|\widehat{\mathbb{1}_{S_r}}(y)\|] = \mathcal{O}(r(\log_2 q + 1 - r))$. From the latter fact it is clear that the bound of Theorem 4 gives us $\text{Var}_{\mathbf{s} \sim \mathbb{Z}_q^n} [\nabla C_{\mathbf{s}}(\mathbf{w})] = \mathcal{O}(\mathbb{E}_{\mathbf{x} \in \mathbb{Z}_q^n} [(\partial_{w_i} p(\mathbf{w}, \mathbf{x}))^2] q^{-\frac{n}{2}} \log^4 q)$ which is better than the bound of Theorem 3 by a factor of $\frac{\sqrt{q}}{\log^4 q}$.

If we set $n = 1$, then the target function is a simple modular multiplication by some number $a \in \mathbb{Z}_q$. As we see, for a large prime q , the gradient of the loss becomes noninformative. \square

Proof of Theorem 4. The expression $\|\Phi_n\|_4$ also can be bounded using methods of discrete Fourier analysis. Let $\varepsilon = e^{\frac{2\pi i}{q}}$ be a primitive q th root of unity. Other primitive roots of unity are $\varepsilon_2, \dots, \varepsilon_{q-1}$ where $\varepsilon_k = \varepsilon^k, k \in \mathbb{Z}_q$. The matrix $\frac{1}{\sqrt{q}} \mathbf{U}_k$, where $\mathbf{U}_k = [\varepsilon_k^{ij}]_{i, j \in \mathbb{Z}_q}$, is unitary for $k \in \mathbb{Z}_q^*$. In fact, \mathbf{U}_1^\top is a discrete Fourier transform (DFT) matrix. Let us denote columns of \mathbf{U}_1 by $\mathbf{b}_0, \dots, \mathbf{b}_{q-1}$. Now everything is ready for the proof of Theorem 4.

The following chain of identities is straightforward,

$$\begin{aligned} & \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle)] = \\ & \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} \frac{\partial l(p(\mathbf{w}, \mathbf{x}), t(\langle \mathbf{a}, \mathbf{x} \rangle))}{\partial p} \partial_{w_i} p(\mathbf{w}, \mathbf{x})] = \\ & \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\langle \partial_{w_i} p(\mathbf{w}, \mathbf{x}), \sum_{j=1}^k \frac{\partial l(p(\mathbf{w}, \mathbf{x}), c_j)}{\partial p} \mathbb{1}_{t(\langle \mathbf{a}, \mathbf{x} \rangle) = c_j} \rangle_{\mathbf{x}}] = \\ & \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\sum_{j=1}^k \langle \frac{\partial l(p(\mathbf{w}, \mathbf{x}), c_j)}{\partial p} \partial_{w_i} p(\mathbf{w}, \mathbf{x}), \mathbb{1}_{t(\langle \mathbf{a}, \mathbf{x} \rangle) = c_j} \rangle_{\mathbf{x}}] \leq \\ & (\sum_{j=1}^k \sqrt{\text{Var}_{\mathbf{a}} [\langle \frac{\partial l(p(\mathbf{w}, \mathbf{x}), c_j)}{\partial p} \partial_{w_i} p(\mathbf{w}, \mathbf{x}), \mathbb{1}_{t(\langle \mathbf{a}, \mathbf{x} \rangle) = c_j} \rangle_{\mathbf{x}}]})^2. \end{aligned}$$

Let us denote $m_j(y) = \mathbb{1}_{t(y) = c_j} - \frac{\sum_{y' \in \mathbb{Z}_q} \mathbb{1}_{t(y') = c_j}}{q}$. Then, every single variance in the latter expression can be bounded using the Boas-Bellman inequality, i.e.

$$\begin{aligned} & \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\langle \frac{\partial l(p(\mathbf{w}, \mathbf{x}), c_j)}{\partial p} \partial_{w_i} p(\mathbf{w}, \mathbf{x}), \mathbb{1}_{t(\langle \mathbf{a}, \mathbf{x} \rangle) = c_j} \rangle_{\mathbf{x}}] \leq \\ & \mathbb{E}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\langle \frac{\partial l(p(\mathbf{w}, \mathbf{x}), c_j)}{\partial p} \partial_{w_i} p(\mathbf{w}, \mathbf{x}), m_j(\langle \mathbf{a}, \mathbf{x} \rangle) \rangle_{\mathbf{x}}^2] \leq \\ & 2q^{-n} c^2 \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 (\sum_{\mathbf{a} \in \mathbb{Z}_q^n} \sum_{\mathbf{b} \in \mathbb{Z}_q^n} \langle m_j(\langle \mathbf{a}, \mathbf{x} \rangle), m_j(\langle \mathbf{b}, \mathbf{x} \rangle) \rangle_{\mathbf{x}}^2)^{1/2}. \end{aligned}$$

Using Lemma 2, the expression $\sum_{\mathbf{a} \in \mathbb{Z}_q^n} \sum_{\mathbf{b} \in \mathbb{Z}_q^n} \langle m_j(\langle \mathbf{a}, \mathbf{x} \rangle), m_j(\langle \mathbf{b}, \mathbf{x} \rangle) \rangle_{\mathbf{x}}$ equals $\|[m_j(\langle \mathbf{a}, \mathbf{x} \rangle)]_{(\mathbf{a}, \mathbf{x}) \in \mathbb{Z}_q^{2n}}\|_4^4$.

The inverse DFT of m_j can be understood as an expansion

$$[m_j(y)]_{y \in \mathbb{Z}_q} = \sum_{i=0}^{q-1} (\mathbf{e}_i^\dagger [m_j(y)]_{y \in \mathbb{Z}_q}) \mathbf{e}_i,$$

where $\{\mathbf{e}_i = \frac{1}{\sqrt{q}} \mathbf{b}_i\}_{i=0}^{q-1}$ is an orthonormal basis in \mathbb{C}^q . Note that

$$(\mathbf{e}_i^\dagger [m_j(y)]_{y \in \mathbb{Z}_q}) = \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \varepsilon^{-yi} m_j(y) = \frac{1}{\sqrt{q}} \widehat{m}_j(i).$$

Thus, we conclude that

$$[m_j(y)]_{y \in \mathbb{Z}_q} = \sum_{i=0}^{q-1} \frac{1}{\sqrt{q}} \widehat{m}_j(i) \mathbf{e}_i = \frac{1}{q} \sum_{i=0}^{q-1} \widehat{m}_j(i) \mathbf{b}_i.$$

From the latter equation we conclude $m_j(y) = \frac{1}{q} \sum_{k=0}^{q-1} \widehat{m}_j(k) \varepsilon^{ky}$, and therefore, $m_j(\langle \mathbf{x}, \mathbf{y} \rangle) = \frac{1}{q} \sum_{k=0}^{q-1} \widehat{m}_j(k) \varepsilon^{k \mathbf{x}^\top \mathbf{y}}$ where $\mathbf{x}^\top \mathbf{y}$ is a dot product over \mathbb{R}^n . Note that for $\mathbf{x} = [x_i]_1^n$, $\mathbf{y} = [y_i]_1^n \in \mathbb{Z}_q^n$ we have $\varepsilon^{k \mathbf{x}^\top \mathbf{y}} = \mathbf{U}_k[x_1, y_1] \times \cdots \times \mathbf{U}_k[x_n, y_n]$, or equivalently,

$$[m_j(\langle \mathbf{x}, \mathbf{y} \rangle)]_{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_q^{2n}} = \frac{1}{q} \sum_{k=0}^{q-1} \widehat{m}_j(k) \mathbf{U}_k^{\otimes n},$$

where $\mathbf{A}^{\otimes n} = \mathbf{A} \otimes \cdots \otimes \mathbf{A}$ is the n th tensor power of \mathbf{A} . Note that $\widehat{m}_j(0) = 0$ due to $\sum_{z \in \mathbb{Z}_q} m_j(z) = 0$.

We have the following bound:

$$\begin{aligned} \|[m_j(\langle \mathbf{x}, \mathbf{y} \rangle)]_{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_q^{2n}}\|_4 &= \frac{1}{q} \left\| \sum_{k=1}^{q-1} \widehat{m}_j(k) \mathbf{U}_k^{\otimes n} \right\|_4 \leq \\ \frac{1}{q} \sum_{k=1}^{q-1} \|\widehat{m}_j(k) \mathbf{U}_k^{\otimes n}\|_4 &= \frac{q^{\frac{3n}{4}}}{q} \sum_{k=1}^{q-1} |\widehat{m}_j(k)| \leq q^{\frac{3n}{4}} \mathbb{E}_{k \sim \mathbb{Z}_q^*} [|\widehat{\mathbf{1}}_{t(x)=c_j}(k)|]. \end{aligned}$$

Thus,

$$\begin{aligned} \text{Var}_{\mathbf{a} \sim \mathbb{Z}_q^n} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle)] &\leq \\ \left(\sum_{j=1}^k \sqrt{2q^{-n} c^2 \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 q^{-n} q^{\frac{3n}{2}} (\mathbb{E}_{z \sim \mathbb{Z}_q^*} [|\widehat{\mathbf{1}}_{t(x)=c_j}(z)|]^2)} \right)^2 & \\ = 2c^2 \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 q^{-\frac{n}{2}} \left(\sum_{j=1}^k \mathbb{E}_{z \sim \mathbb{Z}_q^*} [|\widehat{\mathbf{1}}_{t(x)=c_j}(z)|]^2 \right). & \end{aligned}$$

From the latter the statement of theorem is straightforward. \square

6.1 Application of Theorem 2 to SPNs

Another class of functions to which Theorem 2 can be applied directly is Substitution-Permutation Networks. Let $k, b, r \in \mathbb{N}$ and $n = kb$. Let $\mathcal{X} = \mathcal{Y} = \text{GF}(2^l)$ where $\text{GF}(2^l)$ denotes the Galois field with 2^l elements. A Substitution-Permutation Network (SPN) is a parameterized family of functions

$$\mathcal{H} = \left\{ F_{(k_0, \dots, k_r)}^{(r)} \right\}_{k_0, \dots, k_r \in \text{GF}(2^n)}, \quad (15)$$

defined by the following equations

$$\begin{aligned} F_{(k_0)}^{(0)}(x) &= x \oplus k_0, \\ F_{(k_0, \dots, k_i)}^{(i)}(x) &= P(F_{(k_0, \dots, k_{i-1})}^{(i-1)}(x)) \oplus k_i, i = 1, \dots, r, \end{aligned}$$

where, for an input $x = [x_1, \dots, x_k] \in \text{GF}(2^n)$, $x_i \in \text{GF}(2^b)$, $P : \text{GF}(2^n) \rightarrow \text{GF}(2^n)$ is defined by

$$P(x) = M \begin{bmatrix} S(x_1) \\ \dots \\ S(x_k) \end{bmatrix},$$

and $S : \text{GF}(2^b) \rightarrow \text{GF}(2^b)$ is some fixed nonlinear mapping and $M \in \text{GF}(2^b)^{k \times k}$ is some fixed matrix. By construction, for any $x \in \mathcal{X}$, $\varepsilon(x, x) = 0$.

For a case where $S(x) = x^{2^b-2}$ and M is an invertible matrix without zero entries², it was shown in [LTV21] that

$$\varepsilon(x, x') \leq 2^s \left(\frac{2+8k}{2^b} + \sqrt{\frac{k}{2^b}} \right)^s, \quad (16)$$

where $r = 3s$ and x, x' are distinct. In fact, the requirement on M can be made milder to capture the Advanced Encryption Standard (AES), for which the following bound was proved in [LTV21]:

$$\varepsilon(x, x') \leq 0.944^s,$$

where $r = 6s$ and x, x' are distinct. From (16) the following corollary is straightforward.

Corollary 2 (of Theorem 2). *Let $O \subseteq \mathbb{R}^{N_{\text{par}}}$ and $p : O \times \mathcal{X} \rightarrow \mathbb{R}$ satisfy conditions of Theorem 2. Let the hypothesis set defined in (15) be such that the inequality (16) holds. Then, for $r = 3s$ we have*

$$\begin{aligned} \text{Var}_{h \sim \mathcal{H}} [\partial_{w_i} \mathbb{E}_{X \sim \mathcal{X}} L(p(\mathbf{w}, X), h(X))] &\lesssim \\ \mathbb{E}_{X \sim \mathcal{X}} [(\partial_{w_i} p(\mathbf{w}, X))^2] \cdot \mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} &\left(\frac{2+8k}{2^{b-1}} + \sqrt{\frac{k}{2^{b-2}}} \right)^s. \end{aligned} \quad (17)$$

Proof. From the inequality (16) we conclude

$$\max_{\mathbf{x} \in \mathcal{X}} \mathbb{E}_{X \sim \mathcal{X}} \varepsilon(\mathbf{x}, X)^2 \leq \frac{2^n - 1}{2^n} \left(\frac{2+8k}{2^{b-1}} + \sqrt{\frac{k}{2^{b-2}}} \right)^{2s} + 2^{-n+2} \lesssim \left(\frac{2+8k}{2^{b-1}} + \sqrt{\frac{k}{2^{b-2}}} \right)^{2s}.$$

From Theorem 2 we obtain the final bound

$$\begin{aligned} \text{Var}_{h \sim \mathcal{H}} [\partial_{w_i} \mathbb{E}_{X \sim \mathcal{X}} L(p(\mathbf{w}, X), h(X))] &\lesssim \\ \mathbb{E}_{X \sim \mathcal{X}} [(\partial_{w_i} p(\mathbf{w}, X))^2] \cdot \mathbb{E}_{X \sim \mathcal{X}} [M_X^4]^{1/2} &\left(\frac{2+8k}{2^{b-1}} + \sqrt{\frac{k}{2^{b-2}}} \right)^s. \end{aligned}$$

□

²We use a simpler formulation, though [LTV21] gives more general conditions.

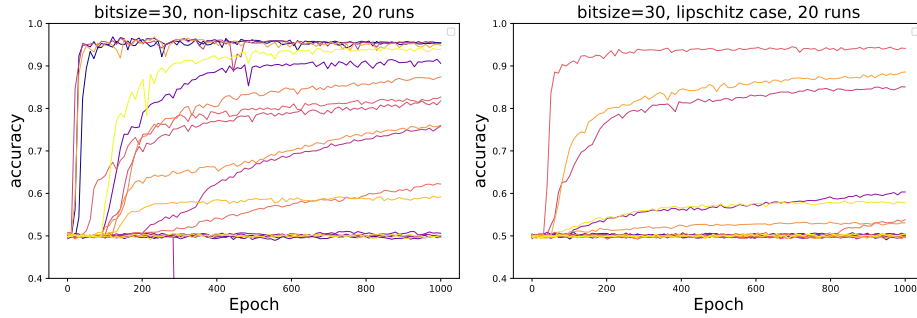


Figure 1: An accuracy on a test set as a function of epoch for different runs. An accuracy drop to 0 means weights blow up to Nan.

7 Experiments

We conducted several computational experiments to compare predictions of our theory with recent findings of LWE cryptanalysis. We verified the barren plateau phenomenon by experimenting with the learnability of simple modular multiplication. Second, we demonstrated that sparse secrets lead to larger RHS expressions, meaning the gradient can be more informative. Third, we made some remarks about recent gradient-based attacks on LWE.

Learnability of modular multiplication by NNs with non-lipschitz activation functions. The factor $\|\nabla_{\mathbf{w}} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2$ from the RHS of bounds of Theorems 3 and 4 is the norm of the gradient of an NN to be trained. If we use a standard activation function in our NN, like tanh or ReLU, this norm’s value stays moderate as long as the norm of \mathbf{w} does not blow up. Thus, according to our analysis, the gradient of an objective has a low information content in the region of a bounded weight vector’s norm.

We studied the learnability of the random mapping $x \rightarrow kx \bmod q$, where $k \sim \mathbb{Z}_q$, as a function of the bitsize of the prime number q , i.e. $\lceil \log_2 q \rceil$. In an attempt to overcome the vanishing of the gradient on the barren plateau, we experimented with the non-lipschitz activation function $a(x) = \text{sign}(x)\sqrt{|x|}$. The derivative of $a(x)$ has a singularity at zero, so to avoid the exploding gradient problem we used the clipping of the gradient. We defined the loss function as

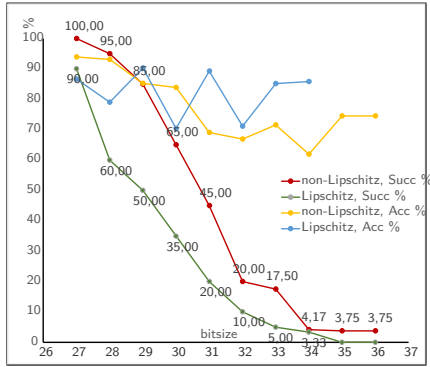
$$L(p(\mathbf{w}, x), y) = \text{hinge}((-1)^y, p(\mathbf{w}, x)),$$

where $\text{hinge}(y, v) = \max(0, 1 - yv)$ and $(-1)^y$ encodes the parity bit of $y \in \mathbb{Z}_q$.

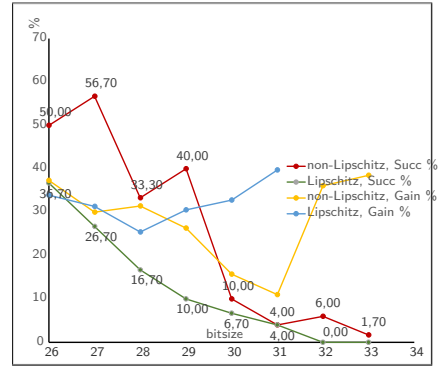
In Figure 1 one can see plots of prediction accuracy on a test set for various bits as a function of the number of epochs. For comparison, we experimented with two architectures of 3-Layer feedforward NNs³: the first one had activation functions [tanh, tanh, a] (non-lipschitz case) and the second one had [tanh, tanh, x] (lipschitz case). Both architectures had the number of neurons [1000, 1000, 1] and were trained by Adam with a learning rate of 0.001. An input $x \in \mathbb{Z}_q$ was treated as a vector of length $\lceil \log_2 q \rceil$ with components equal to bits of the binary representation of x .

The training process itself and its outcome are of stochastic nature, and the variability of the process increases for non-lipschitz activations (see Figure 1). However it is important to note that if the probability of success for a certain bitsize is large enough, say 10%, we observe that a multiplication by any number modulo such a prime can be trained provided enough number of independent runs of the learning algorithm. In other words,

³Note that three layers of NN is enough to approximate the needed function, since LWE mapping is simply a linear function combined with some nonlinear ψ which can be L_2 -approximated using a 2-layer NN.



(a) Performances of NNs with non-lipschitz ($a(x)$) and lipschitz activation functions for various bitsizes: the noiseless case. The first value, Succ, is a percentage of successful runs (we define success as an achieved accuracy of more than 52% on a test set). The second value, Acc, is an average accuracy on a test set for successful runs.



(b) The noisy case. The first value, Succ, is a percentage of successful runs. We define a success as an achieved gain of more than 2% on a test set, where a gain is defined as an accuracy minus $\max(p, 1 - p)$ where $p = \mathbb{P}_{x \sim \mathbb{Z}_q} [\lceil x \rceil_{\lceil \log_2 q \rceil - 1} = 1]$. The second value, Gain, is an average gain on a test set for successful runs.

the probability of successful learning does not depend on any specifics of a prime number q and a key k , it only depends on the bitsize.

To summarize, we verified the hypothesis that non-lipschitz activation functions (or, alternatively, smooth functions with a large Lipschitz constant) can potentially unleash NNs' capacity to learn modular multiplication (see Figure 2a). Although such NNs suffer from numerical instabilities, the gradient of their objective could contain a useful signal that would be suppressed by a smooth activation. Using a non-lipschitz NN we were able to learn the parity bit of modular multiplication for the bitsize 36. For the bitsize 40 none of our attempts ever succeeded.

Experiments with the noisy case. We also studied the learnability of the random mapping $x \rightarrow kx + \chi(x) \bmod q$, where $k \sim \mathbb{Z}_q$ and $\{\chi(x)\}_{x \in \mathbb{Z}_q}$ is a discretized normal random vector with zero mean and the covariance matrix $(0.01q)^2 I_q$. Unlike the noiseless case, we tried to predict not the parity bit of an output y , but the $\lceil \log_2 q \rceil - 1$ -st bit from the end in a binary representation of y . This is due to the fact that the distribution of the parity bit $[kx + \chi(x) \bmod q]_1$, given an input x , is a Bernoulli random variable with parameter $\frac{1}{2}$ (i.e. does not depend on x). In other words, we defined the loss function as

$$L(p(\mathbf{w}, x), y) = \text{hinge}((-1)^{\lfloor y \rfloor_{\lceil \log_2 q \rceil - 1}}, p(\mathbf{w}, x)).$$

The results of our experiments are given in Figure 2b. This figure verifies that non-lipschitz NNs can learn better than regular ones. Our code is available on [github](#) to facilitate the reproducibility of the results. Let us also give some remarks on the previously published attacks on LWE.

Sparse secrets and information content of the gradient. The upper bounds derivation method, based on the Boas-Bellman inequality, can be also applied when the secret key is restricted to belong to a certain subset of \mathbb{Z}_q^n . The scaled square loss function is especially suitable for analysis, so we will assume that $L(p, y) = \frac{1}{2}q^{-2}(p - y)^2$. If $S \subseteq \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$ is such a subset and under the assumption that keys are sampled uniformly from S , the variance of the gradient can be bounded by $\|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 \text{BB}_S$ where

$$\text{BB}_S = \frac{(q-1)^2}{4q^4|S|} + q^{-4} \left(\mathbb{E}_{\mathbf{k} \neq \mathbf{k}' \sim S} \left\langle \langle \mathbf{k}, \mathbf{x} \rangle - \frac{q-1}{2}, \langle \mathbf{k}', \mathbf{x} \rangle - \frac{q-1}{2} \right\rangle^2 \right)^{\frac{1}{2}}$$

and $\mathbf{k} \neq \mathbf{k}' \sim S$ denotes the fact that \mathbf{k} and \mathbf{k}' are sampled from S without replacement.

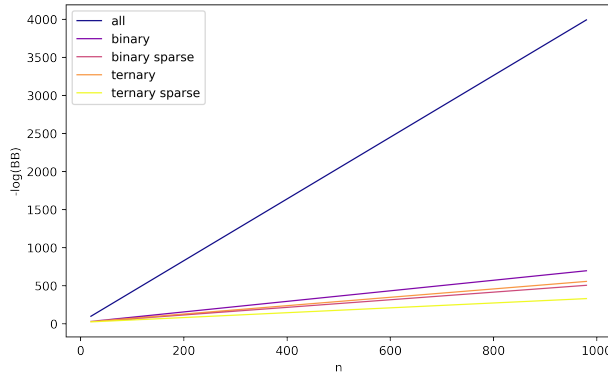


Figure 3: $-\ln BB_S$ as a function of n for different sets of possible keys S .

The first term in the expression of BB_S is inversely proportional to $|S|$ and is unavoidable. The second term measures how nearly orthogonal centered functions $\mathbf{x} \rightarrow \langle \mathbf{k}, \mathbf{x} \rangle$ are. In Section B we demonstrate that in cases defined below the second term is proportional to $\frac{1}{\sqrt{|S|}}$. It is natural to expect that the proportionality of the second term to $\frac{1}{\sqrt{|S|}}$ is a general phenomenon that holds in other interesting cases of S . The second term equals 0 when the latter functions are ideally orthogonal and it becomes dominating when it is non-zero. Then, $-\ln BB_S$ is a natural proxy measure of the hardness of learning such function class when the secret key is restricted to be from S .

We empirically calculated $-\ln BB_S$ for (a) $S = \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$ (all keys); (b) $S = \{0, 1\}^n \setminus \{\mathbf{0}\}$ (binary secrets); (c) $S = \{\mathbf{k} \in \{0, 1\}^n \mid |\mathbf{k}| = l\} \setminus \{\mathbf{0}\}$, where $|\cdot|$ denotes the number of non-zero entries (sparse binary secrets with Hamming weight l); (d) $S = \{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$ (ternary secrets); (e) $S = \{\mathbf{k} \in \{-1, 0, 1\}^n \mid |\mathbf{k}| = l\} \setminus \{\mathbf{0}\}$ (sparse ternary secrets with Hamming weight l). In Figure 3 one can see plots for $q = 3329$ (a popular prime number in LWE applications) as a function of n . For sparse keys, we set $\frac{l}{n} = 0.2$.

As we see, every considered restriction on a set of possible keys leads to a substantial decrease of $-\ln BB_S$. The main reason for this behavior is the sharp difference in the number of secret keys. This is in full correspondence with recent findings on the learnability of LWE with sparse secrets [LWAZ⁺23, WCCL22]. A less obvious phenomenon is the higher level of hardness of binary keys in comparison with ternary keys (the same relationship holds between sparse binary and sparse ternary keys), which seemingly contradicts the fact that $3^n > 2^n$. This can be explained by the strict orthogonality of centered functions $\mathbf{x} \rightarrow \langle \mathbf{k}, \mathbf{x} \rangle$ when \mathbf{k} is binary, i.e. the second term in BB_S vanishes in this case. For ternary keys and sparse ternary keys, the second term in BB_S is dominating. In other words, ternarity makes centered functions $\mathbf{x} \rightarrow \langle \mathbf{k}, \mathbf{x} \rangle$ less orthogonal to each other when varying \mathbf{k} . This, in turn, leads to an increase in the RHS of our bound.

A weakness of the suggested analysis is in the fact that we use the indirect measure of hardness based on BB_S . Although, if we decide to rely on that approach, we will come to a simple practical recommendation of using only those restrictions S for which the second term, i.e. the $\frac{1}{\sqrt{|S|}}$ -proportional term in the expression of BB_S , vanishes completely. Details can be found in Section B.

Remarks on SALSAs and non-uniform distributions over inputs. An approach of SALSA [WCCL22] is completely covered by our formalism, and achievements of SALSA (i.e. the dimension n , the prime size $\lceil \log_2 q \rceil$, and the Hamming height h of the secret key in a sparse binary LWE that were successfully attacked) are in full correspondence with our bounds.

SALSA PICANTE [LSW⁺23] and SALSA VERDE [LWAZ⁺23] are definitely beyond

Table 1: From [WCCL22]. Here d denotes the density $\frac{h}{n}$. The table shows a fraction of the secret recovered by SALSA for $n = 50$ as a function of d and a .

$d \backslash a$	0.35	0.4	0.45	0.5	0.55	0.6	0.65
0.16	1.0	1.0	1.0	1.0	1.0	1.0	0.88
0.18	1.0	1.0	1.0	1.0	0.82	0.86	0.84
0.20	1.0	1.0	1.0	1.0	1.0	0.82	0.82
0.22	0.98	1.0	1.0	0.98	0.80	0.78	0.86
0.24	1.0	1.0	1.0	0.98	0.78	0.78	0.80
0.26	1.0	1.0	0.88	0.92	0.76	0.76	0.76
0.28	0.98	1.0	0.80	0.74	0.74	0.76	0.74
0.30	0.98	1.0	0.93	0.76	0.72	0.74	0.74

our formalism and should be considered as mainly BKZ-based. For example, in a successful attack on LWE with $n = 350$, $\lceil \log_2 q \rceil = 32$, SALSA PICANTE’s preprocessing (that prepares a training set of size four million for a gradient-based training) requires 6000 CPUs working 194 hours in parallel (equivalent to 133 years overall). The training took 105 minutes per epoch and 18 epochs till success took 31.5 hours. So, the ratio between the overall times of the preprocessing and the training is 37000. This indicates that the preprocessing stage is the one that is responsible for most of the work.

According to our bounds, the reported case of $n = 350$, $\lceil \log_2 q \rceil = 32$, $h = 60$ is absolutely infeasible by a direct gradient-based attack, if training would have been on uniformly random (or, uniformly pseudo-random) inputs. Authors of SALSA suggest the following reason for their success. From Table 1 taken from [WCCL22] it is evident that the learnability improves when input vectors of the LWE mapping are sampled from $(\mathbb{Z} \cap [0, aq])^n$ where $0 < a < 1$. For $n = 50$, $a = 0.4$ the whole secret is recovered, whereas for $n = 50$, $a = 0.65$ only $\approx 80\%$ is recovered. Thus, the role of the heavy preprocessing becomes now clear — it computes inputs (with corresponding outputs) from $(\mathbb{Z} \cap [0, aq])^n$.

We have not yet analyzed the case of a general (non-uniform) distribution over inputs. We believe that the vanishing of the gradient holds for this case also, though a bound should be milder (which explains the success of SALSA PICANTE). This is a future work for us.

8 Conclusions and open problems

It has been known for some time that API block ciphers are resilient to differential and linear cryptanalysis attacks. Our analysis shows that any API class of functions is a hard target for learning by gradient-based methods, provided that the NN being trained is regularly parameterized. As an example, we verified that any gradient-based attack on LWE suffers from the barren plateau phenomenon. For SPNs, under a certain choice of its parameters, we also demonstrated that the gradient of a training loss becomes noninformative. It is an open question as to what LWE secret key size is susceptible to an attack by a non-regular NN parameterization. Also, it is an open question to verify the barren plateau phenomenon for modern ciphers such as AES.

Acknowledgments

This research has been funded by Nazarbayev University under Faculty-development competitive research grants program for 2023-2025 Grant #20122022FD4131, PI R. Takhanov.

References

- [AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Information Processing Letters*, 88(3):107–110, 2003. URL: <https://www.sciencedirect.com/science/article/pii/S020019003003594>, doi:10.1016/S0020-0190(03)00359-4.
- [AKJM21] Ezat Ahmadzadeh, Hyunil Kim, Ongee Jeong, and Inkyu Moon. A novel dynamic attack on classical ciphers using an attention-based lstm encoder-decoder model. *IEEE Access*, 9:60960–60970, 2021. doi:10.1109/ACCESS.2021.3074268.
- [Ala12] Mohammed M. Alani. Neuro-cryptanalysis of des and triple-des. In Tingwen Huang, Zhigang Zeng, Chuandong Li, and Chi Sing Leung, editors, *Neural Information Processing*, pages 637–646, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-34500-5_75.
- [BBV15] Céline Blondeau, Ash Bay, and Serge Vaudenay. Protecting against multi-dimensional linear and truncated differential cryptanalysis by decorrelation. In Gregor Leander, editor, *Fast Software Encryption*, pages 73–91, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. doi:10.1007/978-3-662-48116-5_4.
- [BDK⁺18] Joppe Bos, Leo Ducas, Eike Kiltz, T Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals - kyber: A cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367, 2018. doi:10.1109/EuroSP.2018.00032.
- [BFJ⁺94] Avrim Blum, Merrick L. Furst, Jeffrey C. Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 253–262. ACM, 1994. doi:10.1145/195058.195147.
- [BHK⁺99] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway. Umac: Fast and secure message authentication. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO’ 99*, pages 216–233, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-48405-1_14.
- [BK20] Seunggeun Baek and Kwangjo Kim. Recent advances of neural attacks against block ciphers. In *Symposium on Cryptography and Information Security*, 2020.
- [BLP⁺13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, STOC ’13*, page 575–584, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488608.2488680.
- [CKLS18] Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. Lizard: Cut off the tail! a practical post-quantum public-key encryption from lwe and lwr. In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks*, pages 160–177, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-98113-0_9.

- [CMLea22] Lily Chen, Dustin Moody, Yi-Kai Liu, and et al. Announcing four candidates to be standardized, plus fourth round candidates. <https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4>, 2022. Accessed: 2023-12-12.
- [CN11] Yuanmi Chen and Phong Q. Nguyen. Bkz 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 1–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi:10.1007/978-3-642-25385-0_1.
- [CW79] J.Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979. URL: <https://www.sciencedirect.com/science/article/pii/002200079900448>, doi:10.1016/0022-0000(79)90044-8.
- [CY21] Yi Chen and Hongbo Yu. Bridging machine learning and cryptanalysis via edlct. *Cryptology ePrint Archive*, Paper 2021/705, 2021. <https://eprint.iacr.org/2021/705>. URL: <https://eprint.iacr.org/2021/705>.
- [DNGW23] Elena Dubrova, Kalle Ngo, Joel Gärtner, and Ruize Wang. Breaking a fifth-order masked implementation of crystals-kyber by copy-paste. In *Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop, APKC '23*, page 10–20, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3591866.3593072.
- [Dra04] Sever S Dragomir. On the boas-bellman inequality in inner product spaces. *Bulletin of the Australian Mathematical Society*, 69(2):217–225, 2004. doi:10.1017/S0004972700035954.
- [FGV17] Vitaly Feldman, Cristóbal Guzmán, and Santosh S. Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1265–1277. SIAM, 2017. doi:10.1137/1.9781611974782.82.
- [FPY15] Hilary Finucane, Ron Peled, and Yariv Yaari. A recursive construction of t-wise uniform permutations. *Random Structures & Algorithms*, 46(3):531–540, 2015. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.20509>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.20509>, doi:10.1002/rsa.20509.
- [HKM18] Gottfried Herold, Elena Kirshanova, and Alexander May. On the asymptotic complexity of solving lwe. *Designs, Codes and Cryptography*, 86(1):55–83, Jan 2018. doi:10.1007/s10623-016-0326-0.
- [HMMR05] Shlomo Hoory, Avner Magen, Steven Myers, and Charles Rackoff. Simple permutations mix well. *Theoretical Computer Science*, 348(2):251–261, 2005. Automata, Languages and Programming: Algorithms and Complexity (ICALP-A 2004). URL: <https://www.sciencedirect.com/science/article/pii/S0304397505005360>, doi:10.1016/j.tcs.2005.09.016.
- [ITYY21] Mohamed Fadl Idris, Je Sen Teh, Jasy Liew Suet Yan, and Wei-Zhu Yeoh. A deep learning approach for active s-box prediction of lightweight generalized feistel block ciphers. *IEEE Access*, 9:104205–104216, 2021. doi:10.1109/ACCESS.2021.3099802.

- [JEP⁺21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Židek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstern, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, July 2021. doi:10.1038/s41586-021-03819-2.
- [Kea93] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 392–401. ACM, 1993. doi:10.1145/167088.167200.
- [KEI⁺22] Hayato Kimura, Keita Emura, Takanori Isobe, Ryoma Ito, Kazuto Ogawa, and Toshihiro Ohigashi. Output prediction attacks on block ciphers using deep learning. In *Applied Cryptography and Network Security Workshops*, pages 248–276, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-031-16815-4_15.
- [KNR09] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, Sep 2009. doi:10.1007/s00453-008-9267-y.
- [KR06] Ted Krovetz and Phillip Rogaway. Variationally universal hashing. *Information Processing Letters*, 100(1):36–39, 2006. URL: <https://www.sciencedirect.com/science/article/pii/S0020019006001566>, doi:10.1016/j.ipl.2005.11.026.
- [KV94] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, jan 1994. doi:10.1145/174644.174647.
- [KvHW19] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL: <https://openreview.net/forum?id=ByxBFsRqYm>.
- [LCK18] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, page 537–546, Red Hook, NY, USA, 2018. Curran Associates Inc. doi:10.5555/3326943.3326993.
- [LSW⁺23] Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin Lauter. Salsapicante: A machine learning attack on lwe with binary secrets. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS ’23*, page 2606–2620, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3576915.3623076.
- [LTV21] Tianren Liu, Stefano Tessaro, and Vinod Vaikuntanathan. The t-wise independence of substitution-permutation networks. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 454–483, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-84259-8_16.

- [LW06] Michael Luby and Avi Wigderson. Pairwise independence and derandomization. *Foundations and Trends® in Theoretical Computer Science*, 1(4):237–301, 2006. URL: <http://dx.doi.org/10.1561/0400000009>, doi:10.1561/0400000009.
- [LWAZ⁺23] Cathy Li, Emily Wenger, Zeyuan Allen-Zhu, Francois Charton, and Kristin E. Lauter. Salsa verde: a machine learning attack on lwe with sparse small secrets. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53343–53361. Curran Associates, Inc., 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/a75db7d2ee1e4bee8fb819979b0a6cad-Paper-Conference.pdf, doi:10.5555/3666122.3668444.
- [LYDD22] Zidu Liu, Li-Wei Yu, L.-M. Duan, and Dong-Ling Deng. Presence and absence of barren plateaus in tensor-network based machine learning. *Phys. Rev. Lett.*, 129:270501, Dec 2022. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.129.270501>, doi:10.1103/PhysRevLett.129.270501.
- [MBS⁺18] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, Nov 2018. doi:10.1038/s41467-018-07090-4.
- [NDJ23] Kalle Ngo, Elena Dubrova, and Thomas Johansson. A side-channel attack on a masked and shuffled software implementation of saber. *Journal of Cryptographic Engineering*, 13(4):443–460, Nov 2023. doi:10.1007/s13389-023-00315-3.
- [Ope22] OpenAI. Introducing ChatGPT. <https://openai.com/blog/chatgpt>, 2022. Accessed: 2023-05-30.
- [PSMF20] David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Phys. Rev. Res.*, 2:033429, Sep 2020. URL: <https://link.aps.org/doi/10.1103/PhysRevResearch.2.033429>, doi:10.1103/PhysRevResearch.2.033429.
- [RBA⁺19] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019. URL: <https://proceedings.mlr.press/v97/rahaman19a.html>.
- [Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery. doi:10.1145/1060590.1060603.
- [Sha18] Ohad Shamir. Distribution-specific hardness of learning neural networks. *J. Mach. Learn. Res.*, 19:32:1–32:29, 2018. doi:10.5555/3291125.3291157.
- [SHM⁺16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John

- Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. doi:10.1038/NATURE16961.
- [Sho05] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, USA, 2005. doi:10.5555/1529931.
- [SLB⁺19] Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. Learning a SAT solver from single-bit supervision. In *International Conference on Learning Representations*, 2019. URL: https://openreview.net/forum?id=HJMC_iA5tm.
- [SSS17] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3067–3075. PMLR, 2017. URL: <http://proceedings.mlr.press/v70/shalev-shwartz17a.html>, doi:10.5555/3305890.3305998.
- [TTJ23] Wei Jian Teng, Je Sen Teh, and Norziana Jamil. On the security of lightweight block ciphers against neural distinguishers: Observations on lbc-iot and slim. *Journal of Information Security and Applications*, 76:103531, 2023. URL: <https://www.sciencedirect.com/science/article/pii/S221421262301151>, doi:10.1016/j.jisa.2023.103531.
- [TTP⁺24] Rustem Takhanov, Maxat Tezkbayev, Artur Pak, Arman Bolatov, and Zhenisbek Assylbekov. Gradient descent fails to learn high-frequency functions and modular arithmetic, 2024. URL: <https://arxiv.org/abs/2310.12660>, arXiv:2310.12660.
- [Vau03] Serge Vaudenay. Decorrelation: A theory for block cipher security. *Journal of Cryptology*, 16(4):249–286, Sep 2003. doi:10.1007/s00145-003-0220-6.
- [WC81] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981. URL: <https://www.sciencedirect.com/science/article/pii/0022000081900337>, doi:10.1016/0022-0000(81)90033-7.
- [WCCL22] Emily Wenger, Mingjie Chen, François Charton, and Kristin E. Lauter. SALSA: attacking lattice cryptography with transformers. In *NeurIPS*, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/e28b3369186459f57c94a9ec9137fac9-Abstract-Conference.html.
- [Yan01] Ke Yang. On learning correlated boolean functions using statistical queries (extended abstract). In Naoki Abe, Roni Khardon, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, pages 59–76, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. doi:10.1007/3-540-45583-3_7.

A Application of Theorem 4 to the r th bit from the end

For $x \in \mathbb{Z}_q$, let $[x]_r \in \{0, 1\}$ denote the r th bit from the end of the binary encoding of x , i.e. $x = \sum_{i=1}^{\lceil \log_2 q \rceil} [x]_i 2^{i-1}$, $x_i \in \{0, 1\}$. Also, let us denote $q - 2^{r-1} \lfloor \frac{q}{2^{r-1}} \rfloor$ by q_{r-1} . Note

that $0 \leq q_{r-1} < 2^{r-1}$ and q_{r-1} is just a number whose binary encoding forms the last $r-1$ bits of q . For $a : \mathbb{N}^l \rightarrow \mathbb{R}$ and $b : \mathbb{N}^l \rightarrow \mathbb{R}^+$, $a \ll b$ denotes that there exists $\alpha, \beta > 0$ such that for all $\mathbf{n} = (n_1, \dots, n_l) \in \mathbb{N}^l$ satisfying $n_1 + \dots + n_l > \beta$ we have $|a(\mathbf{n})| \leq \alpha b(\mathbf{n})$. If $a \ll b$ and $b \ll a$, we simply write $a \asymp b$. \mathbb{Z}_q^* denotes $\{0, \dots, q-1\}$. We will prove the following bound.

Theorem 5. *For any natural r such that $1 \leq r \leq \lceil \log_2 q \rceil$, we have $\mathbb{E}_{\omega \sim \mathbb{Z}_q^*} [|\widehat{[\cdot]_r}(\omega)|] \ll r(\log_2 q + 1 - r)$.*

Let us denote $t_r(x) = (-1)^{[x]_r}$, $x \in \mathbb{Z}_q$. The DFT of t_r is easier to calculate than that of $[\cdot]_r$, so we will deal with t_r first.

Lemma 4. *We have*

$$\widehat{t}_r(\omega) = \frac{1 - (-1)^{[q]_r} \varepsilon^{-(q-q_{r-1})\omega}}{1 + \varepsilon^{-2^{r-1}\omega}} \frac{1 - \varepsilon^{-2^{r-1}\omega}}{1 - \varepsilon^{-\omega}} + (-1)^{[q]_r} \frac{\varepsilon^{-(q-q_{r-1})\omega} - \varepsilon^{-q\omega}}{1 - \varepsilon^{-\omega}},$$

for $\omega \in \mathbb{Z}_q^*$.

Proof. Note that

$$\widehat{t}_r(\omega) = \sum_{x=0}^{q-1} \varepsilon^{-x\omega} (-1)^{[x]_r} = \sum_{x=0}^{2^{r-1} \lfloor \frac{q}{2^{r-1}} \rfloor - 1} \varepsilon^{-x\omega} (-1)^{[x]_r} + \sum_{x=2^{r-1} \lfloor \frac{q}{2^{r-1}} \rfloor}^{q-1} \varepsilon^{-x\omega} (-1)^{[x]_r}.$$

The first term equals

$$\begin{aligned} \sum_{x=0}^{2^{r-1} \lfloor \frac{q}{2^{r-1}} \rfloor - 1} \varepsilon^{-x\omega} (-1)^{[x]_r} &= \sum_{t=0}^{\lfloor \frac{q}{2^{r-1}} \rfloor - 1} (-1)^t \varepsilon^{-2^{r-1}t\omega} \sum_{x=0}^{2^{r-1}-1} \varepsilon^{-x\omega} = \\ &= \frac{1 - (-1)^{[q]_r} \varepsilon^{-(q-q_{r-1})\omega}}{1 + \varepsilon^{-2^{r-1}\omega}} \frac{1 - \varepsilon^{-2^{r-1}\omega}}{1 - \varepsilon^{-\omega}}, \end{aligned}$$

if $\omega \neq 0$. If $\omega = 0$, then it equals $2^{r-1} [q]_r$. Whereas, the second term equals

$$(-1)^{[q]_r} \sum_{x=q-q_{r-1}}^{q-1} \varepsilon^{-x\omega} = (-1)^{[q]_r} \frac{\varepsilon^{-(q-q_{r-1})\omega} - \varepsilon^{-q\omega}}{1 - \varepsilon^{-\omega}},$$

if $\omega \neq 0$. If $\omega = 0$, then it equals $(-1)^{[q]_r} q_{r-1}$. Thus, we conclude that

$$\widehat{t}_r(\omega) = \frac{1 - (-1)^{[q]_r} \varepsilon^{-(q-q_{r-1})\omega}}{1 + \varepsilon^{-2^{r-1}\omega}} \frac{1 - \varepsilon^{-2^{r-1}\omega}}{1 - \varepsilon^{-\omega}} + (-1)^{[q]_r} \frac{\varepsilon^{-(q-q_{r-1})\omega} - \varepsilon^{-q\omega}}{1 - \varepsilon^{-\omega}},$$

if $\omega \in \mathbb{Z}_q^*$. □

For bounding $\mathbb{E}_{\omega \sim \mathbb{Z}_q^*} [|\widehat{t}_r(\omega)|]$ we will need the following technical lemma.

Lemma 5. *We have $\sum_{k \in \mathbb{Z}_q} \frac{1}{|1 + \varepsilon^{rk}|} \ll q \log q$ and $\sum_{k \in \mathbb{Z}_q^*} \frac{1}{|1 - \varepsilon^{rk}|} \ll q \log q$ for any $r \in \mathbb{Z}_q^*$.*

Proof. Since $\sum_{k \in \mathbb{Z}_q} \frac{1}{|1 + \varepsilon^{rk}|} = \sum_{k \in \mathbb{Z}_q} \frac{1}{|1 + \varepsilon^k|}$, it is enough to prove the first statement for $r = 1$.

Let $\theta = \frac{2\pi k}{q}$. Note that $\theta \in [0, 2\pi)$ and

$$|1 + \varepsilon^k| = |1 + e^{i\theta}| = (2 + 2 \cos(\theta))^{1/2} = 2 \left| \cos\left(\frac{\theta}{2}\right) \right|.$$

Let us denote $2\psi = \theta - \pi$. Thus, $|1 + \varepsilon^k| = 2 \left| \cos\left(\frac{2\psi + \pi}{2}\right) \right| = 2|\sin(\psi)| \geq |\psi|$ if $\psi \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. Note that $\psi \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ if and only if $-\frac{\pi}{4} \leq \frac{\pi k}{q} - \frac{\pi}{2} \leq \frac{\pi}{4}$, or $\frac{1}{4}q \leq k \leq \frac{3}{4}q$. Thus, we have

$$\begin{aligned} \sum_{k \in [\frac{1}{4}q, \frac{3}{4}q] \cap \mathbb{Z}_q} \frac{1}{|1 + \varepsilon^k|} &\leq \sum_{k \in [\frac{1}{4}q, \frac{3}{4}q] \cap \mathbb{Z}_q} \frac{1}{\left| \frac{\pi k}{q} - \frac{\pi}{2} \right|} = \\ \frac{2q}{\pi} \sum_{k \in [\frac{1}{4}q, \frac{3}{4}q] \cap \mathbb{Z}_q} \frac{1}{|2k - q|} &\leq \frac{4q}{\pi} \sum_{i=1}^{\lceil q/2 \rceil} \frac{1}{i} \ll q \log q. \end{aligned}$$

Since $|1 + \varepsilon^k| = 2|\sin(\psi)| \geq 1$ if $\psi \in [-\frac{\pi}{2}, -\frac{\pi}{4}] \cup [\frac{\pi}{4}, \frac{\pi}{2}]$, then

$$\sum_{k \in \mathbb{Z}_q: \psi \in [-\frac{\pi}{2}, -\frac{\pi}{4}] \cup [\frac{\pi}{4}, \frac{\pi}{2}]} \frac{1}{|1 + \varepsilon^k|} \asymp q.$$

Thus, the total sum satisfies

$$\sum_{k \in \mathbb{Z}_q} \frac{1}{|1 + \varepsilon^k|} \ll q \log q.$$

Let us now prove the second statement, i.e. $\sum_{k \in \mathbb{Z}_q^*} \frac{1}{|1 - \varepsilon^{rk}|} \ll q \log q$. Again, it is enough to prove it for $r = 1$. Since

$$\sum_{k \in \mathbb{Z}_q^*} \frac{1}{|1 - \varepsilon^k|} = \sum_{k=-\frac{q-1}{2}}^{-1} \frac{1}{|1 - \varepsilon^k|} + \sum_{k=1}^{\frac{q-1}{2}} \frac{1}{|1 - \varepsilon^k|},$$

let us prove first that $\sum_{k=1}^{\frac{q-1}{2}} \frac{1}{|1 - \varepsilon^k|} \ll q \log q$.

Let $\theta = \frac{2\pi k}{q}$, $1 \leq k \leq \frac{q-1}{2}$. Note that $\theta \in (0, \pi)$ and

$$|1 - \varepsilon^k| = |1 - e^{i\theta}| = (2 - 2\cos(\theta))^{1/2} = 2 \left| \sin\left(\frac{\theta}{2}\right) \right|.$$

Let us denote $2\psi = \theta$. The condition $0 < \psi \leq \frac{\pi}{4}$ is equivalent to $1 \leq k \leq \frac{q}{4}$. Under that condition, we have $2|\sin(\psi)| \geq |\psi|$. Therefore, we have

$$\sum_{k \in [1, \frac{q}{4}] \cap \mathbb{Z}_q} \frac{1}{|1 - \varepsilon^k|} \leq \sum_{k \in [1, \frac{q}{4}] \cap \mathbb{Z}_q} \frac{1}{\left| \frac{\pi k}{q} \right|} = \frac{q}{\pi} \sum_{k \in [1, \frac{q}{4}] \cap \mathbb{Z}_q} \frac{1}{k} \ll q \log q.$$

Also, $2|\sin(\psi)| \geq 1$, $\psi \in [\frac{\pi}{4}, \frac{\pi}{2})$, therefore

$$\sum_{k \in [\frac{q}{4}, \frac{q-1}{2}] \cap \mathbb{Z}_q} \frac{1}{|1 - \varepsilon^k|} \leq \sum_{k \in [1, \frac{q}{4}] \cap \mathbb{Z}_q} \frac{1}{1} \ll q.$$

Thus, $\sum_{k=1}^{\frac{q-1}{2}} \frac{1}{|1 - \varepsilon^k|} \ll q \log q$. Analogously one can prove $\sum_{k=-\frac{q-1}{2}}^{-1} \frac{1}{|1 - \varepsilon^k|} \ll q \log q$. \square

Now everything is ready to estimate the sum $\sum_{\omega \in \mathbb{Z}_q^*} |\hat{t}_r(\omega)|$ which is made in the following lemma.

Lemma 6. *We have $\sum_{\omega \in \mathbb{Z}_q^*} |\hat{t}_r(\omega)| \ll qr(\log_2 q + 1 - r)$.*

Proof. First let us consider the case of $[q]_r = 0$. Using Lemma 4, we have

$$\sum_{\omega \in \mathbb{Z}_q^*} |\widehat{t}_r(\omega)| \leq \sum_{\omega \in \mathbb{Z}_q^*} \frac{|1 - \varepsilon^{-2^{r-1} \lfloor \frac{q}{2^{r-1}} \rfloor \omega}| |1 - \varepsilon^{-2^{r-1} \omega}|}{|1 + \varepsilon^{-2^{r-1} \omega}| |1 - \varepsilon^{-\omega}|} + \frac{2}{|1 - \varepsilon^{-\omega}|}.$$

From Lemma 5 we have $\sum_{\omega \in \mathbb{Z}_q^*} \frac{2}{|1 - \varepsilon^{-\omega}|} \ll q \log q$. Thus, it remains to bound the sum of terms $a_\omega = \frac{|1 - \varepsilon^{-2^{r-1} \lfloor \frac{q}{2^{r-1}} \rfloor \omega}| |1 - \varepsilon^{-2^{r-1} \omega}|}{|1 + \varepsilon^{-2^{r-1} \omega}| |1 - \varepsilon^{-\omega}|}$. Using $\sum_{\omega \in \mathbb{Z}_q^*} a_\omega = \sum_{\omega=1}^{\frac{q-1}{2}} a_\omega + \sum_{\omega=-\frac{q-1}{2}}^{-1} a_\omega$ and $\frac{1}{|1 - \varepsilon^{-\omega}|} \ll 1$ for $\frac{q}{4} \leq \omega \leq \frac{q-1}{2}$ or $-\frac{q-1}{2} \leq \omega \leq -\frac{q}{4}$ we conclude that

$$\sum_{\omega=\lceil \frac{q}{4} \rceil}^{\frac{q-1}{2}} a_\omega \ll 4 \sum_{\omega=\lceil \frac{q}{4} \rceil}^{\frac{q-1}{2}} \frac{1}{|1 + \varepsilon^{-2^{r-1} \omega}|} \ll q \log q$$

and $\sum_{\omega=-\frac{q-1}{2}}^{-\lceil \frac{q}{4} \rceil} a_\omega \ll q \log q$ (using Lemma 5). Thus, a bound of the total sum directly follows from bounds of $\sum_{\omega=1}^{\lfloor \frac{q}{4} \rfloor} a_\omega$ (and $\sum_{\omega=-\lfloor \frac{q}{4} \rfloor}^{-1} a_\omega$). For brevity, let us only show how to bound $U = \sum_{\omega=1}^{\lfloor \frac{q}{4} \rfloor} a_\omega$.

Using $|1 - \varepsilon^x| = 2|\sin(\frac{\pi x}{q})|$ and $|1 + \varepsilon^x| = 2|\cos(\frac{\pi x}{q})|$, this sum can be rewritten as

$$U = \sum_{i=1}^{\lfloor \frac{q}{4} \rfloor} \frac{|\sin(2^r k \frac{\pi i}{q})| \cdot |\sin(2^{r-1} \frac{\pi i}{q})|}{|\cos(2^{r-1} \frac{\pi i}{q}) \sin(\frac{\pi i}{q})|}.$$

where $2k = \lfloor \frac{q}{2^{r-1}} \rfloor$. For arbitrary x and $y > 0$, let us denote the interval $[x - y, x + y]$ by $x \pm y$. Also, $[s]$ denote $\{1, \dots, s\}$. Let us denote $n = \lfloor 2^{r-2} \frac{i}{q} \rfloor$. By construction, we have $0 \leq n \leq 2^{r-4} - 1$. For any $i = 1, \dots, \lfloor \frac{q}{4} \rfloor$ at least one of the following inclusions holds

- 1) $2^{r-1} \frac{\pi i}{q} \in \frac{\pi}{2} + 2\pi n \pm \frac{\pi}{4}$,
- 2) $2^{r-1} \frac{\pi i}{q} \in \frac{3\pi}{2} + 2\pi n \pm \frac{\pi}{4}$, or
- 3) $2^{r-1} \frac{\pi i}{q} \notin (\frac{\pi}{2} + 2\pi n \pm \frac{\pi}{4}) \cup (\frac{3\pi}{2} + 2\pi n \pm \frac{\pi}{4})$.

In the third case we have $\frac{1}{|\cos(2^{r-1} \frac{\pi i}{q})|} \ll 1$ and the summation over all such i asymptotically cannot exceed $\sum_{i=1}^{\lfloor \frac{q}{4} \rfloor} \frac{1}{|\sin(\frac{\pi i}{q})|} \ll \sum_{i=1}^{\lfloor \frac{q}{4} \rfloor} \frac{1}{|\frac{\pi i}{q}|} \ll q \log q$. The summation over terms that satisfy either 1) or 2) are similar, therefore we will consider only the first case, i.e. we will bound

$$\tilde{U} = \sum_{n=0}^{2^{r-4}-1} \sum_{i \in [\lfloor \frac{q}{4} \rfloor]: 2^{r-1} \frac{\pi i}{q} \in \frac{\pi}{2} + 2\pi n \pm \frac{\pi}{4}} \frac{|\sin(2^r k \frac{\pi i}{q})| \cdot |\sin(2^{r-1} \frac{\pi i}{q})|}{|\cos(2^{r-1} \frac{\pi i}{q}) \sin(\frac{\pi i}{q})|}.$$

We have $2^{r-1} \frac{\pi i}{q} \in \frac{\pi}{2} + 2\pi n \pm \frac{\pi}{4}$, $i \in [\lfloor \frac{q}{4} \rfloor]$ if and only if $i \in [\lfloor \frac{q}{4} \rfloor] \cap \frac{q}{2r} + \frac{nq}{2^{r-2}} \pm \frac{q}{2^{r+1}}$. Let us denote $e = i - (\frac{q}{2r} + \frac{nq}{2^{r-2}})$. From $i \in [\lfloor \frac{q}{4} \rfloor] \cap \frac{q}{2r} + \frac{nq}{2^{r-2}} \pm \frac{q}{2^{r+1}}$ we deduce $e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z} - \{\frac{q}{2r} + \frac{nq}{2^{r-2}}\})$ where $\{x\} = x - [x]$ and $\mathbb{Z} - s$ denotes $\{z - s \mid z \in \mathbb{Z}\}$.

Using $2|\cos(2^{r-1}\frac{\pi i}{q})| \geq |2^{r-1}\frac{\pi i}{q} - \frac{\pi}{2} - 2\pi n|$ for $2^{r-1}\frac{\pi i}{q} \in \frac{\pi}{2} + 2\pi n \pm \frac{\pi}{4}$ we obtain

$$\begin{aligned} \tilde{U} &\ll \sum_{n=0}^{2^{r-2}-1} \sum_{i \in [\frac{q}{4}] \cap \frac{q}{2^r} + \frac{nq}{2^{r-2}} \pm \frac{q}{2^{r+1}}} \frac{|\sin(2^r k \frac{\pi i}{q}) \sin(2^{r-1} \frac{\pi i}{q})|}{|(2^{r-1} \frac{\pi i}{q} - \frac{\pi}{2} - 2\pi n) \frac{\pi i}{q}|} \leq \\ &\sum_{n=0}^{2^{r-2}-1} \sum_{i \in [\frac{q}{4}] \cap \frac{q}{2^r} + \frac{nq}{2^{r-2}} \pm \frac{q}{2^{r+1}}} \frac{|\sin(2^r k \frac{\pi i}{q})|}{|(2^{r-1} \frac{\pi i}{q} - \frac{\pi}{2} - 2\pi n) \frac{\pi i}{q}|} \leq \\ &\sum_{n=0}^{2^{r-2}-1} \sum_{e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z} - \{\frac{q}{2^r} + \frac{nq}{2^{r-2}}\})} \frac{|\sin(2^r k \frac{\pi e}{q})|}{\frac{2^{r-1}\pi}{q} |e| (\frac{\pi}{2^r} + \frac{\pi n}{2^{r-2}} + \frac{\pi}{q} e)}. \end{aligned}$$

Let us denote $x = \frac{\pi 2^r k}{q}$. Note that $\frac{\pi}{2} \leq \frac{\pi 2^r k}{q} \leq \pi$. We have

$$\sum_{e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z} - \{\frac{q}{2^r} + \frac{nq}{2^{r-2}}\})} \frac{|\sin(xe)|}{\frac{2^{r-1}\pi}{q} |e| (\frac{\pi}{2^r} + \frac{\pi n}{2^{r-2}} + \frac{\pi}{q} e)} \ll \frac{C_r}{\frac{2^{r-1}\pi}{q} (\frac{\pi}{2^{r+1}} + \frac{\pi n}{2^{r-2}})},$$

where

$$C_r = \max_{s \in [0,1]} \sum_{e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z} - s)} \frac{|\sin(xe)|}{|e|}.$$

Obviously, we have $C_r \leq \sum_{e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z} - s^*)} \frac{|\sin(xe)|}{|e|}$ for some $s^* \in [0,1]$, and the latter is bounded by $2x + 2 \sum_{i=1}^{\lceil \frac{q}{2^{r+1}} \rceil} \frac{1}{i} \ll \log(\frac{q}{2^{r+1}} + 1)$.

Thus, \tilde{U} is asymptotically bounded by

$$qC_r \sum_{n=0}^{2^{r-2}-1} \frac{1}{2^{r-1}(\frac{1}{2^{r+1}} + \frac{n}{2^{r-2}})} \ll q \log_2(\frac{q}{2^{r+1}} + 1) \log_2(2^{r-2} + 1) \ll qr(\log_2 q + 1 - r),$$

and therefore, the total sum is bounded by $qr(\log_2 q + 1 - r)$.

Let us now consider the case of $[q]_r = 1$. As in the previous case, we can reduce bounding the total sum to bounding the sum $V = \sum_{i=1}^{\lfloor \frac{q}{4} \rfloor} b_i$ where $b_i = \frac{|1+\varepsilon^{-2^{r-1}\lfloor \frac{q}{2^{r-1}} \rfloor i}| |1-\varepsilon^{-2^{r-1}i}|}{|1+\varepsilon^{-2^{r-1}i}| |1-\varepsilon^{-i}|}$, which is equal to

$$V = \sum_{i=1}^{\lfloor \frac{q}{4} \rfloor} \frac{|\cos(2^{r-1}(2k+1)\frac{\pi i}{q})| \cdot |\sin(2^{r-1}\frac{\pi i}{q})|}{|\cos(2^{r-1}\frac{\pi i}{q}) \sin(\frac{\pi i}{q})|}.$$

where $2k+1 = \lfloor \frac{q}{2^{r-1}} \rfloor$. As in the previous case, V can be bounded according to

$$\begin{aligned} V &\ll \sum_{n=0}^{2^{r-2}-1} \sum_{i \in [\frac{q}{4}] \cap \frac{q}{2^r} + \frac{nq}{2^{r-2}} \pm \frac{q}{2^{r+1}}} \frac{|\cos(2^{r-1}(2k+1)\frac{\pi i}{q}) \sin(2^{r-1}\frac{\pi i}{q})|}{|(2^{r-1}\frac{\pi i}{q} - \frac{\pi}{2} - 2\pi n) \frac{\pi i}{q}|} \leq \\ &\sum_{n=0}^{2^{r-2}-1} \sum_{i \in [\frac{q}{4}] \cap \frac{q}{2^r} + \frac{nq}{2^{r-2}} \pm \frac{q}{2^{r+1}}} \frac{|\cos(2^{r-1}(2k+1)\frac{\pi i}{q})|}{|(2^{r-1}\frac{\pi i}{q} - \frac{\pi}{2} - 2\pi n) \frac{\pi i}{q}|} \leq \\ &\sum_{n=0}^{2^{r-2}-1} \sum_{e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z} - \{\frac{q}{2^r} + \frac{nq}{2^{r-2}}\})} \frac{|\sin(2^{r-1}(2k+1)\frac{\pi e}{q})|}{\frac{2^{r-1}\pi}{q} |e| (\frac{\pi}{2^r} + \frac{\pi n}{2^{r-2}} + \frac{\pi}{q} e)}. \end{aligned}$$

The latter sum is asymptotically bounded by

$$qC'_r \sum_{n=0}^{2^{r-2}-1} \frac{1}{2^{r-1} \left(\frac{1}{2^{r+1}} + \frac{n}{2^{r-2}} \right)}$$

where $C'_r = \max_{s \in [0,1]} \sum_{e \in \pm \frac{q}{2^{r+1}} \cap (\mathbb{Z}-s)} \frac{|\sin(x'e)|}{|e|}$, $x' = \frac{2^{r-1}(2k+1)\pi}{q}$ and $C'_r \ll (\log_2 q + 1 - r)$. Thus, $V \ll qr(\log_2 q + 1 - r)$. \square

Proof of Theorem 5. After noting that $t_r(x) = 1 - 2[x]_r$, we obtain $|\widehat{t}_r(\omega)| = 2|\widehat{[\cdot]}_r(\omega)|$ for $\omega \in \mathbb{Z}_q^*$, and therefore,

$$\mathbb{E}_{\omega \sim \mathbb{Z}_q^*} [|\widehat{[\cdot]}_r(\omega)|] = \frac{1}{2} \mathbb{E}_{\omega \sim \mathbb{Z}_q^*} [|\widehat{t}_r(\omega)|].$$

Then, from Lemma 6 we conclude that

$$\mathbb{E}_{\omega \sim \mathbb{Z}_q^*} [|\widehat{[\cdot]}_r(\omega)|] \ll r(\log_2 q + 1 - r). \quad \square$$

B Calculation of BB_S for different S

Let us first show that the variance that we are interested in is indeed bounded by a factor of BB_S . Recall that $L(p, y) = \frac{1}{2}q^{-2}(p - y)^2$ and $S \subseteq \mathbb{Z}_q^* \setminus \{\mathbf{0}\}$. Then, we have

$$\text{Var}_{\mathbf{a} \sim S} [\partial_{w_i} \mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} L(p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle)] \leq q^{-4} \mathbb{E}_{\mathbf{a} \sim S} [\langle \partial_{w_i} p(\mathbf{w}, \mathbf{x}), \langle \mathbf{a}, \mathbf{x} \rangle - \frac{q-1}{2} \rangle_{\mathbf{x}}^2].$$

Using the Boas-Bellman inequality we bound the latter expectation in the following way,

$$\begin{aligned} & |S|^{-1} \sum_{\mathbf{a} \in S} \langle \langle \mathbf{a}, \mathbf{x} \rangle - \frac{q-1}{2}, \partial_{w_i} p(\mathbf{w}, \mathbf{x}) \rangle_{\mathbf{x}}^2 \leq \\ & |S|^{-1} \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 \left[\left(\frac{q-1}{2} \right)^2 + \left(\sum_{\mathbf{k} \neq \mathbf{k}' \in S} \langle \langle \mathbf{k}, \mathbf{x} \rangle - \frac{q-1}{2}, \langle \mathbf{k}', \mathbf{x} \rangle - \frac{q-1}{2} \rangle_{\mathbf{x}} \right)^2 \right]^{1/2} \leq \\ & \|\partial_{w_i} p(\mathbf{w}, \mathbf{x})\|_{\mathbf{x}}^2 \left[\frac{(q-1)^2}{4|S|} + \mathbb{E}_{\mathbf{k} \neq \mathbf{k}' \sim S} [\mathbb{E}_{\mathbf{x} \sim \mathbb{Z}_q^n} [\langle \langle \mathbf{k}, \mathbf{x} \rangle - \frac{q-1}{2}, \langle \mathbf{k}', \mathbf{x} \rangle - \frac{q-1}{2} \rangle_{\mathbf{x}}^2]^{1/2} \right]. \end{aligned}$$

For $\lambda \in \mathbb{Z}_q^* \setminus \{1\}$, let us denote $\mathbb{P}_{\mathbf{k} \neq \mathbf{k}' \sim S} [\mathbf{k}' = \lambda \mathbf{k}]$ by $p(\lambda)$, and $\mathbb{E}_{x \sim \mathbb{Z}_q} [(x - \frac{q-1}{2})((\lambda x \bmod q) - \frac{q-1}{2})]$ by $r(\lambda)$. We need these two functions due to

$$\begin{aligned} & \mathbb{E}_{\mathbf{k} \neq \mathbf{k}' \sim S} \left[\langle \langle \mathbf{k}, \mathbf{x} \rangle - \frac{q-1}{2}, \langle \mathbf{k}', \mathbf{x} \rangle - \frac{q-1}{2} \rangle_{\mathbf{x}}^2 \right] = \\ & \mathbb{P}_{\mathbf{k} \neq \mathbf{k}' \sim S} [\text{rank}[\mathbf{k}, \mathbf{k}'] = 2] \times 0^2 + \sum_{\lambda \in \mathbb{Z}_q^* \setminus \{1\}} \mathbb{P}_{\mathbf{k} \neq \mathbf{k}' \sim S} [\mathbf{k}' = \lambda \mathbf{k}] r(\lambda)^2 = \sum_{\lambda \in \mathbb{Z}_q^* \setminus \{1\}} p(\lambda) r(\lambda)^2. \end{aligned}$$

In other words, the calculation of $p(\lambda)$ and $r(\lambda)$ allows us to calculate

$$\text{BB}_S = \frac{(q-1)^2}{4q^4|S|} + q^{-4} \left(\sum_{\lambda \in \mathbb{Z}_q^* \setminus \{1\}} p(\lambda) r(\lambda)^2 \right)^{1/2}.$$

The function $r(\lambda)$ is the same for any S and it can be computed numerically. The situation for the function $p(\lambda)$ is trickier, it should be computed distinctly for each S based on the fact that for each $\lambda \in \mathbb{Z}_q^* \setminus \{1\}$ we have

$$\mathbb{P}_{\mathbf{k}' \sim S \setminus \{\mathbf{k}\}} [\mathbf{k}' = \lambda \mathbf{k} \mid \mathbf{k}] = [\lambda \mathbf{k} \in S \setminus \{\mathbf{k}\}] (|S| - 1)^{-1},$$

and

$$p(\lambda) = |S|^{-1} \sum_{\mathbf{k} \in S} \mathbb{P}_{\mathbf{k}' \sim S \setminus \{\mathbf{k}\}}[\mathbf{k}' = \lambda \mathbf{k} \mid \mathbf{k}] = (|S| - 1)^{-1} \mathbb{P}_{\mathbf{k} \sim S}[\lambda \mathbf{k} \in S].$$

After we plug the latter expression into the second term of BB_S , we obtain

$$\frac{1}{q^4 \sqrt{|S| - 1}} \left(\sum_{\lambda \in \mathbb{Z}_q^* \setminus \{1\}} \mathbb{P}_{\mathbf{k} \sim S}[\lambda \mathbf{k} \in S] r(\lambda)^2 \right)^{1/2}.$$

and conclude that it is $\mathcal{O}(|S|^{-1/2} q^{-2})$.

For the case (a) we have $\mathbb{P}_{\mathbf{k} \sim S}[\lambda \mathbf{k} \in S] = 1$. In case (b) and (c) we have $\mathbb{P}_{\mathbf{k} \sim S}[\lambda \mathbf{k} \in S] = 0$. In case (d) and (e) we have $\mathbb{P}_{\mathbf{k} \sim S}[\lambda \mathbf{k} \in S] = 1$ if $\lambda = -1$ and $\mathbb{P}_{\mathbf{k} \sim S}[\lambda \mathbf{k} \in S] = 0$ if otherwise. Thus, we obtained

$$\text{BB}_S = \begin{cases} \frac{(q-1)^2}{4q^4|S|} + q^{-4}(|S| - 1)^{-\frac{1}{2}} \left(\sum_{\lambda \in \mathbb{Z}_q^* \setminus \{1\}} r(\lambda)^2 \right)^{\frac{1}{2}}, & \text{for (a)} \\ \frac{(q-1)^2}{4q^4|S|}, & \text{for (b)} \\ \frac{(q-1)^2}{4q^4|S|}, & \text{for (c)} \\ \frac{(q-1)^2}{4q^4|S|} + q^{-4}(|S| - 1)^{-\frac{1}{2}} |r(-1)|, & \text{for (d)} \\ \frac{(q-1)^2}{4q^4|S|} + q^{-4}(|S| - 1)^{-\frac{1}{2}} |r(-1)|, & \text{for (e)} \end{cases}$$

Obtained formulas were implemented in the form of a Python code that can be accessed at [github](#).