Intractability of Learning the Discrete Logarithm with Gradient-Based Methods

Rustem Takhanov RUSTEM.TAKHANOV@NU.EDU.KZ
Maxat Tezekbayev MAXAT.TEZEKBAYEV@NU.EDU.KZ
Artur Pak ARTUR.PAK@NU.EDU.KZ
Department of Mathematics, Nazarbayev University, Astana, Kazakhstan

Arman Bolatov ARMAN.BOLATOV@NU.EDU.KZ
Department of Computer Science, Nazarbayev University, Astana, Kazakhstan

Zhibek Kadyrsizova ZHIBEK.KADYRSIZOVA@NU.EDU.KZ
Department of Mathematics, Nazarbayev University, Astana, Kazakhstan

Zhenisbek Assylbekov ZASSYLBE@PFW.EDU
Department of Mathematical Sciences, Purdue University Fort Wayne, Fort Wayne, IN, USA

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

The discrete logarithm problem is a fundamental challenge in number theory with significant implications for cryptographic protocols. In this paper, we investigate the limitations of gradient-based methods for learning the parity bit of the discrete logarithm in finite cyclic groups of prime order. Our main result, supported by theoretical analysis and empirical verification, reveals the concentration of the gradient of the loss function around a fixed point, independent of the logarithm's base used. This concentration property leads to a restricted ability to learn the parity bit efficiently using gradient-based methods, irrespective of the complexity of the network architecture being trained.

Our proof relies on Boas-Bellman inequality in inner product spaces and it involves establishing approximate orthogonality of discrete logarithm's parity bit functions through the spectral norm of certain matrices. Empirical experiments using a neural network-based approach further verify the limitations of gradient-based learning, demonstrating the decreasing success rate in predicting the parity bit as the group order increases.

Keywords: Discrete Logarithm, Gradient-based Learning, Cryptographic Protocols.

1. Introduction

Today, artificial intelligence is able to solve problems that seemed extremely difficult for machines 10 years ago. The most famous success stories include the victory of the machine over a professional Go player (Silver et al., 2016); prediction of the spatial structure of a protein with high accuracy (Jumper et al., 2021); a chatbot capable of working in a conversational mode, supporting requests in natural languages (OpenAI, 2022). Since all these stories are based on deep neural networks trained by gradient-based methods, it may seem that at this pace there will soon be no problems that would be beyond the capacity of gradient-based learning.

However, Shalev-Shwartz et al. (2017) showed the failure of gradient-based methods to learn some rather simple functions such as the parity function. In this paper, we give an example of another simple function that gradient-based methods provably cannot learn. This function is a parity bit of the **discrete logarithm** in the additive group of integers modulo p. It is known that with the help of the extended Euclidean algorithm, the discrete logarithm in this group can be computed in $O(n)$ operations, where n is the bitlength of p (see Section 2 for details). So there is a Boolean circuit with complexity $O(n^2)$, which implements a parity bit of the discrete logarithm (with a fixed base). Since each logic gate can be implemented by a small number of neurons and weights, this circuit can be converted into a compact neural network with $\mathrm{poly}(n)$ parameters. However, we prove formally that gradient-based methods cannot efficiently train such a network.

In fact, we have obtained a more general result (Theorem 1) which says that when trying to learn the parity bit of the discrete logarithm in *any* finite cyclic group of prime order, and not only in $(\mathbb{Z}_p, +)$, the gradient carries negligible information about the target function. It has long been known in cryptography that the discrete logarithm problem (DLP) in a carefully chosen cyclic group (for example, in the group of points on an elliptic curve (Miller, 1985)) is hard in the sense that at the moment there is no $\mathrm{poly}(n)$ algorithm for solving the DLP in general case. The intractability of the DLP in such groups forms the basis for various cryptographic protocols, including public-key encryption, digital signatures (Gamal, 1985), and key exchange (Diffie and Hellman, 1976). From this point of view, our result is the **provable security of DLP-based cryptosystems against gradient-based attacks**.

1.1. Related Work

The main source of inspiration for us is the work of Shalev-Shwartz et al. (2017), which, among other things, shows the intractability of learning a class of orthogonal functions using gradient-based methods. We emphasize that their result is not directly applicable to the class of functions that we consider in this paper (the parity bit of the discrete logarithm), since these functions are not orthogonal with respect to a uniform distribution over the domain. However, they are *approximately* pairwise orthogonal, and the proof of this fact is the core of our work (Section 5). In addition, our adaptation of the proof method by Shalev-Shwartz et al. (2017) using the Boas-Bellman inequality (Section 5.3) deserves special attention, as it allows us to extend the failure of gradient-based learning to a wider class of approximately orthogonal functions.

It should be noted that the relationship between orthogonal functions and hardness of learning is not new and has been established in the context of statistical query (SQ) learning model of Kearns (1993). Moreover, this relationship was characterized by Blum et al. (1994) in terms of the statistical dimension of the function class, which essentially corresponds to the largest possible set of functions in the class which are all *approximately* pairwise orthogonal. The hardness of learning a class of boolean functions in an SQ model is usually proven through a lower bound on the statistical dimension of the class. It is noteworthy that gradient-based learning with an approximate gradient oracle can be implemented through the SQ algorithm (Feldman et al., 2017), which means that our result on the approximate orthogonality of the considered class of functions (Lemma 12) immediately gives the hardness of learning this class with gradient-based methods. Nevertheless, we believe that the proof of this result directly (without resorting to the SQ proxy) deserves attention, since it allows us to establish that the low information content of the gradient is the very reason why gradient learning fails.

Theorem 1 of Liu et al. (2021) states that assuming the classical hardness of the DLP, no efficient classical algorithm can learn the concept class constructed by the authors. Thus their result, although applicable to all classical learning algorithms, is conditioned by a strong assumption. In our paper, we show the *unconditional* hardness of learning the parity bit of a discrete logarithm by any gradient-based method (e.g., SGD, RMSProp, Adam, etc.), i.e. we do not make any assumptions on the hardness of the DLP itself.

1.2. Notation

Bold-faced lowercase letters ($\mathbf{x}$) denote vectors, bold-faced uppercase letters ($\mathbf{A}$) denote matrices. Regular lowercase letters (x) denote scalars (or set elements), and regular uppercase letters (X) denote random variables (or random elements). $\|\cdot\|$ denotes the Euclidean norm: $\|\mathbf{x}\| := \sqrt{\mathbf{x}^\top \mathbf{x}}$. For $\mathbf{x} \in \mathbb{C}^n$, conjugate transpose is denoted by $\mathbf{x}^\dagger$. For any finite set $\mathcal{S}$, sampling X uniformly from $\mathcal{S}$ is denoted by $X \sim \mathcal{S}$. For two functions f, g on a finite set $\mathcal{S}$, let $\langle f, g \rangle := \mathbb{E}_{X\sim\mathcal{S}}[f(X) \cdot g(X)]$ and $\|f\| := \sqrt{\langle f, f \rangle}$. For a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, its spectral norm is denoted by $\|\mathbf{A}\|$.

We use Vinogradov notation, i.e. given $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}_+$, we write $f \ll g$ if there exist $x_0, \alpha \in \mathbb{R}_+$ such that for all $x > x_0$ we have $|f(x)| \le \alpha g(x)$. When $f : \mathbb{R} \to \mathbb{R}_+$, we write $f \asymp g$ if $f \ll g$ and $g \ll f$. For $x > 0$, we write $f = \mathrm{poly}(x)$ if there exists $k \in \mathbb{N}$ such that $f \ll x^k$. We write $f = \widetilde{O}(g)$ if $f \ll g \ln^k g$ for some $k > 0$. Similarly, $f = \widetilde{\Omega}(g)$ means that $g \ln^k g \ll f$ for some $k > 0$.

$\mathbb{Z}_p$ is the set $\{0, 1, \dots, p-1\}$, equipped with two operations, + and ×, which work as usual addition and multiplication, except that the results are reduced modulo p. $\mathbb{Z}_p^*$ denotes the set of elements in $\mathbb{Z}_p$ that are relatively prime to p. We are mainly interested in the case when p is a prime number greater than 2. In this case $\mathbb{Z}_p^* = \{1, \dots, p-1\}$. By abuse of notation, we sometimes treat elements of $\mathbb{Z}_p$ (and of $\mathbb{Z}_p^*$) as integers in $\mathbb{Z}$. Given two positive integers a and p, $a \bmod p$ is the remainder of the Euclidean division of a by p, where a is the dividend and p is the divisor.

2. The Discrete Logarithm Problem

Let $(\mathcal{G}, \circ)$ be a finite group, $a \in \mathcal{G}$ an element of order p, and $x \in \langle a \rangle$, where $\langle a \rangle := \{\underbrace{a \circ a \circ \ldots \circ a}_{k \text{ times}} : 0 \le k \le p-1\}$ is the cyclic group generated by a. The **discrete logarithm problem (DLP)** is finding the integer k, $0 \le k \le p-1$, such that

$$\underbrace{a \circ a \circ \ldots \circ a}_{k \text{ times}} = x.$$

This integer k is called the *discrete logarithm* of x to the base a, and we will denote it by $\log_a x$.

It is important to understand that there are finite groups in which the DLP is not hard (computationally). As an example, consider the additive group of integers modulo prime. For example, if we take $p = 11$, $(Z_{11}, +)$ is a finite cyclic group in which every non-zero element is primitive.[1] Here, for example, is how the element $a = 2$ generates the entire group:

k	0	1	2	3	4	5	6	7	8	9	10
$k \cdot a$	0	2	4	6	8	10	1	3	5	7	9

Table 1: Generating the group $(\mathbb{Z}_{11}, +)$ from $a = 2$.

Suppose we want to solve the DLP for the element $x = 3$, that is, we want to find an integer k such that

$$\underbrace{2 + 2 + \ldots + 2}_{k \text{ times}} \equiv 3 \bmod 11.$$

This can be done as follows. Even though the group operation is addition, we can express the relationship between x, k, and the discrete logarithm using multiplication:

$$k \cdot 2 \equiv 3 \bmod 11 \tag{1}$$

To solve the equation (1) for k, we just need to find the (multiplicative) inverse for $a = 2$:

$$k \equiv 2^{-1} \cdot 3 \bmod 11.$$

Using, for example, the extended Euclidean algorithm, we can compute $2^{-1} \equiv 6 \bmod 11$ and so the value of the discrete logarithm is

$$k \equiv 2^{-1} \cdot 3 \equiv 7 \bmod 11.$$

Table 1 confirms the correctness of the found value.

The above technique can be used for any group $(\mathbb{Z}_p, +)$ and any non-zero elements $a, x \in \mathbb{Z}_p$. Accordingly, the DLP is a computationally easy problem over $(\mathbb{Z}_p, +)$. Such groups cannot be used for cryptography. However, our subsequent analysis will show that even in them, learning just a single bit of the discrete logarithm is intractable for gradient-based methods.

One may wonder why the discrete logarithm over $(\mathbb{Z}_p, +)$ is easy, but over a general finite cyclic $(\mathcal{G}, \circ)$ of order p—which is isomorphic to $(\mathbb{Z}_p, +)$—may be hard. The reason is that an isomorphism between $(\mathbb{Z}_p, +)$ and $(\mathcal{G}, \circ)$ is established through the correspondence $k \leftrightarrow \underbrace{a \circ a \circ \ldots \circ a}_{k \text{ times}}$, where $k \in \mathbb{Z}_p$ and a is an arbitrary non-identity element of $\mathcal{G}$. But it is widely believed that this isomorphism itself is a one-way function[2] for sufficiently complex groups (like elliptic ones).

We note that the best classical methods for solving the DLP in general case—Baby-step Giant-step (Daniel, 1971) and Pollard's rho (Pollard, 1975)—require $O(\sqrt{p})$ computational steps, where p is the group order, i.e. they are exponential in the bitlength of p.

[1] An element a of a cyclic group $(\mathcal{G}, \circ)$ is called *primitive* if every element $x \in \mathcal{G}$ can be written as $x = \underbrace{a \circ a \circ \ldots \circ a}_{k \text{ times}}$ for some k, i.e. a generates the entire group.

[2] Informally, a one-way function is a function that is easy to compute on every input, but hard to invert given the image of a random input.

3. Main Result

Let $(\mathcal{G}, \circ)$ be a finite cyclic group of prime order p, and $a \in \mathcal{G} \setminus \{1\}$. Consider a function

$$h_a(x) := (-1)^{\log_a x} = \begin{cases} -1 & \text{if } \log_a x \equiv 1 \bmod 2, \\ +1 & \text{if } \log_a x \equiv 0 \bmod 2, \end{cases}, \quad x \in \mathcal{G} \setminus \{1\} \tag{2}$$

which is essentially a parity bit of the discrete logarithm $\log_a x$. Suppose we want to learn $h_a(x)$ using a gradient-based method (e.g., deep learning). For this, consider the stochastic optimization problem

$$L_a(\mathbf{w}) := \mathop{\mathbb{E}}_{X \sim \mathcal{G} \setminus \{1\}} [\ell(f_{\mathbf{w}}(X), h_a(X))] \to \min_{\mathbf{w}}, \tag{3}$$

where ℓ is a loss function, X are the random inputs (from $\mathcal{G} \setminus \{1\}$), and $f_{\mathbf{w}}$ is some model parametrized by a parameter vector $\mathbf{w}$ (e.g. a neural network of a certain architecture). We assume that $L_a(\mathbf{w})$ is differentiable with respect to $\mathbf{w}$. We are interested in studying the *variance* of the gradient of L_A when A is drawn uniformly at random from $\mathcal{G} \setminus \{1\}$. The following theorem bounds this variance term.

Theorem 1 *Suppose that $f_{\mathbf{w}}(\mathbf{x})$ is differentiable w.r.t. $\mathbf{w}$, and for some scalar $d(\mathbf{w})$, satisfies $\mathbb{E}_{X \sim \mathcal{G} \setminus \{1\}} \left[\left\| \frac{\partial}{\partial \mathbf{w}} f_{\mathbf{w}}(X) \right\|^2 \right] \leq d(\mathbf{w})^2$. Let the loss function ℓ in (3) be either the square loss $\ell(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$ or a classification loss of the form $\ell(\hat{y}, y) = s(\hat{y} \cdot y)$ for some 1-Lipschitz function s. Then*

$$\mathop{\mathbb{E}}_{A \sim \mathcal{G} \setminus \{1\}} \| \nabla L_A(\mathbf{w}) - \boldsymbol{\mu}(\mathbf{w}) \|^2 \leq \frac{c \cdot d(\mathbf{w})^2 \ln p}{\sqrt{p}}, \tag{4}$$

where $\boldsymbol{\mu}(\mathbf{w}) := \mathbb{E}_{A \sim \mathcal{G} \setminus \{1\}} \nabla L_A(\mathbf{w})$, and c is an absolute constant.

Remark 2 *Theorem 1 says that the gradient of $L_a(\mathbf{w})$ at any point $\mathbf{w}$ is extremely concentrated around a fixed point* ***independent*** *of the base a.*[3] *Using this one can show (Shamir, 2018, Theorem 10) that a gradient-based method will likely fail in returning a reasonable predictor of the discrete logarithm's parity bit unless the number of iterations is exponentially large in the bitlength n of p. This provides strong evidence that gradient-based methods cannot learn even a single bit of the discrete logarithm in* poly(n) *time. The result holds* ***regardless of which class of predictors we use*** *(e.g. arbitrarily complex neural networks) — the problem lies in using gradient-based method to train them.*

Proof Idea. Our result is an extension of the work of Shalev-Shwartz et al. (2017) which shows that the gradient is not informative when learning a class of orthogonal functions (see their Theorem 1). In their proof, they rely on the Bessel inequality, which is valid for an orthonormal sequence in the inner-product space. Unfortunately, their result cannot be applied to the parity bit of the discrete logarithm, because the functions $\{h_a(x) \mid a \in \mathcal{G} \setminus \{1\}\}$, where $h_a(x)$ is defined by (2), are not orthogonal, i.e.

[3] Using our result and Chebyshev's inequality, one can show that the gradient deviates in 2-norm from a fixed point $\boldsymbol{\mu}(\mathbf{w})$ by more than $\widetilde{\Omega}(2^{-n/6})$ with probability at most $\widetilde{O}(2^{-n/6})$, where n is the bit length of p (group order).

$\langle h_a, h_b \rangle := \mathbb{E}_{X\sim\mathcal{G}\setminus\{1\}}[h_a(X)\cdot h_b(X)] \neq 0$ for some $a, b \in \mathcal{G}\setminus\{1\}$, $a \neq b$. However, we can show that, on average over a and b, the inner product $\langle h_a, h_b\rangle$ is small (Lemma 12). More precisely, it satisfies

$$\mathop{\mathbb{E}}_{\substack{A,B\sim\mathcal{G}\setminus\{1\}\\ A\neq B}}\left[\langle h_A, h_B\rangle^2\right] \ll \frac{\ln^2 p}{p}. \tag{5}$$

Further, using the Boas-Bellman inequality (Lemma 10) instead of the Bessel inequality in the proof of Shalev-Shwartz et al. (2017), we can get the bound (4).

We note that in order to prove (5), we have established some intermediate results that may be of independent interest. Namely, we have shown that for the matrix

$$\boldsymbol{\Phi}' := \{(-1)^{jk \bmod p}\}_{j,k\in\mathbb{Z}_p^*},$$

the spectral norm $\|\boldsymbol{\Phi}'\| \ll \sqrt{p}\ln p$ (Lemma 9). Using this fact, we proved that for $Y \sim \mathbb{Z}_p^*$ a random variable

$$f(Y) = \mathop{\mathbb{E}}_{X\sim\mathbb{Z}_p^*}\left[(-1)^X \cdot (-1)^{YX \bmod p}\right], \tag{6}$$

is concentrated around its mean $\mathbb{E}[f(Y)] = 0$ with a variance $\mathrm{Var}[f(Y)] \ll \frac{\ln^2 p}{p}$. And this, in turn, implies (5), as is shown in Lemma 12.

The proof of Theorem 1 is given in Section 5.

4. Empirical Verification

Our code is available at https://github.com/armanbolatov/hardness_of_learning.

Concentration of the Gradient. As mentioned earlier, Theorem 1 is true for any finite cyclic group of prime order, including the additive group $(\mathbb{Z}_p, +)$. Let us verify empirically the statement of the theorem for this group. Let $f_{\mathbf{w}}(x)$ be a neural network[4] that we may want to train to learn the mapping[5] $x \mapsto (-1)^{\log_a x}$, $x \in \mathbb{Z}_p^*$, where $\mathbf{w} \in \mathbb{R}^d$ are all parameters of the neural network. Let $\nabla L_a(\mathbf{w})$ be the gradient of the binary cross-entropy loss function at $\mathbf{w}$. We sample $\mathbf{w}_1, \ldots, \mathbf{w}_{20}$ from $\mathbb{R}^d$ using the default PyTorch initializer[6], and for each $\mathbf{w}_i$, we compute

$$v(\mathbf{w}_i) = \mathop{\mathbb{E}}_{A\sim\mathbb{Z}_p^*}\left\|\nabla L_A(\mathbf{w}_i) - \mathop{\mathbb{E}}_{A'\sim\mathbb{Z}_p^*}\nabla L_{A'}(\mathbf{w}_i)\right\|^2, \tag{7}$$

$$g(\mathbf{w}_i) = \mathop{\mathbb{E}}_{X\sim\mathbb{Z}_p^*}\left\|\frac{\partial}{\partial\mathbf{w}} f_{\mathbf{w}}(X)\right\|^2. \tag{8}$$

According to Theorem 1, the values $\frac{v(\mathbf{w}_i)}{g(\mathbf{w}_i)}$ should be of order $\widetilde{O}\left(\frac{1}{\sqrt{p}}\right)$. Thus we plot $\mathbb{E}_{i\sim\{1,\ldots,20\}}\left[\frac{v(\mathbf{w}_i)}{g(\mathbf{w}_i)}\cdot\sqrt{p}\right]$ against p in Figure 1. As we can see, this expression is bounded

[4] 3-layer dense neural network with 1000 neurons on each hidden layer, sigmoid activation, binary cross-entropy loss.

[5] We remind the reader that for prime p, in $(\mathbb{Z}_p, +)$ we have $\log_a x \equiv a^{-1}x \bmod p$, where a^{-1} is the multiplicative inverse of a (see Section 5.2).

[6] For each dense layer of shape $d_{\text{in}} \times d_{\text{out}}$, PyTorch initializes its parameters uniformly at random from the interval $\left[-1/\sqrt{d_{\text{in}}}, 1/\sqrt{d_{\text{in}}}\right]$, where d_{in} is the size of the input, and d_{out} is the size of the output.

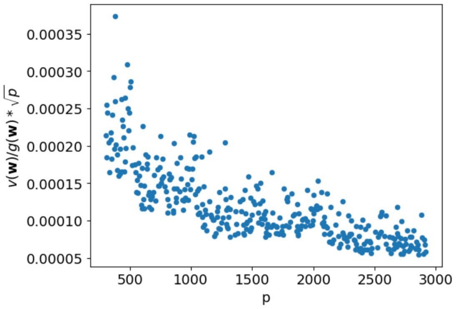

Figure 1: Verifying the statement of Theorem 1. For prime numbers p in $[300, 3000]$, we plot the left-hand side of (4) divided by the average squared norm of the neural network's gradient (8) and multiplied by $\sqrt{p}$. The resulting curve is of order $\widetilde{O}(1)$. Moreover, it even decreases.

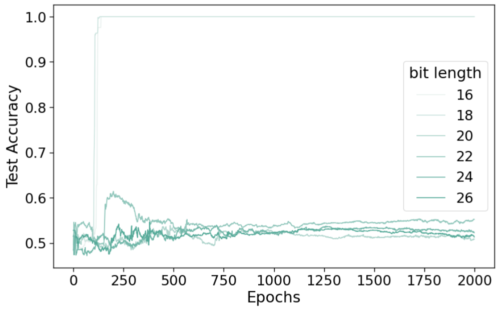

Figure 2: Learning the parity bit of the discrete logarithm in $(\mathbb{Z}_p, +)$ with a 3-layer width-1000 dense network. Darker shades correspond to longer bitlengths. For each bitlength n, the group order p is chosen randomly from the prime numbers in the interval $[2^{n-1}, 2^n - 1]$.

as p grows, which confirms the statement of the theorem. In fact, it is not only bounded but actually decreases, suggesting that our upperbound (4) can be improved.

Failure of Gradient-Based Learning. According to Remark 2, any gradient-based method most likely will fail to learn the parity bit of a discrete logarithm. To test this claim, we generated a labeled sample

$$(x_1, (-1)^{\log_a x_1}), \ldots, (x_m, (-1)^{\log_a x_m})$$

where $x_1, \ldots, x_m$, and a are taken randomly from $\mathbb{Z}_p^*$. Using this sample, we trained a dense 3-layer neural network with 1000 neurons in each hidden layer. We used Adam with a learning rate of 0.001 (default), $m = 5000$, a 70/30 split between training and test sets, batch size 100, and we trained for 2000 epochs. The results for different bitlengths n are shown in Figure 2. The group order p for each bitlength n was taken randomly from the prime numbers in the interval $[2^{n-1}, 2^n - 1]$. We can see that as the bitlength increases, the chances of successful learning decrease, as predicted by our theory.

5. Proofs

5.1. Some Statistical Properties of $\mathbb{Z}_p^*$

The goal of this subsection is to study the distribution of the random variable $f(Y)$, defined by (6), where Y is sampled uniformly at random from $\mathbb{Z}_p^*$. Our key result is the following theorem that will be crucial for our further analysis of the discrete logarithm's parity bit in Section 5.2.

Theorem 3 *Let $f(Y)$ be a random variable defined by (6), where $Y \sim \mathbb{Z}_p^*$. Then*

$$\mathbb{E}[f(Y)] = 0, \qquad \mathrm{Var}[f(Y)] \le \frac{c \ln^2 p}{p},$$

where c is some universal constant.

Let us denote $\phi(X,Y) = (-1)^X \cdot (-1)^{YX \bmod p}$ and assume that X is also sampled uniformly from $\mathbb{Z}_p^*$. Then, $f(Y) = \mathbb{E}_X[\phi(X,Y)]$. Let us first study the distribution of the random variable $\phi(X,Y)$. Our significant finding is the following lemma.

Lemma 4 $\mathbb{E}_Y[\phi(x,Y)] = 0$ *for any $x \in \mathbb{Z}_p^*$.*

Proof Direct computation gives

$$\mathbb{E}_Y[\phi(x,Y)] = (-1)^x \frac{1}{p-1} \sum_{y \in \mathbb{Z}_p^*} (-1)^{yx \bmod p} = (-1)^x \frac{1}{p-1} \sum_{y' \in \mathbb{Z}_p^*} (-1)^{y'} = 0,$$

due to the fact that $\{yx \bmod p \mid y \in \mathbb{Z}_p^*\} = \mathbb{Z}_p^*$ and $|\mathbb{Z}_p^*|$ is even. ■

From Lemma 4 we conclude that

$$\mathbb{E}_Y[f(Y)] = \mathbb{E}_{X,Y}[\phi(X,Y)] = 0. \tag{9}$$

Thus, $f(Y)$ is distributed around its mean 0 and the first statement of Theorem 3 is proved. Let us now study its variance. The following lemma shows that the spectral norm of the matrix $\mathbf{\Phi} = [\phi(x,y)]_{(x,y)\in(\mathbb{Z}_p^*)^2} \in \mathbb{R}^{(p-1)\times(p-1)}$ bounds the variance.

Lemma 5 $\mathrm{Var}[f(Y)] \le \frac{\sigma_1(\mathbf{\Phi})^2}{(p-1)^2}$.

Proof Formula (9) implies that the variance is equal to the second moment, therefore

$$\begin{aligned}
\mathrm{Var}[f(Y)] = \mathbb{E}[f(Y)^2] &= \frac{1}{p-1} \sum_{y \in \mathbb{Z}_p^*} \frac{1}{(p-1)^2} \left(\sum_{x \in \mathbb{Z}_p^*} \phi(x,y) \right)^2 \\
&= \frac{1}{(p-1)^3} \sum_{(y,x,x') \in (\mathbb{Z}_p^*)^3} \phi(x,y)\phi(x',y) = \frac{1}{(p-1)^3} \sum_{(x,x') \in (\mathbb{Z}_p^*)^2} (\mathbf{\Phi}\mathbf{\Phi}^\top)_{x,x'} \\
&= \frac{1}{(p-1)^2} \sum_{(x,x') \in (\mathbb{Z}_p^*)^2} (\mathbf{\Phi}\mathbf{\Phi}^\top)_{x,x'} \xi_x \xi_{x'},
\end{aligned}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{p-1}$ satisfies $\xi_i = \frac{1}{\sqrt{p-1}}, i \in \mathbb{Z}_p^*$. The latter quadratic form is bounded by the largest eigenvalue of the symmetric matrix $\mathbf{\Phi}\mathbf{\Phi}^\top$, i.e.

$$\mathrm{Var}[f(Y)] = \mathbb{E}[f(Y)^2] \le \frac{1}{(p-1)^2} \lambda_1(\mathbf{\Phi}\mathbf{\Phi}^\top) = \frac{1}{(p-1)^2} \sigma_1(\mathbf{\Phi})^2.$$

■

Our next goal will be to study the spectral norm of $\mathbf{\Phi}$. The following lemma simplifies our task.

Lemma 6 *Let $\mathbf{\Phi}' = [(-1)^{yx \bmod p}]_{(x,y)\in(\mathbb{Z}_p^*)^2}$. Then we have $\sigma_\ell(\mathbf{\Phi}) = \sigma_\ell(\mathbf{\Phi}')$, $\ell \in \mathbb{Z}_p^*$.*

Proof Let $\mathbb{1} \in \mathbb{R}^{p-1}$ be a vector with all components equal to 1 and $\boldsymbol{\xi} \in \mathbb{R}^{p-1}$ be such that $\xi_x = (-1)^x$. Since $\mathbf{\Phi}_{x,y} = (-1)^x(-1)^{yx \bmod p}$ we have $\mathbf{\Phi} = (\mathbb{1}\boldsymbol{\xi}^\top) \odot \mathbf{\Phi}'$ where $\odot$ is the Hadamard product. Let us consider the singular value decomposition (SVD) of $\mathbf{\Phi}'$, i.e. $\mathbf{\Phi}' = \sum_{\ell=1}^{p-1} \sigma_\ell(\mathbf{\Phi}')\mathbf{u}_\ell\mathbf{v}_\ell^\top$. Then, we have

$$\mathbf{\Phi} = (\mathbb{1}\boldsymbol{\xi}^\top) \odot \mathbf{\Phi}' = \sum_{\ell=1}^{p-1} \sigma_i(\mathbf{\Phi}')(\mathbb{1}\boldsymbol{\xi}^\top) \odot (\mathbf{u}_\ell\mathbf{v}_\ell^\top) = \sum_{\ell=1}^{p-1} \sigma_\ell(\mathbf{\Phi}')(\mathbb{1} \odot \mathbf{u}_\ell)(\boldsymbol{\xi} \odot \mathbf{v}_\ell)^\top.$$

Since $\{\mathbb{1} \odot \mathbf{u}_\ell \mid \ell \in [p-1]\}$ and $\{\boldsymbol{\xi} \odot \mathbf{v}_\ell \mid \ell \in [p-1]\}$ are both orthonormal systems of vectors, the latter expression is an SVD of $\mathbf{\Phi}$. Therefore, $\sigma_\ell(\mathbf{\Phi}) = \sigma_\ell(\mathbf{\Phi}')$. ■

A final result concerning the largest singular value of $\mathbf{\Phi}'$ requires some additional lemmas. Let us denote the vector $[(-1)^x]_{x\in\mathbb{Z}_p} \in \mathbb{R}^p$ by $\mathbf{a}$. Let $\omega = e^{\frac{2\pi i}{p}}$ be a primitive pth root of unity. Other primitive roots of unity are $\omega_2, \cdots, \omega_{p-1}$ where $\omega_\ell = \omega^\ell, \ell \geq 0$. Let $\mathbf{U}_\ell = [\omega_\ell^{jk}]_{j,k\in\mathbb{Z}_p}$, then the matrix $\frac{1}{\sqrt{p}}\mathbf{U}_\ell$ is unitary for $\ell \in \mathbb{Z}_p^*$. In fact, $\frac{1}{\sqrt{p}}\mathbf{U}_1$ is a discrete Fourier transform (DFT) matrix. From unitarity, we obtain $\|\mathbf{U}_\ell\| = \sqrt{p}$. Let us denote $\mathbf{U}_1 = \big[\mathbf{b}_0, \cdots, \mathbf{b}_{p-1}\big]$.

Lemma 7 *The vector $\mathbf{a}$ can be decomposed as $\mathbf{a} = \frac{2}{p}\sum_{\ell=0}^{p-1} \frac{1}{1+\omega^{-\ell}}\mathbf{b}_\ell$.*

Proof From unitarity, we conclude that $\left\{\mathbf{e}_\ell = \frac{1}{\sqrt{p}}\mathbf{b}_\ell\right\}_{\ell=0}^{p-1}$ is an orthonormal basis in $\mathbb{C}^p$. Therefore, $\mathbf{a} = \sum_{\ell=0}^{p-1}(\mathbf{e}_\ell^\dagger\mathbf{a})\mathbf{e}_\ell$. After computation

$$(\mathbf{e}_\ell^\dagger\mathbf{a}) = \frac{1}{\sqrt{p}}\sum_{x=0}^{p-1}\omega^{-x\ell}(-1)^x = \frac{1}{\sqrt{p}}\sum_{x=0}^{p-1}(-\omega^{-\ell})^x = \frac{1}{\sqrt{p}} \cdot \frac{1-(-\omega^{-\ell})^p}{1+\omega^{-\ell}} = \frac{2}{\sqrt{p}} \cdot \frac{1}{1+\omega^{-\ell}},$$

we conclude that

$$\mathbf{a} = \sum_{\ell=0}^{p-1} \frac{2}{\sqrt{p}} \cdot \frac{1}{1+\omega^{-\ell}}\mathbf{e}_\ell = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}}\mathbf{b}_\ell.$$

■

Corollary 8 *The matrix $\mathbf{\Phi}'$ can be represented as $\mathbf{\Phi}' = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}}\mathbf{\Omega}_\ell$, where $\mathbf{\Omega}_\ell$ is a submatrix of $\mathbf{U}_\ell$ obtained after deletion of the first row and the first column.*

Proof Unfolded component-wise, the corollary is equivalent to

$$(-1)^{jk \bmod p} = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}}\omega_\ell^{jk},$$

for any $j, k \in \mathbb{Z}_p^*$. After setting $s = jk \bmod p$ and using Lemma 7 we conclude

$$\begin{aligned}(-1)^{jk \bmod p} = (\mathbf{a})_s = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}} (\mathbf{b}_\ell)_s = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}} \omega^{\ell s} \\ = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}} \omega^{\ell (jk \bmod p)} = \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}} \omega_\ell^{jk},\end{aligned}$$

which concludes the proof. ■

Now it remains to bound $\sigma_1(\mathbf{\Phi'})$.

Lemma 9 *$\sigma_1(\mathbf{\Phi'}) \leq c_1 p^{1/2} \ln p$ where c_1 is some universal constant.*

Proof Using Corollary 8 we conlcude

$$\|\mathbf{\Phi'}\| = \left\| \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{1+\omega^{-\ell}} \mathbf{\Omega}_\ell \right\| \leq \sum_{\ell=0}^{p-1} \frac{2}{p} \cdot \frac{1}{|1+\omega^{-\ell}|} \|\mathbf{\Omega}_\ell\|.$$

The spectral norm of every submatrix is not greater than that of the matrix, therefore, $\|\mathbf{\Omega}_\ell\| \leq \|\mathbf{U}_\ell\| = \sqrt{p}$. Thus,

$$\|\mathbf{\Phi'}\| \leq \sqrt{p} \sum_{\ell=0}^{p-1} \frac{2}{p} \frac{1}{|1+\omega^{-\ell}|}.$$

Now it remains to bound the sum $\sum_{\ell=0}^{p-1} \frac{2}{p} \frac{1}{|1+\omega^{-\ell}|} = \sum_{\ell=0}^{p-1} \frac{2}{p} \frac{1}{|1+\omega^{\ell}|}$. Let $\theta = \frac{2\pi\ell}{p}$. Note that $\theta \in [0, 2\pi)$ and

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

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

$$\begin{aligned}\sum_{\ell \in \left[\frac{1}{4}p, \frac{3}{4}p\right] \cap \mathbb{Z}_p} \frac{2}{p} \cdot \frac{1}{|1+\omega^\ell|} \leq \sum_{\ell \in \left[\frac{1}{4}p, \frac{3}{4}p\right] \cap \mathbb{Z}_p} \frac{2}{p} \cdot \frac{1}{\left|\frac{\pi\ell}{p} - \frac{\pi}{2}\right|} = \frac{4}{\pi} \sum_{\ell \in \left[\frac{1}{4}p, \frac{3}{4}p\right] \cap \mathbb{Z}_p} \frac{1}{|2\ell - p|} \\ \leq \frac{8}{\pi} \sum_{j=1}^{\lceil p/2 \rceil} \frac{1}{j} \ll \ln p.\end{aligned}$$

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

$$\sum_{\ell \in \mathbb{Z}_p \colon \psi \in \left[-\frac{\pi}{2} - \frac{\pi}{4}\right] \cup \left[\frac{\pi}{4}, \frac{\pi}{2}\right]} \frac{2}{p} \cdot \frac{1}{|1+\omega^\ell|} \asymp 1.$$

Thus, the total sum satisfies

$$\sum_{\ell \in \mathbb{Z}_p} \frac{2}{p} \frac{1}{|1+\omega^{\ell}|} \ll \ln p.$$

■

Proof of Theorem 3. Using Lemma 5 we have $\mathrm{Var}[f(Y)] \leq \frac{\sigma_1(\mathbf{\Phi})^2}{(p-1)^2}$. Lemma 6 additionally gives us $\mathrm{Var}[f(Y)] \leq \frac{\sigma_1(\mathbf{\Phi}')^2}{(p-1)^2}$. Finally, using Lemma 9 we conlude that $\mathrm{Var}[f(Y)] \ll \frac{\ln^2 p}{p}$. ■

5.2. Near Orthogonality of Discrete Logarithm's Parity Bits

As mentioned in Section 3, the main tool for adapting the proof of Shalev-Shwartz et al. (2017) to our needs is the Boas-Bellman inequality, which we present below.

Lemma 10 (Boas-Bellman inequality) *Let $h_1, \ldots, h_m, g$ be elements of an inner product space. Then*

$$\sum_{i=1}^{m} \langle h_i, g \rangle^2 \leq \|g\|^2 \left(\max_i \|h_i\|^2 + \sqrt{\sum_{i \neq j} \langle h_i, h_j \rangle^2} \right).$$

Proof Can be found in the works of Boas (1941) and Bellman (1944). ■

As we can see, this inequality turns into Bessel's inequality for an orthonormal sequence $\{h_i\}$. In our case, the functions $h_a(x)$ given by (2) are not pairwise orthogonal. However, it can be shown that they are approximately pairwise orthogonal. This is what we will do in this subsection. First, we derive the distribution of $\log_A B$ when A and B are sampled uniformly at random from $\mathcal{G} \setminus \{1\}$.

Lemma 11 *Let $(\mathcal{G}, \circ)$ be a finite cyclic group of prime order $p > 2$. Let $A, B \sim \mathcal{G} \setminus \{1\}$. Then the distribution of $\log_B A$ is uniform over $\{1, \ldots, p-1\}$.*

Proof For any $y \in \{1, \ldots, p-1\}$, we have

$$\begin{aligned}
&\Pr_{A,B \sim \mathcal{G} \setminus \{1\}} [\log_B A = y] = \mathop{\mathbb{E}}_{A,B \sim \mathcal{G} \setminus \{1\}} [\mathbb{I}(\log_B A = y)] \\
&= \frac{1}{(p-1)^2} \sum_{a,b \in \mathcal{G} \setminus \{1\}} \mathbb{I}[\log_b a = y] = \frac{1}{(p-1)^2} \sum_{b \in \mathcal{G} \setminus \{1\}} \sum_{a \in \mathcal{G} \setminus \{1\}} \underbrace{\mathbb{I}[\log_b a = y]}_{=1 \text{ iff } a = b^y} \\
&= \frac{1}{(p-1)^2} \sum_{b \in \mathcal{G} \setminus \{1\}} 1 = \frac{1}{p-1}.
\end{aligned}$$

■

Now we are ready to prove the advertised bound (5).

Lemma 12 *Let $(\mathcal{G},\circ)$ be a finite cyclic group of prime order $p>2$. Let $a\in\mathcal{G}\setminus\{1\}$. Consider a function $h_a(x)$ defined in (2). Then*

$$\sum_{\substack{a,b\in\mathcal{G}\setminus\{1\}\\ a\neq b}}\mathbb{E}[h_a(X)\cdot h_b(X)]^2\le c_2p\ln^2 p$$

for some universal constant c_2.

Proof

$$\begin{aligned}
&\frac{1}{(p-1)^2}\sum_{a\in\mathcal{G}\setminus\{1\}}\sum_{b\in\mathcal{G}\setminus\{1\}}\mathbb{E}[h_a(X)\cdot h_b(X)]^2\\
&=\mathop{\mathbb{E}}_{A,B\sim\mathcal{G}\setminus\{1\}}\left[\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[h_A(X)\cdot h_B(X)]^2\right]\\
&=\mathop{\mathbb{E}}_{A,B\sim\mathcal{G}\setminus\{1\}}\left[\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}\left[(-1)^{\log_A X}\cdot(-1)^{\log_B X}\right]^2\right]\\
&=\mathop{\mathbb{E}}_{A,B\sim\mathcal{G}\setminus\{1\}}\left[\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}\left[(-1)^{\log_A X}\cdot(-1)^{\log_B A\cdot\log_A X \bmod p}\right]^2\right]\\
&=\mathop{\mathbb{E}}_{A,B\sim\mathcal{G}\setminus\{1\}}\left[\mathop{\mathbb{E}}_{X\sim\mathbb{Z}_p^*}\left[(-1)^{X}\cdot(-1)^{\log_B A\cdot X \bmod p}\right]^2\right]\\
&\overset{\text{Lem. 11}}{=}\mathop{\mathbb{E}}_{Y\sim\mathbb{Z}_p^*}\left[\mathop{\mathbb{E}}_{X\sim\mathbb{Z}_p^*}\left[(-1)^{X}\cdot(-1)^{Y\cdot X \bmod p}\right]^2\right]=\mathop{\mathrm{Var}}_{Y\sim\mathbb{Z}_p^*}[f(Y)]\overset{\text{Thm. 3}}{\ll}\frac{\ln^2 p}{p}
\end{aligned}$$

Therefore, $\sum_{a\in\mathcal{G}\setminus\{1\}}\sum_{b\in\mathcal{G}\setminus\{1\}}\mathbb{E}[h_a(X)\cdot h_b(X)]^2\ll p\ln^2 p$ and

$$\sum_{a\neq b}\mathbb{E}[h_a(X)\cdot h_b(X)]^2\le c_3p\ln^2 p-(p-1)\ll p\ln^2 p.$$

■

5.3. Proof of Theorem 1

Proof We prove the result for the squared loss $\ell(\hat{y},y)=\frac{1}{2}(\hat{y}-y)^2$. The classification loss is handled analogously. Define the vector-valued function

$$\mathbf{g}(x)=\frac{\partial}{\partial\mathbf{w}}f_{\mathbf{w}}(x),$$

and let $\mathbf{g}(x)=(g_1(x),g_2(x),\dots,g_d(x))$ for real-valued functions $g_1,\dots,g_d$. Then we have

$$\begin{aligned}
\boldsymbol{\mu}(\mathbf{w}):=\mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}}\nabla L_A(\mathbf{w})&=\mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}}\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[(f_{\mathbf{w}}(X)-h_A(X))\mathbf{g}(X)]\\
&=\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[f_{\mathbf{w}}(X)\mathbf{g}(X)]-\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}\left\{\mathbf{g}(X)\mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}}[(-1)^{\log_A X}]\right\}\\
&=\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[f_{\mathbf{w}}(X)\mathbf{g}(X)]-\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}\mathbf{g}(X)\cdot\underbrace{\mathop{\mathbb{E}}_{Y\sim\mathbb{Z}_p^*}[(-1)^Y]}_{0}=\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[f_{\mathbf{w}}(X)\mathbf{g}(X)]
\end{aligned}$$

Thus,

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \|\nabla L_A(\mathbf{w}) - \boldsymbol{\mu}(\mathbf{w})\|^2 \\
&= \mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \left\| \mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[(f_{\mathbf{w}}(X) - h_A(X))\mathbf{g}(X)] - \mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[f_{\mathbf{w}}(X)\mathbf{g}(X)] \right\|^2 \\
&= \mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \left\| \mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[h_A(X)\mathbf{g}(X)] \right\|^2 = \mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \sum_{j=1}^{d} \left(\mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[h_A(X)g_j(X)] \right)^2 \\
&= \mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \sum_{j=1}^{d} \langle h_A, g_j\rangle^2. \qquad (10)
\end{aligned}
$$

From Lemmas 10 and 12, we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \langle h_A, g\rangle^2 = \frac{1}{p-1} \sum_{a\in\mathcal{G}\setminus\{1\}} \langle h_a, g\rangle^2 \\
&\le \frac{\|g\|^2}{p-1} \left(\max_{a\in\mathcal{G}\setminus\{1\}} \|h_a\|^2 + \sqrt{\sum_{\substack{a,b\in\mathcal{G}\setminus\{1\}\\ a\neq b}} \langle h_a, h_b\rangle^2} \right) \le \|g\|^2 \left(\frac{1}{p-1} + \frac{\sqrt{cp}\ln p}{p-1} \right) \qquad (11)
\end{aligned}
$$

From (10) and (11), we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{A\sim\mathcal{G}\setminus\{1\}} \left\| \nabla L_A(\mathbf{w}) - \mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}}[f_{\mathbf{w}}(X)\mathbf{g}(X)] \right\|^2 \le \left(\frac{1}{p-1} + \frac{\sqrt{cp}\ln p}{p-1} \right) \sum_{j=1}^{d} \|g_j\|^2 \\
&= \left(\frac{1}{p-1} + \frac{\sqrt{cp}\ln p}{p-1} \right) \mathop{\mathbb{E}}_{X\sim\mathcal{G}\setminus\{1\}} \|\mathbf{g}(X)\|^2 \le \left(\frac{1}{p-1} + \frac{\sqrt{cp}\ln p}{p-1} \right) d(\mathbf{w})^2 \\
&\ll \frac{\ln p}{\sqrt{p}} d(\mathbf{w})^2.
\end{aligned}
$$

The proof for the classification loss $\ell(\hat{y}, y) = s(\hat{y} \cdot y)$ can be reduced to the proof for the squared loss as it is done in Theorem 1 of Shalev-Shwartz et al. (2017). ■

6. Additional Experiments

Here we present the results of experiments that extend the scope of the paper. Namely, we are empirically investigating the learnability of the discrete logarithm *itself* and of *all* its bits, not just one bit.

Low correlation of discrete logarithms. We computed the mean squared covariance

$$
\mathop{\mathbb{E}}_{A,B\sim\mathbb{Z}_p^*} \left(\mathop{\mathrm{Cov}}_{X\sim\mathbb{Z}_p^*} [\log_A X, \log_B X] \right)^2 \qquad (12)
$$

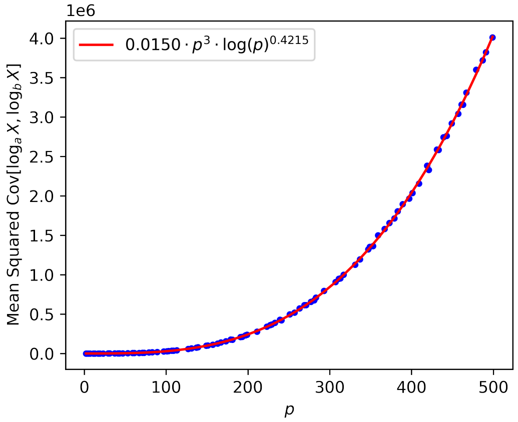

Figure 3: Mean squared covariance between two logarithms, $\log_a X$ and $\log_b X$, when X is a random variable uniformly distributed on $\mathbb{Z}_p^*$.

for prime numbers in the interval $[3, 500]$. The results are shown in Figure 3. As we can see, the expression (12) fits the curve $p \mapsto 0.015 \cdot p^3 \cdot (\ln p)^{0.42}$ well. This suggests that $\mathrm{Cov}_{X\sim\mathbb{Z}_p^*}[\log_a X, \log_b X] = \widetilde{O}(p^{3/2})$ on average over $a, b \in \mathbb{Z}_p^*$. Since the variance of the discrete logarithm is

$$\operatorname*{Var}_{X\sim\mathbb{Z}_p^*}[\log_a X] = \frac{1}{p-1}\sum_{k=1}^{p-1} k^2 - \left(\frac{1}{p-1}\sum_{k=1}^{p-1} k\right)^2 = \frac{p^2}{12} - \frac{p}{6} = O(p^2)$$

we can conjecture that the average correlation is

$$\operatorname*{Corr}_{X\sim\mathbb{Z}_p^*}[\log_a X, \log_b X] = \widetilde{O}\left(\frac{1}{\sqrt{p}}\right). \tag{13}$$

Thus, using this estimate in the Boas-Bellman inequality for the class of "standardized" discrete logarithms $\{f_a(x) \mid a \in \mathbb{Z}_p^*\}$, where

$$f_a(x) = \frac{\log_a x - \frac{p}{2}}{\sqrt{\frac{p^2}{12} - \frac{p}{6}}}, \qquad x \in \mathbb{Z}_p^*,$$

we can show that this class is also hard to learn by gradient-based methods. The only thing missing is a rigorous proof of the bound (13). We leave it to our future work.

Failure to learn all bits of the discrete logarithm. Here we follow the experimental setup from Section 4 with the difference that the output of the neural network is not only the parity bit, but all the bits of the discrete logarithm. As a loss function, we use the sum of the cross-entropies for each bit. The results for two different bit lengths are shown in Figure 4. As in the case of one bit, we see that for a longer bit length, the gradient method is not able to learn all the bits of the discrete logarithm. Note that in both cases, the more significant bits are learned better than the less significant ones. We leave the study of this phenomenon to our future work.

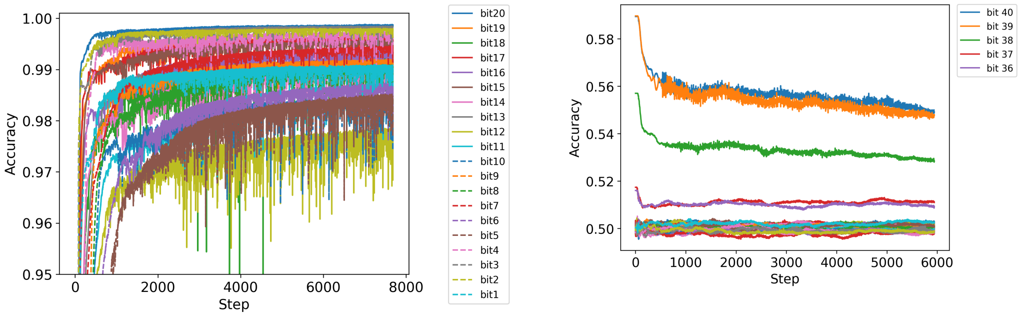

Figure 4: Test Accuracies when learning *all* bits of the discrete logarithm in $(\mathbb{Z}_p, +)$ with a single neural network. Bitlengths of p: 20 (left) and 40 (right).

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

Richard Bellman. Almost orthogonal series. *Bulletin of the American Mathematical Society*, 50(8):517–519, 1944.

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. URL https://doi.org/10.1145/195058.195147.

RP Boas. A general moment problem. *American Journal of Mathematics*, 63(2):361–370, 1941.

Shanks Daniel. Class number, a theory of factorization, and genera. In *Proc. Sympos. Pure Math.*, volume 20, pages 415–440, 1971.

Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976. doi: 10.1109/TIT.1976.1055638. URL https://doi.org/10.1109/TIT.1976.1055638.

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. URL https://doi.org/10.1137/1.9781611974782.82.

Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory*, 31(4):469–472, 1985. doi: 10.1109/TIT.1985.1057074. URL https://doi.org/10.1109/TIT.1985.1057074.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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. URL https://doi.org/10.1145/167088.167200.

Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, 2021.

Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985. doi: 10.1007/3-540-39799-X_31. URL https://doi.org/10.1007/3-540-39799-X_31.

OpenAI. Introducing ChatGPT. https://openai.com/blog/chatgpt, 2022. Accessed: 2023-05-30.

John M. Pollard. A monte carlo method for factorization. *BIT Numerical Mathematics*, 15: 331–334, 1975.

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.

Ohad Shamir. Distribution-specific hardness of learning neural networks. *J. Mach. Learn. Res.*, 19:32:1–32:29, 2018.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.