Learning Versus Pseudorandom Generators in Constant Parallel Time

Shuichi Hirahara ✉
National Institute of Informatics, Tokyo, Japan

Mikito Nanashima ✉
Tokyo Institute of Technology, Japan

Abstract

A polynomial-stretch pseudorandom generator (PPRG) in NC^0 (i.e., constant parallel time) is one of the most important cryptographic primitives, especially for constructing highly efficient cryptography and indistinguishability obfuscation. The celebrated work (Applebaum, Ishai, and Kushilevitz, SIAM Journal on Computing, 2006) on randomized encodings yields the characterization of sublinear-stretch pseudorandom generators in NC^0 by the existence of logspace-computable one-way functions, but characterizing PPRGs in NC^0 seems out of reach at present. Therefore, it is natural to ask which sort of hardness notion is *essential* for constructing PPRGs in NC^0. Particularly, to the best of our knowledge, all the previously known candidates for PPRGs in NC^0 follow only one framework based on Goldreich's one-way function.

In this paper, we present a new learning-theoretic characterization for PPRGs in NC^0 and related classes. Specifically, we consider the average-case hardness of learning for well-studied classes in parameterized settings, where the number of samples is restricted to fixed-parameter tractable (FPT), and show that the following are equivalent:

- The existence of (a collection of) PPRGs in NC^0.
- The average-case hardness of learning sparse $\mathbb{F}_2$-polynomials on a sparse example distribution and an NC^0-samplable target distribution (i.e., a distribution on target functions).
- The average-case hardness of learning Fourier-sparse functions on a sparse example distribution and an NC^0-samplable target distribution.
- The average-case hardness of learning constant-depth parity decision trees on a sparse example distribution and an NC^0-samplable target distribution.

Furthermore, we characterize a (single) PPRG in $\oplus\text{-}\mathsf{NC}^0$ by the average-case hardness of learning constant-degree $\mathbb{F}_2$-polynomials on a *uniform example distribution* with FPT samples. Based on our results, we propose new candidates for PPRGs in NC^0 and related classes under a hardness assumption on a natural learning problem. An important property of PPRGs in NC^0 constructed in our framework is that the output bits are computed by various predicates; thus, it seems to resist an attack that depends on a specific property of one fixed predicate.

Conceptually, the main contribution of this study is to formalize a theory of FPT dualization of concept classes, which yields a meta-theorem for the first result. For the second result on PPRGs in $\oplus\text{-}\mathsf{NC}^0$, we use a different technique of pseudorandom $\mathbb{F}_2$-polynomials.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Theory of computation → Boolean function learning

Keywords and phrases Parallel cryptography, polynomial-stretch pseudorandom generators in NC^0, PAC learning, average-case complexity, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.ITCS.2023.70

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2022/164/`

Funding This work was done when the second author was supported by JST, ACT-X Grant Number JPMJAX190M.
Shuichi Hirahara: JST, PRESTO Grant Number JPMJPR2024.

Acknowledgements We thank the anonymous ITCS reviewers for providing helpful comments and suggestions.

1 Introduction

A dichotomy between learning and cryptography is one of the central topics in theoretical computer science. An implication from cryptography to hardness of learning has already been studied in the pioneering work by Valiant [60], who observed that the existence of a secure cryptographic primitive implies the hardness of learning polynomial-size circuits ($\mathsf{P/poly}$). Many follow-up studies further showed the hardness of learning more restricted classes such as AC^0 under several cryptographic or deeply related assumptions [39, 40, 3, 11, 23, 24, 18, 59, 25]. The opposite implication from hardness of learning to cryptography is relatively less understood and first studied by Impagliazzo and Levin [33] and Blum, Furst, Kearns, and Lipton [14]. Particularly, Blum, Furst, Kearns, and Lipton [14] formulated the average-case hardness of PAC learning and presented constructions of several cryptographic primitives based on the average-case hardness of learning. These early studies characterized a central cryptographic primitive called a one-way function (OWF) by the average-case hardness of learning $\mathsf{P/poly}$. The dichotomy between learning and cryptography has been further studied over decades in various settings [47, 50, 55, 44, 45].

In general, the complexity for computing cryptographic primitives is deeply related to the complexity of a concept class for learning (i.e., a class of target functions learners try to learn). This observation leads us to study the dichotomy between learning and cryptography in low complexity classes. One motivation of this is highly efficient cryptography based on the hardness assumption of learning simple classes, as mentioned by Blum, Furst, Kearns, and Lipton [14]. This direction is successful in certain fields; e.g., several candidates for a cryptographic primitive called a weak pseudorandom function were proposed in low complexity based on the hardness of learning problems for which no efficient algorithm is known at present [1, 17]. Another motivation is identifying the capability of efficient learning based on well-established arguments in cryptography. This direction has also been demonstrated for decades in studies on cryptographic hardness of learning (e.g.,[39, 40, 11, 25]).

In this work, we study a dichotomy between learning and polynomial-stretch pseudorandom generators (PPRGs) computable in constant-depth circuits (i.e., NC^0), where a PPRG is a fundamental cryptographic primitive stretching a given n-bit random seed into an $n^{1+\Theta(1)}$-bit pseudorandom string that is indistinguishable from a truly random string by efficient adversaries. This research question is strongly motivated by both sides of constructing highly efficient cryptography and identifying the capability of efficient learning. Below, we explain further backgrounds.

From the perspective of cryptography. A PPRG in NC^0 is one of the most studied primitives in parallel cryptography (cf. [22, 9]) because of its remarkable applications, such as highly efficient cryptography [34] and a recent breakthrough on indistinguishability obfuscation ($i\mathcal{O}$) based on well-founded assumptions [37, 38]. Despite its importance, to the best of our knowledge, the only known framework for constructing PPRGs in NC^0 is one based on Goldreich's OWF [30]. For example, the celebrated work by Applebaum, Ishai, and Kushilevitz [5] on randomized encodings only yields the characterization of *sublinear-stretch* PRGs in NC^0, but characterizing PPRGs in NC^0 seems out of reach at present. Therefore, it is natural to inquire into a new candidate for PPRGs in NC^0 and a characterization result through the lens of the dichotomy between learning and cryptography.

Strictly speaking, we mainly discuss a generator defined as a collection of PPRGs, where the generator has a public index randomly and efficiently (but not in NC^0) selected in the preprocessing (cf. [29, Section 2.4.2]). This relaxed setting is standard, especially when we discuss a PPRG in NC^0 (cf. [9, Remark 1.1]), and such relaxation does not affect the applications mentioned above.

From the perspective of computational learning theory. An ultimate goal in computational learning theory is to identify the *simplest* concept class that is not efficiently learnable under a plausible hardness assumption. Many hardness results of learning in the current frontline are related to PPRGs in NC^0. Applebaum, Barak, and Wigderson [10] proved the hardness of learning $O(\log n)$-junta functions under the existence of PPRGs in NC^0 with an additional assumption on input-output connections. Applebaum and Raykov [11] and Daniely and Vardi [25] proved the hardness of learning for central classes such as depth-3 AC^0 circuits and $\omega(1)$-term DNF formulas under assumptions related to polynomial-stretch Goldreich's PRG, which is a special case where the output bits are computed by one fixed predicate. Oliveira, Santhanam, and Tell [51] proved that a security of polynomial-stretch Goldreich's PRG implies the impossibility of improving parameters of natural properties for simple classes such as DNF-XOR circuits under a plausible assumptions on the existence of suitable expanders, where a natural property is a notion deeply related to learning [18, 19].

Since the equivalence between pseudorandomness and unpredictability follows from the well-known result by Yao [61], a reader might expect a correspondence between PPRG in NC^0 and hardness of learning NC^0. However, this intuition seems incorrect because while a PPRG in NC^0 is conjectured to exist, learning NC^0 (i.e., functions with constant locality) is trivially feasible by applying Occam's razor [15]. In this sense, there seems to exist a gap between pseudorandomness and hardness of learning when we consider considerably low complexity classes such as NC^0. Nevertheless, can we obtain some learning-theoretic characterization for a collection of PPRG in NC^0? In this work, we provide an affirmative answer to this question.

1.1 Our Learning Model

We introduce the learning model mainly discussed in this work, which is a natural variant of the PAC learning model. For the formal definition, see the full paper.

We consider a distribution-specific average-case learning model, introduced by Blum, Furst, Kearns, and Lipton [14]. In this model, an unknown Boolean-valued target function f (contained in some concept class $\mathscr{C}$) is selected according to a known *target distribution*, and a learner is given samples of the form $(x, f(x))$, where x is called an example and selected identically and independently according to a known *example distribution*. After learning with the samples, the learner tries to guess a value of $f(x)$ for an additionally given input x (called a *challenge*) selected according to the same example distribution with good probability; specifically, with probability at least $1/2 + \gamma$ (we refer to γ as an *advantage*) over the choices of randomness for the learner, samples, and a target function. We define the sample complexity as the number of samples the learner requires. We say that a class $\mathscr{C}$ is not learnable with respect to some class (e.g., polynomial-time samplable) of example distributions and target distributions in this distribution-specific model if there exist an example distribution and a target distribution in the class such that $\mathscr{C}$ is not learnable under these example and target distributions.

A new perspective in this paper is to consider parameterized complexity of learning for a parameterized concept class and parameterized classes of example distributions and target distributions. We remark that parameterized learnability has been discussed in certain previous studies (e.g. [12]). The main difference from the previous formulation is the separate consideration of time complexity and sample complexity. In this paper, we only consider fixed-parameter tractability on sample complexity, and the time complexity can be arbitrary polynomial depending on parameters (or sub-exponential functions). Specifically, for parameters $k_1, \ldots, k_c$ on a concept class $\mathscr{C}$ and classes of example distributions and target distributions, we say that $\mathscr{C}$ is learnable with $(k_1, \ldots, k_c)$-FPT samples if $\mathscr{C}$ is learnable

with $f(k_1, \ldots, k_c) \cdot n^{\Theta(1)}$ samples, where f is some computable function. Our learning model captures a (natural) situation in which collecting labeled data is more expensive than using computational resources. This formulation also provides a new perspective on parameterized complexity of learning; e.g., PAC learning k-junta (i.e., functions depending on only k coordinates of the input) is known to be W[2]-hard [12], but feasible with FPT samples (with $k2^k \cdot n^{\Theta(1)}$ samples and in $O(n^k)$ time) by Occam's razor [15]. By contrast, it can be shown that learning degree-d $\mathbb{F}_2$-polynomials is infeasible even in this setting based on the VC theory (cf. [58]).[1]

We define the sparsity of a distribution as the maximum Hamming weight of samples.

▶ **Definition 1.** *For $c \in \mathbb{N}$, we say that a family $D = \{D_n\}_{n\in\mathbb{N}}$ of distributions on $\{0,1\}^*$ is c-sparse if $\Pr_{x\leftarrow D_n}[wt(x) \leq c] \geq 1 - \mathsf{negl}(n)$, where $wt(x)$ represents the Hamming weight of x, and $\mathsf{negl}(n)$ represents some negligible function, i.e., for any polynomial $p(n)$, it holds that $\mathsf{negl}(n) < 1/p(n)$ for any sufficiently large $n \in \mathbb{N}$.*

1.2 Our Results

As a main result, we show that a collection of PPRGs in NC^0 is characterized by the learnability of various central classes with FPT samples with respect to a sparse example distribution and an NC^0-samplable target distribution.

▶ **Theorem 2** (informal)**.** *The following are equivalent:*

1. *There exists a collection of (infinitely-often secure[2]) PPRGs in NC^0.*
2. *c-sparse $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with respect to a target distribution samplable by a depth-d NC^0 circuit and a samplable distribution on c'-sparse example distributions with (c, c', d)-FPT samples.*
3. *c-Fourier-sparse functions are not polynomial-time learnable on average with respect to a target distribution samplable by a depth-d NC^0 circuit and a samplable distribution on c'-sparse example distributions with (c, c', d)-FPT samples.*
4. *For any $f \in \{\mathrm{OR}\} \cup \{\mathrm{MOD}_m : m \in \mathbb{N} \setminus \{1\}\}$, degree-$d$ f-decision trees are not polynomial-time learnable on average with respect to a target distribution samplable by a depth-d' NC^0 circuit and a samplable distribution on c-sparse example distributions with (d, c, d')-FPT samples.*

Informally, Theorem 2 yields a new dichotomy between highly efficient pseudorandom generators and sample-efficient heuristics for learning with sparse data. Below we argue that the learning settings of Theorem 2 are natural.

Concept classes. For the formal descriptions of each parameterized concept class, see the full paper. Here, we remark that the sparsity of $\mathbb{F}_2$-polynomials and Fourier representations is one of the most important complexities of Boolean functions (cf. [48]). The fourth item above concerns the extensions of decision trees, containing the well-studied class of parity

[1] In the full paper, we show that learning degree-d $\mathbb{F}_2$-polynomials with FPT samples is infeasible even in the *average-case* setting over uniformly random degree-d $\mathbb{F}_2$-polynomials.

[2] In this paper, we mainly discuss the relationships between learnability for all example sizes and PPRGs with infinitely often security (i.e., the security holds for infinitely many seed lengths). Note that the same results hold for generators with sufficiently large security (i.e., the security holds for any sufficiently large seed length) by considering the learnability on infinitely many example sizes.

decision trees[3](e.g. [41]). Although OR decision trees have received relatively less research attention compared with the other concepts, learning OR decision trees with sparse data seems to be a natural setting where the decision is made by a few queries about whether some unusual features are observed. Interestingly, our result shows that the average-case learnability for these various concepts becomes equivalent when data are sparse through the existence of a collection of PPRGs in NC^0.

Example distributions. We remark two points. First, we consider a *distribution of* example distributions (i.e., average cases on example distributions), where the example distribution is selected at the initialization step (see the full paper for the formal description). Note that this captures more general settings of learning than the previous distribution-specific setting in [14]; e.g., our framework captures a distribution determined by some hidden random parameter. From the perspective of cryptography, the hardness assumption on a distribution of example distributions is weaker than ones in distribution-specific settings. Second, we consider learning on sparse example distributions. Such a learning framework naturally captures learning on data with rarely observed features, such as symptoms of patients.

Target distributions. We consider NC^0-samplable distributions as target distributions, and this is a natural assumption in average-case complexity theory in learning; e.g., the uniform distribution over functions in $\mathscr{C}$ is often regarded as a projection of random strings onto the binary representations for functions in $\mathscr{C}$ (e.g., random DNFs), and almost all target distributions considered in previous studies on average-case learning are NC^0-samplable [36, 56, 57, 35, 2].

We also remark that Theorem 2 holds even in super-polynomial regimes; e.g., sub-exponential-time average-case hardness of learning with FPT samples corresponds to a collection of PPRGs secure against sub-exponential-time adversaries (where the loss of security is only polynomial). Note that super-polynomial security is applied for the construction of $i\mathcal{O}$ based on well-founded assumptions [37, 38]. Particularly, Jain, Lin, and Sahai [37] assumed (i) the hardness of learning problems LWE and LPN, (ii) the existence of a collection of PPRGs in NC^0, and (iii) the Diffie-Hellman-style assumption (i.e., SXDH). Our result characterizes assumption (ii) based on the hardness of learning and, along with their work, opens an interesting research direction: Is the well-founded hardness assumption of learning sufficient for constructing $i\mathcal{O}$ (i.e., Obfustopia)?

Next, we present several related results on the hardness of learning and PPRGs in relaxed complexity classes, which are obtained by relaxing some conditions in Theorem 2.

On removing sparsity conditions. Although Theorem 2 shows one characterization of a collection of PPRGs in NC^0 by learnability with sparse data, the sparsity is somewhat restrictive, and there exist a large amount of non-sparse data in the real world. As a second result, we show that learnability with non-sparse data for the classes in Theorem 2 still characterizes a collection of PPRGs in superclasses of NC^0.

[3] In fact, the equivalence between constant-depth parity decision trees and constant-Fourier-sparse functions follows from the work by Kushilevitz and Mansour [41]. However, it is unclear whether these learning settings are equivalent when we restrict the target distributions to NC^0-samplable because the transformation between these representations may be infeasible in NC^0.

▶ **Theorem 3** (informal). *The following hold:*

1. *There exists a collection of PPRGs in $O(1)$-sparse $\mathbb{F}_2$-polynomials iff c-sparse $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with respect to a target distribution samplable by a c'-sparse $\mathbb{F}_2$-polynomial and a samplable distribution on example distributions with (c, c')-FPT samples.*
2. *There exists a collection of PPRGs in $O(1)$-Fourier-sparse functions iff c-Fourier sparse functions are not polynomial-time learnable on average with respect to a target distribution samplable by a c'-Fourier sparse functions and a samplable distribution on example distributions with (c, c')-FPT samples.*

The generators above still have good parallelism in the sense that each output bit is computable by a constant number of parallel and simple computations (i.e., logical AND or logical XOR).

On obtaining a single PPRG. The theorems above hold only in the case of a collection of PPRGs, and the learning-theoretic characterization of a single PPRG is currently open. Although a collection of PPRGs is standard in parallel cryptography, a single parallel PPRG is still a natural and desirable primitive because it does not require the additional public random strings.

As a third result, we show that if we allow NC^0 circuits to have one top-most XOR-gate with unbounded fan-in, where the other types of gates (i.e., NOT, OR, and AND) have bounded fan-in (we denote this class[4] by $\oplus$-NC^0), then a single PPRG in $\oplus$-NC^0 is characterized by the hardness of learning constant-degree $\mathbb{F}_2$-polynomials on the *uniform* example distribution.

▶ **Theorem 4** (informal). *For any polynomial $r(n)$, the following are equivalent:*

1. *There exists a PPRG in $\oplus$-NC^0.*
2. *Degree-d $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with respect to a uniform example distribution and a target distribution samplable by a degree-d' $\mathbb{F}_2$-polynomial using $r(n)$-bit random seeds with (d, d')-FPT samples.*

We remark several points. First, in the theorem above, the length of the seeds for selecting a target function is also fixed to some polynomial $r(n)$ independent of the parameters (i.e., degree of $\mathbb{F}_2$-polynomials). This restriction is essential for the result because if we remove this restriction, then unlearnability with FPT samples holds unconditionally even for time-unbounded learners (the formal proof can be found in the full version). Second, the average-case hardness of learning on the uniform example distribution is equivalent to weak pseudorandom functions (WPRFs), where a WPRF is an efficiently samplable family of functions indistinguishable from a random function on inputs passively selected uniformly at random [46]. Thus, Theorem 4 can also be regarded as the equivalence between PPRG and WPRF within the class $\oplus$-NC^0.

Finally, we show that if we consider a general case of samplable distributions of example distributions (instead of the uniform example distribution), then the dichotomy in Theorem 4 is extended to a collection of PPRGs in $\oplus$-NC^0. In other words, we can characterize the difference between a single PPRG and a collection of PPRGs in $\oplus$-NC^0 by the difference in the generality of example distributions on the hardness of learning.

[4] It is not hard to verify that $\oplus$-NC^0 is indeed equivalent to $\mathsf{NC}^0[\oplus]$ (i.e., a class of NC^0 circuits with XOR-gates with unbounded fan-in) and a class of constant-degree $\mathbb{F}_2$-polynomials.

▶ **Theorem 5** (informal). *For any polynomial $r(n)$, the following are equivalent:*

1. *There exists a collection of PPRGs in $\oplus$-NC^0.*
2. *Degree-d $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with respect to a target distribution samplable by a degree-d' $\mathbb{F}_2$-polynomial using $r(n)$-bit random seeds and a samplable distribution on example distributions with (d, d')-FPT samples.*

Note that Theorems 3–5 also hold in super-polynomial regimes with polynomial security loss.

Theorems 2–5 indicate that by selecting a parameterized example distribution and a parameterized target distribution arbitrarily and by assuming the hardness of learning with FPT samples, we can construct a secure parallel PPRG. Conversely, if we believe in PPRGs in the correspondence class, then such a hard-to-learn parameterized setting must exist. However, we remark that Theorems 2–5 are general results on the dichotomy between the hardness of learning and parallelly computable PPRGs, and they do not explicitly specify the distributions with respect to which learning is hard on average with FPT samples.

Here, we propose a natural learning task, learning random parity decision trees, whose hardness does not contradict our current knowledge.

▶ **Definition 6** (Learning random parity decision trees). *Let $D = \{D_n\}_{n\in\mathbb{N}}$ be an arbitrary example distribution, where D_n is a distribution on $\{0,1\}^n$ for each $n \in \mathbb{N}$. For any $d \in \mathbb{N}$ and $m\colon \mathbb{N} \to \mathbb{N}$, we define a problem of learning random depth-d parity decision trees (d-LRPDT) on D with $m(n)$ samples as follows:*

Input: samples $\{(x^{(i)}, T(x^{(i)}))\}_{i\in\{1,\dots,m(n)\}}$ and a challenge x_c, where $x^{(1)}, \dots, x^{(m(n))}, x_c \in \{0,1\}^n$ are selected according to D_n, and T is a random parity decision tree of depth d and size 2^d in which each query at internal nodes is $\oplus_{i\in S} x_i$ for a uniformly random subset $S \subseteq \{1,\dots,n\}$ (selected independently for each node) and each leaf is labeled by a uniformly random value in $\{0,1\}$ (selected independently for each leaf).
Output: $T(x_c)$

For any polynomial $m(n)$ and $p(n)$, we say that d-LRPDT is $(m(n), 1/p(n))$-hard on D if no randomized polynomial-time algorithm solves d-LRPDT on D with $m(n)$ samples with probability at least $1/2 + 1/p(n)$, i.e., for any randomized polynomial-time algorithm A and sufficiently large $n \in \mathbb{N}$,

$$\Pr_{A,D_n,T}\left[A\left((x^{(1)}, T(x^{(1)})), \dots, (x^{(m(n))}, T(x^{(m(n))})), x_c\right) = T(x_c)\right] < \frac{1}{2} + \frac{1}{p(n)}.$$

By Theorem 2, if d-LRPDT is hard with FPT samples on some parametrized sparse example distribution, then a collection of PPRGs exists in NC^0. By inspecting our proof, we show that the sample complexity can be made as small as $n^{1+\epsilon}$ for an arbitrarily small constant $\epsilon > 0$.

▶ **Corollary 7.** *Let $\epsilon \in (0,1)$ be an arbitrary constant. Suppose that there exist $d \in \mathbb{N}$ and an example distribution D such that d-LRPDT is hard on D with $n^{1+\epsilon}$ samples[5]. Then, we can construct parallel PPRGs according to the complexity of D as follows:*

- *If D is $O(1)$-sparse and samplable, then a collection of PPRGs in NC^0 exists.*
- *If D is the uniform distribution, then a PPRG in $\oplus$-NC^0 exists.*
- *If D is samplable, then a collection of PPRGs in $\oplus$-NC^0 exists.*

The first and third items hold even for samplable distributions on example distributions.

[5] For the requirement for the advantage of learning, see the full version.

For instance, as a natural candidate for $O(1)$-sparse example distributions, we propose the uniform distribution over binary strings of Hamming weight $c \in \mathbb{N}$.

▶ **Corollary 8.** *If there exist $c, d \in \mathbb{N}$ and $\epsilon \in (0, 1)$ such that d-LRPDT is hard on the uniform example distribution over binary strings of Hamming weight c with $n^{1+\epsilon}$ samples, then a collection of PPRGs in NC^0 exists.*

We remark that it is consistent with our knowledge that d-LRPDT cannot be solved. Depth-d parity decision trees are exactly learnable by the Goldreich–Levin algorithm when additional query access to the target function (i.e., membership query) is available [32, 41]. However, it is a central open question whether the membership query is necessary, and d-LRPDT is a natural test case for this question. An efficient learner for random log-depth decision trees was developed by Jackson and Servedio [36], but it is unclear whether this algorithm can be extended to the case of random parity decision trees. From Corollary 7, we propose further learning-theoretic and cryptographic analysis of the hardness of learning parity decision trees as a future research direction. Particularly, one important property of the PPRGs constructed in Corollary 7 is that the output bits are computed by various predicates. Therefore, they seem to resist an attack that depends on a specific property of one fixed predicate, even in the setting in Corollary 8.

1.3 Related Work

Applebaum, Barak, and Wigderson [10] proved the hardness of learning $O(\log n)$-junta functions under the existence of PRGs in NC^0 with an additional assumption that (roughly speaking) a small subset of output bits can be embedded indistinguishably with good local expansion. Applebaum and Raykov [11] proved the hardness of learning depth-3 AC^0 circuits under the assumption related to polynomial-stretch Goldreich's PRGs, which matches the unconditional upper bound presented in [42]. We remark that their assumption is reducible to a more reliable assumption on Goldreich's OWFs due to the search-to-decision reduction developed in [9, 11], where they essentially use the structures of Goldreich's OWFs. Daniely and Vardi [25] showed the hardness of learning $\omega(1)$-term DNF formulas and related classes on a product example distribution by assuming Goldreich's PRG for arbitrary polynomial stretch. We remark that our results are incomparable with these previous studies. We assume the existence of the more general cryptographic primitive (i.e., a collection of PPRGs in NC^0) to show the hardness of learning other simple and central classes. This generalization weakens the hardness result to a more general class of example distributions instead of product distributions compared with [25], while we can also obtain the opposite direction from the hardness of learning to cryptography. The result of [51] on natural properties also differs in the learning setting, particularly natural properties essentially correspond to learning with membership queries on the uniform example distribution [18].

Blum, Furst, Kearns, and Lipton [14] constructed OWFs, PRGs, and private-key encryption schemes based on the average-case hardness of learning. To construct PPRGs by using their technique, we need to assume a stronger hardness assumption on learning with membership queries. The use of membership queries was removed by Naor and Reingold [46], and we apply the same technique to show one direction in Theorem 4. Note that the complexity of these PPRGs depends on the complexity of evaluating concept classes. Thus, this approach does not seem to yield a PPRG in NC^0 because if a concept class has the evaluation performed in NC^0, then such a class is trivially learnable. The followup studies [47, 50, 55, 44, 45] discussed relationships between cryptography and hardness of learning in P and $\mathsf{P/poly}$. Other studies (e.g. [53]) developed various cryptographic schemes based on the hardness of

learning linear functions with noise, but it is not clear whether PPRGs in NC^0 are obtained as a consequence of these studies. LRPDT is regarded as a related problem in which we learn parity with noise determined by a constant number of other parities, and it is indeed reducible to learning parity with noise in the case of a uniform example distribution [26].

With regard to parallel cryptography, the constructions of PRGs in NC^0 were presented by Applebaum, Ishai, and Kushilevitz [5] (sublinear-stretch) and Applebaum, Ishai, and Kushilevitz [6] (linear-stretch). Recently, Ren and Santhanam [54] and Liu and Pass [43] characterized the existence of OWF in NC^0 based on the average-case meta-complexity notion, which only yields sublinear-stretch PRGs in NC^0, and PPRGs in NC^0 seem out of reach at present. Some candidates for a collection of PPRGs in NC^0 were studied by Cook, Etesami, Miller, and Trevisan [20], Bogdanov and Qiao [16], Applebaum, Bogdanov, and Rosen [4], Applebaum [9], O'Donnell and Witmer [49], Applebaum and Lovett [8], and Couteau, Dupin, Méaux, Rossi, and Rotella[21] based on the framework of Goldreich's OWF [30]. This type of generator is natural but somewhat restrictive in the sense that all output bits are computed by the same predicate fixed in advance. One advantage of the previous framework is that the security of the generator can be based on a hardness notion of one-wayness, which is more reliable than pseudorandomness [9].[6] By contrast, an advantage of the framework proposed in this study is that the output bits of the resulting generator are computed by various predicates; thus, it seems to resist an attack that depends on a specific property of one fixed predicate.

We will introduce a key notion of FPT dualization with the junta-sparse condition in Section 2, and it seems conceptually related to the analysis of Boolean functions on Hamming balls and slices (i.e., substrings of fixed Hamming weight). Particularly, Filmus and Ihringer [28] and Filmus [27] proved that every constant-degree polynomial on a slice is also $O(1)$-junta on the same slice. By contrast, our result can also be rephrased as that every sparse polynomial on a Hamming ball is a *dual* of $O(1)$-junta.

2 Techniques

In this section, we present an overview of key notions and proof sketches of the main results.

2.1 Proof Techniques for Theorems 2 and 3

The key notion to show Theorems 2 and 3 is the dualization of concept classes, which was explicitly discussed independently by Applebaum, Barak, and Wigderson [10] and Vadhan [59] and applied (implicitly or explicitly) in recent studies on the hardness of learning [24, 23, 44, 45, 25]. Informally, the dualization of a concept class $\mathscr{C}$ consists of two mappings from examples to target functions in $\mathscr{C}$ and from target functions in $\mathscr{C}$ to examples satisfying the following condition. If an example x (resp. a target function $f \in \mathscr{C}$) is mapped to a target function $x^* \in \mathscr{C}$ (resp. an example f^*) by these mappings, then the value of $x^*(f^*)$ is equal to $f(x)$. We refer to this x^* (resp. f^*) as a dual of x (resp. f) and use the superscript $*$ to represent duals.

First, we observe that the dualization of a concept class $\mathscr{C}$ provides a relationship between a collection of PRGs and learnability for $\mathscr{C}$. On the one hand, if there exists a collection G of PRGs in $\mathscr{C}$, then we can construct a sample set of size m from the pseudorandom string

[6] In terms of learning, the difference between one-wayness and pseudorandomness is similar to the difference between proper learning and improper learning. In general, proper learning is often harder than improper learning (cf. [52]).

$y = G(x)$ of length m (where x is a random seed) as $\{(G_i^*, y_i)\}_{i\in[m]}$, where $G_i \in \mathscr{C}$ represents the function computing the i-th bit of G, and G_i^* is its dual. Notice that $x^*(G_i^*) = G_i(x) = y_i$ for each $i \in [m]$. Therefore, if we consider this x^* as a target function for learning $\mathscr{C}$ and the uniform distribution over the samples as the example distribution, any feasible learner cannot distinguish these labels from random labels unless the learner looks at almost all samples in the set. On the other hand, we can obtain a collection of PRGs from the problem of learning $\mathscr{C}$ by translating a sample set $\{(x^{(i)}, f(x^{(i)}))\}_{i\in[m]}$ (where f is a target function) into a generator $G(f^*) = (x^{(1)})^*(f^*) \circ \cdots \circ (x^{(m)})^*(f^*)$. By the equivalence between pseudorandomness and unpredictability [61], if learning $\mathscr{C}$ is hard even with non-negligible advantage, then the value of $G(f^*) = f(x^{(1)}) \circ \cdots \circ f(x^{(m)})$ must be pseudorandom. If we assume that the target distribution is samplable in a complexity class $\mathscr{C}'$ and regard the seed to the sampler as a random seed to G, then we can implement this G in $\mathscr{C} \circ \mathscr{C}'$.

At a high level, we will use the argument above to show Theorems 2 and 3. However, there are the obstacles. First, the argument from PRG to the hardness of learning only yields hardness of learning with a fixed sample complexity depending on the stretch of the PRG. Second, more importantly, NC^0 cannot be dualized. Intuitively, for an NC^0-computable $f\colon \{0,1\}^n \to \{0,1\}$ (i.e., f depends on only $O(1)$ coordinates) and input $x \in \{0,1\}^n$, the value of $f(x)$ depends on $\Omega(\log n)$-bit information of f, such as relevant coordinates. Thus, we cannot regard $f(x)$ as a function depending on only $O(1)$-bit information in a representation of f. In the full paper, we formally show the impossibility of the dualization of NC^0 based on the lower bound on communication complexity. Below we present how we deal with these two obstacles.

FPT Dualization

We deal with the first obstacle by assuming polynomial-stretch PRGs. The merit of a PPRG is that we can amplify the stretch of a PRG to an arbitrary polynomial within NC^0 by applying the original generator constant times based on the GGM construction [31]. After applying the original generator computable by a depth-d circuit c times, the depth of the generator increases up to cd, whereas c affects the exponent of the stretch of the PRG. Intuitively, this observation leads to the hardness of learning with FPT samples for a parameter involved in the depth.

To apply the dualization technique above in the parameterized setting, we extend the notion of dualization to the parameterized setting as follows. For any parameterized concept class $\mathscr{C}$, we use a subscript and superscript to refer to an input size and a parameter, respectively.

▶ **Definition 9** (FPT dualizable). *Let $\mathscr{C}^k$ be a parameterized concept class. We say that $\mathscr{C}$ is fixed-parameter tractably (FPT) dualizable if there exist a polynomial $p_{dual}\colon \mathbb{N} \to \mathbb{N}$, computable functions $f_1, f_2\colon \mathbb{N} \to \mathbb{N}$, and polynomial-time computable mappings $g\colon \mathbb{N} \times \{0,1\}^* \to \mathscr{C}$ and $h\colon \mathbb{N} \times \mathscr{C} \to \{0,1\}^*$ such that for any $k, n \in \mathbb{N}$, $x \in \{0,1\}^n$, and $f \in \mathscr{C}_n^k$, the following hold: (i) $g(k,x) \in \mathscr{C}^{f_2(k)}_{f_1(k)\cdot p_{dual}(n)}$, (ii) $h(k,f) \in \{0,1\}^{f_1(k)\cdot p_{dual}(n)}$, and (iii) $(g(k,x))(h(k,f)) = f(x)$.*

We use the notation $x^{*(k)}$ or x^* (resp. $f^{*(k)}$ or f^*) to refer to $g(k,x)$ (resp. $h(k,f)$) in the definition above; e.g., the third condition above can be written as $x^*(f^*) = f(x)$ for each f and x.

Junta-Sparse Condition

At a high level, the idea to overcome the second obstacle is applying the dualization of superclasses of NC^0 and focusing on its substructure, i.e., the correspondence between NC^0 and a subset of strings, particularly in our case, sparse strings. To formalize this idea, we introduce a key condition of FPT dualization named the *junta-sparse condition*, which serves as dualization of NC^0 partially in the actual dualization of the superclass. Intuitively, the junta-sparse condition claims that (i) any $O(1)$-junta function (i.e., a function that depends on only $O(1)$ coordinates) is contained in the concept class, and (ii) $O(1)$-junta functions and strings of constant Hamming weight get interchanged by the FPT dualization. The condition is formally stated as follows:

▶ **Definition 10** (junta-sparse condition). *Let $\mathscr{C}^k$ be an FPT dualizable class. We say that $\mathscr{C}$ satisfies the junta-sparse condition if the following hold:*

1. *There exist computable functions $g, h\colon \mathbb{N} \to \mathbb{N}$ such that for any $k \in \mathbb{N}$ and any k-junta f, it holds that $f \in \mathscr{C}^{g(k)}$ and $wt(f^*) \leq h(k)$.*
2. *There exists a computable function $g\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for any $c, k \in \mathbb{N}$ and any $x \in \{0,1\}^*$ with $wt(x) \leq c$, it holds that $x^{*(k)}$ is $g(c,k)$-junta.*

For instance, we show a class of c-sparse $\mathbb{F}_2$-polynomials is FPT dualizable and satisfies the junta-sparse condition. Fix $n, c \in \mathbb{N}$ with $c \leq 2^n$ arbitrarily. Let $f\colon \{0,1\}^n \to \{0,1\}$ be an $\mathbb{F}_2$-polynomial of sparsity c, i.e., f is written as $f(x) = M_1(x) + \ldots + M_c(x)$ (under operations of $\mathbb{F}_2$), where $M_1, \ldots, M_s$ are monomials. Then, we define the dual of f (i.e., $h(c,f)$ in Definition 9) as the following $c(n+1)$-bit string f^* indexed by $\{0,1,\ldots,n\} \times \{1,\ldots,c\}$:

$$f^*_{(i,j)} = \begin{cases} \mathbb{1}(x_i \in M_j) & \text{if } i \neq 0 \\ \mathbb{1}(M_j \equiv 1) & \text{if } i = 0 \end{cases}$$

For each input $x \in \{0,1\}^n$, we also define the dual of x (i.e., $g(c,x)$ in Definition 9) as the following function $x^*\colon \mathbb{F}_2^{c(n+1)} \to \mathbb{F}_2$:

$$x^*(f^*) = \sum_{j \in \{1,\ldots,c\}} \prod_{i : x_i = 1} f^*_{(i,j)} + \sum_{j \in \{1,\ldots,c\}} f^*_{(0,j)}.$$

Then, we can verify the correctness of the FPT dualization as follows:

$$\begin{aligned} x^*(f^*) &= \sum_{j \in \{1,\ldots,c\}} \prod_{i : x_i = 1} f^*_{(i,j)} + \sum_{j \in \{1,\ldots,c\}} f^*_{(0,j)} \\ &= \sum_{j \in \{1,\ldots,c\}} \left(\prod_{i : x_i = 1} \mathbb{1}(x_i \in M_j) + \mathbb{1}(M_j \equiv 1) \right) \\ &= \sum_{j \in \{1,\ldots,c\}} M_j(x) \\ &= f(x). \end{aligned}$$

In addition, we can easily verify the junta-sparse condition as follows. Any k-junta function f is represented as an $\mathbb{F}_2$-polynomial of degree k and sparsity 2^k. By the construction of f^*, the Hamming weight of the dual of any $\mathbb{F}_2$-polynomial of degree k and sparsity 2^k is at most $2^k \cdot k$. In addition, for any input $x \in \{0,1\}^n$ to a c-sparse $\mathbb{F}_2$-polynomial, the dual of x depends on only $c \cdot wt(x) + c$ coordinates by the construction of x^*.

Meta-Theorem

The proof of Theorem 2 consists of the following two parts. As the first step, we prove meta-theorem which shows that if a parameterized concept class $\mathscr{C}$ is FPT dualizable by mappings computable in NC^0 and it satisfies the junta-sparse condition, then the existence of a collection of PPRGs in NC^0 corresponds to the average-case hardness of learning $\mathscr{C}$ with FPT samples with respect to (a samplable distribution of) sparse example distributions and an NC^0-samplable target distribution[7]. Note that verifying the condition in the meta-theorem (i.e., dualization with the junta-sparse condition) is purely a puzzle-like problem involved in representation for Boolean functions and directly related to neither learning theory nor cryptography, as seen in the case of c-sparse $\mathbb{F}_2$-polynomials. Namely, if you can solve the puzzle for some concept class $\mathscr{C}$, then it automatically implies the equivalence between the existence of a collection of PPRGs in NC^0 and the average-case hardness of learning $\mathscr{C}$ with sparse data based on our meta-theorem. As the second step to show Theorem 2, we solve this puzzle, i.e., demonstrate that concept classes in Theorem 2 (i.e., c-sparse $\mathbb{F}_2$-polynomials, c-Fourier-sparse functions, and depth-d $\{\mathrm{OR},\mathrm{Mod}_m\}$-decision trees) are FPT dualizable by NC^0-computable mappings and satisfy the junta-sparse condition.

We show the outline of the proof of the meta-theorem based on the argument mentioned at the beginning of this subsection.

A collection of PPRGs in $\mathsf{NC}^0 \Rightarrow$ *hardness of learning*: Suppose that there exists a collection G of PPRGs. For contradiction, we assume that there exists an efficient learner L for $\mathscr{C}$ that requires only FPT samples. We amplify the stretch of G by the GGM construction [31] within NC^0, let G' be the amplified generator, and construct the sample set S from the duals of G' and a pseudorandom string $y = G'(x)$. Since G' is computable in NC^0, each function computing each bit of G' is $O(1)$-junta. Thus, by the junta-sparse condition, the Hamming weight of each example is bounded above by a constant (depending on the depth of G'). In addition, since the mappings in FPT dualization are computable in NC^0, the target distribution of the dual of the random seed x is NC^0-samplable. Thus, the learning problem on the uniform distribution over the samples in S is a valid setting for L. Let c be the number of applications of G to construct G'. Then, the sample complexity of L increases in the sense of FPT for c, whereas c affects the exponent of the stretch of G'. Therefore, for a sufficiently large $c \in \mathbb{N}$, the learner L cannot read a large fraction of S. Thus, L can predict some bit in $G'(x)$ from other bits, and this contradicts that G is PRG.

Hardness of learning $\Rightarrow$ *a collection of PPRGs in* NC^0: Suppose that learning $\mathscr{C}$ is hard on average with FPT samples. Since the target distribution is NC^0-samplable, each bit of the representation of $\mathscr{C}$ depends on only constant bits of a random seed. By the technical assumption (in footnote 7) on the FPT upper bound on the length of the representation of $\mathscr{C}$, we can assume that the length of the seed for the target distribution is bounded above by some FPT function. Using the hardness assumption for a sample complexity $m(n)$ polynomially larger than the upper bound on the length of the seed, we construct the collection G of PRGs by taking duals of examples. Remember that the input size of G is the length of the seed for the target distribution, and the output size is $m(n)$. Thus, G has polynomial-stretch. In addition, the Hamming weight of the examples is constant except with negligible probability by the hardness assumption. Thus, by the sparse-junta condition, each bit of G is $O(1)$-junta, and G is implemented in NC^0. Technically, when we

[7] Strictly speaking, we also need a technical assumption that the length of the binary representation for $\mathscr{C}$ is bounded above by some FPT function.

consider the advantage in learning, this argument only yields a collection of PPRGs with a fixed indistinguishable parameter. We can convert such a collection of weak PPRGs into a collection of standard PPRGs (with a negligible indistinguishable parameter) by applying the technique by Applebaum and Kachlon [7].

Theorem 3 is shown based on the following observation: If a concept class $\mathscr{C}$ is FPT dualizable and closed under the composition (where the junta-sparse condition is no longer needed), the above argument yields the equivalence between a collection of PPRGs in $\mathscr{C}$ and the average-case hardness of learning $\mathscr{C}$ with FPT samples.

2.2 Proof Techniques for Theorem 4

Theorem 4 shows the equivalence between the existence of a (single) PPRG in $\oplus$-NC^0 and the average-case hardness of learning constant-degree $\mathbb{F}_2$-polynomials with FPT samples with respect to a uniform example distribution and a target distribution samplable by a constant-degree $\mathbb{F}_2$-polynomial. In fact, $\oplus$-NC^0 is equivalent to the class of constant-degree $\mathbb{F}_2$-polynomials because (i) any constant-degree $\mathbb{F}_2$-polynomial is implemented by a $\oplus$-NC^0 circuit that first computes monomials in parallel and takes the summation of them by the top-most XOR gate, and (ii) any $\oplus$-NC^0 circuit is implemented by a constant-degree $\mathbb{F}_2$-polynomial by expressing each sub-circuit connected to the top-most XOR-gate as a constant-degree $\mathbb{F}_2$-polynomial (note that the top-most XOR-gate does not increase the degree of the resulting $\mathbb{F}_2$-polynomial). Therefore, we only need to establish the relationship between a PPRG and learnability within the class of constant-degree $\mathbb{F}_2$-polynomials.

Before presenting the idea, we briefly explain why we cannot apply the dualization techniques in Section 2.1 directly to show Theorem 4. In fact, the class of degree-d $\mathbb{F}_2$-polynomials is simply dualizable as follows: for any degree-d $\mathbb{F}_2$-polynomial $f(x) = \sum_{S:|S|\leq d} f_S \prod_{i\in S} x_i$, where f_S represents the coefficient of f on $\prod_{i\in S} x_i$, we regard the coefficients of f as the input and the value of $\prod_{i\in S} x_i$ as a coefficient on the monomial f_S for each subset S, i.e., the dual of x is a degree-1 $\mathbb{F}_2$-polynomial taking the coefficients of f as the input. An issue is that this dualization is no longer FPT in the sense that each n-input degree-d polynomial is converted into a string of length $\sum_{i=1}^{d} \binom{n}{i} = \Theta(n^d)$. If we apply this dualization in the argument in Section 2.1, then a parameter affects the exponent of the sample complexity of learners, and this causes several problems: e.g., in the direction from PPRG to the hardness of learning, we cannot prepare a sufficient number of samples using the GGM construction so that the learner cannot read the entire sample set. In addition, the argument in Section 2.1 yields only a collection of PPRGs.

An alternative to show the direction from a PPRG to hardness of learning is to construct an $\mathbb{F}_2$-polynomial pseudorandomly. As a preliminary observation, if we select a polynomial f uniformly at random from all n-input $\mathbb{F}_2$-polynomials of degree d, then for $m = \frac{1}{2}\sum_{i=1}^{d} \binom{n}{i}$ inputs $x^{(1)}, \ldots, x^{(m)} \in \{0,1\}^n$ selected uniformly at random, we can show that the distribution of $f(x^{(1)}), \ldots, f(x^{(m)})$ is statistically close to an m-tuple of random bits even when $x^{(1)}, \ldots, x^{(m)}$ are given. In the formal proof, we verify this by applying the results obtained by Ben-Eliezer, Hod, and Lovett [13]. For now, we assume this. Then, we observe that even if we select a degree-d $\mathbb{F}_2$-polynomial f by a pseudorandom string generated by a PPRG, the labels of the sample set $\{(x^{(i)}, f(x^{(i)}))\}$ must be computationally indistinguishable from random labels. By the equivalence of pseudorandomness and unpredictability [61], such a pseudorandom $\mathbb{F}_2$-polynomial f must be unpredictable.

Based on the argument above, we can create a hard learning problem with FPT samples based on a PPRG G, as follows. For contradiction, we assume that there exists an efficient learner L that requires only FPT samples. Then, we use the GGM construction to amplify

the stretch of G, let G' denote the amplified PRG, and select a pseudorandom $\mathbb{F}_2$-polynomial using G'. Remember that the number c of applications of G affects the exponent of the stretch of G'. Thus, for each $d \in \mathbb{N}$, we can select a sufficiently large c such that a degree-d pseudorandom $\mathbb{F}_2$-polynomial can be selected by G'. Note that G' is still computable by an $\mathbb{F}_2$-polynomial of degree d^c. We regard this G' as a sampling algorithm for selecting a target function in degree-d $\mathbb{F}_2$-polynomials. For the degree-d pseudorandom $\mathbb{F}_2$-polynomial, we can retrieve $\frac{1}{2}\sum_{i=1}^{d}\binom{n}{i} = \Theta(n^d)$ samples that are hard to predict. By contrast, each d determines c and the degree of the sampling algorithm for the target distribution; thus, d affects the required number of samples only in the FPT sense. Therefore, by taking a sufficiently large d, we can prepare a sufficient number of samples for L, and L yields an efficient adversary for G' and G. This is a contradiction.

To show the opposite direction from the average-case hardness of learning to a PPRG, we apply the idea presented by Naor and Reingold [46]. First, we observe that for each constant-degree $\mathbb{F}_2$-polynomial f and input x, the value of $f(x)$ is evaluated by a constant-degree $\mathbb{F}_2$-polynomial taking x and the binary representation of f as the input (where we naturally assume that each f is represented by the coefficients of f). Then, the construction of a PPRG G is outlined as follows. We use the hardness assumption for a sample complexity $m(n)$ sufficiently larger than $(n + r(n))^2$, where $r(n)$ is the upper bound on the seed length for the target distribution in Theorem 4. Let $R = n + r(n)$. Then, G selects R^2 examples $x^{(1)}, \ldots, x^{(R^2)}$ and R^2 target functions $f^{(1)}, \ldots, f^{(R^2)}$ according to the hard example distribution and target distribution by using its own random seed. Then, G outputs R^4 bits $f^{(i)}(x^{(j)})$ for each $i, j \in \{1, \ldots, R^2\}$ as a pseudorandom string. We can prove the pseudorandomness of G using the hybrid argument and the equivalence between unpredictability and pseudorandomness [61]. Since G requires only a $R^2(n+r(n))$-bit random seed to select the examples and the target functions, G stretches an R^3-bit random seed into an R^4-bit pseudorandom string. Thus, G has polynomial-stretch. Note that we apply the standard padding technique to obtain a PPRG defined on all input lengths. Since the sampling algorithm for the target distribution and the evaluation algorithm are computable by constant-degree $\mathbb{F}_2$-polynomials, this generator is implemented by a constant-degree $\mathbb{F}_2$-polynomial by taking composition. Thus, we obtain a PPRG computable by a constant-degree $\mathbb{F}_2$-polynomial. Note that the construction in the formal proof is more complicated because we apply the XOR lemma to amplify the success probability of the adversary to the desired advantage of a learner.

2.3 Proof Ideas for Theorem 5

Theorem 5 shows the equivalence of a collection of PPRGs in $\oplus$-NC^0 and the average-case hardness of learning constant-degree $\mathbb{F}_2$-polynomials with FPT samples with respect to (a samplable distribution of) example distributions and a target distribution samplable by a constant-degree $\mathbb{F}_2$-polynomial. One direction from the average-case hardness of learning to a collection of PPRGs is shown in the same way as in Section 2.2 except that the sampling algorithm for the example distributions is simulated during preprocessing, where the examples are hardwired in the generator.

We present a rough idea to show the other direction from a collection of PPRGs to the hardness of learning. Note that we cannot apply the technique in Section 2.2 because the sampler of generators cannot be implemented in constant-degree $\mathbb{F}_2$-polynomials in general, and the sampling algorithm for selecting a pseudorandom $\mathbb{F}_2$-polynomial is not always implemented in constant-degree $\mathbb{F}_2$-polynomials. Thus, we take the strategy based on FPT dualization again. As discussed in Section 2.2, it is unclear whether FPT dualization of

constant-degree $\mathbb{F}_2$-polynomials is feasible. However, to show the direction from a PPRG to hardness of learning based on the argument in Section 2.1, the type of functions we need to dualize is restrictive, i.e., composite functions of the original pseudorandom generator G (in the GGM construction). We apply this observation to avoid the obstacle involved in the dualization of general constant-degree $\mathbb{F}_2$-polynomials.

The outline follows the argument in Section 2.1. Let G' be the collection of PPRGs obtained by applying G c times to amplify the stretch. We create the sample set from G' and a pseudorandom string $y = G'(x)$, where each example corresponds to the dual of the function computing each bit of G'. Intuitively, for each position i, we define the dual of the i-th bit of G' as c concatenated descriptions of G that are relevant for computing the i-th bit of G'. Then, we consider a target function as a constant-degree $\mathbb{F}_2$-polynomial that computes the description of G' by taking the composition of the given descriptions of G and then applies the random seed x, where we regard this x to be hardwired by another constant-degree $\mathbb{F}_2$-polynomial given x as the input. We regard the latter $\mathbb{F}_2$-polynomial as the sampling algorithm for the target distribution. Consequently, we can prevent the dependence of c and the degree d of G' on the exponent of the input size and the sample complexity in learning. By contrast, c affects the exponent of the stretch of G'. Thus, based on the similar argument as in Section 2.1, we can show the average-case hardness of learning by selecting sufficiently large c.

In the full version, we present the formal proofs based on the ideas above.

References

1. A. Akavia, A. Bogdanov, S. Guo, A. Kamath, and A. Rosen. Candidate Weak Pseudorandom Functions in $AC^0 \circ MOD_2$. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 251–260, New York, NY, USA, 2014. Association for Computing Machinery.
2. D. Angluin and D. Chen. Learning a Random DFA from Uniform Strings and State Information. In *Proceedings of the 26th International Conference on Algorithmic Learning Theory*, ALT'15, pages 119–133. Springer International Publishing, 2015.
3. D. Angluin and M. Kharitonov. When Won't Membership Queries Help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995.
4. B. Applebaum, A. Bogdanov, and A. Rosen. A Dichotomy for Local Small-Bias Generators. In Ronald Cramer, editor, *Theory of Cryptography*, pages 600–617. Springer Berlin Heidelberg, 2012.
5. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006.
6. B. Applebaum, Y. Ishai, and E. Kushilevitz. On Pseudorandom Generators with Linear Stretch in NC0. *Comput. Complex.*, 17(1):38–69, April 2008.
7. B. Applebaum and E. Kachlon. Sampling Graphs without Forbidden Subgraphs and Unbalanced Expanders with Negligible Error. In *IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS19)*, pages 171–179, 2019.
8. B. Applebaum and S. Lovett. Algebraic Attacks against Random Local Functions and Their Countermeasures. *SIAM Journal on Computing*, 47(1):52–79, 2018.
9. Benny Applebaum. Pseudorandom Generators with Long Stretch and Low Locality from Random Local One-Way Functions. *SIAM J. Comput.*, 42(5):2008–2037, 2013.
10. Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 171–180. ACM, 2010. doi:10.1145/1806689.1806715.

11 Benny Applebaum and Pavel Raykov. Fast Pseudorandom Functions Based on Expander Graphs. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 27–56, 2016.

12 V. Arvind, J. Köbler, and W. Lindner. Parameterized Learnability of Juntas. *Theor. Comput. Sci.*, 410(47-49):4928–4936, November 2009.

13 I. Ben-Eliezer, R. Hod, and S. Lovett. Random low-degree polynomials are hard to approximate. *Comput. Complex.*, 21(1):63–81, 2012. doi:10.1007/s00037-011-0020-6.

14 A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, pages 278–291, 1994.

15 A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Occam's Razor. *Inf. Process. Lett.*, 24(6):377–380, April 1987.

16 A. Bogdanov and Y. Qiao. On the Security of Goldreich's One-Way Function. *Comput. Complex.*, 21(1):83–127, March 2012.

17 E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl. Low-Complexity Weak Pseudorandom Functions in AC0[MOD2]. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021*, volume 12828 of *Lecture Notes in Computer Science*, pages 487–516. Springer, 2021.

18 M. Carmosino, R. Impagliazzo, V. Kabanets, and A. Kolokolova. Learning Algorithms from Natural Proofs. In *Proceedings of the 31st Conference on Computational Complexity*, CCC'16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

19 M. Carmosino, R. Impagliazzo, V. Kabanets, and A. Kolokolova. Agnostic Learning from Tolerant Natural Proofs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, volume 81 of *LIPIcs*, pages 35:1–35:19, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

20 J. Cook, O. Etesami, R. Miller, and L. Trevisan. Goldreich's One-Way Function Candidate and Myopic Backtracking Algorithms. In Omer Reingold, editor, *Theory of Cryptography*, pages 521–538, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

21 G. Couteau, A. Dupin, P. Méaux, M. Rossi, and Y. Rotella. On the Concrete Security of Goldreich's Pseudorandom Generator. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 96–124. Springer, 2018.

22 M. Cryan and P. B. Miltersen. On Pseudorandom Generators in NC^0. In *Mathematical Foundations of Computer Science 2001, 26th International Symposium, MFCS 2001 Marianske Lazne, Czech Republic, August 27-31, 2001, Proceedings*, volume 2136 of *Lecture Notes in Computer Science*, pages 272–284. Springer, 2001.

23 A. Daniely. Complexity Theoretic Limitations on Learning Halfspaces. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC'16, pages 105–117, New York, NY, USA, 2016. ACM.

24 A. Daniely and S. Shalev-Shwartz. Complexity Theoretic Limitations on Learning DNF's. In *Proceedings of 29th Conference on Learning Theory*, volume 49 of *COLT'16*, pages 815–830, Columbia University, New York, USA, 23–26 June 2016. PMLR.

25 A. Daniely and G. Vardi. From Local Pseudorandom Generators to Hardness of Learning. In *Conference on Learning Theory, COLT 2021, 15-19 August 2021, Boulder, Colorado, USA*, volume 134 of *Proceedings of Machine Learning Research*, pages 1358–1394. PMLR, 2021.

26 V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 563–574, October 2006.

27 Yuval Filmus. Junta threshold for low degree boolean functions on the slice. *CoRR*, abs/2203.04760, 2022. doi:10.48550/arXiv.2203.04760.

28 Yuval Filmus and Ferdinand Ihringer. Boolean constant degree functions on the slice are juntas. *Discret. Math.*, 342(12), 2019. doi:10.1016/j.disc.2019.111614.

29 O. Goldreich. *Foundations of Cryptography: Volume 1*. Cambridge University Press, New York, NY, USA, 2006.

30 O Goldreich. *Candidate One-Way Functions Based on Expander Graphs*, pages 76–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

31 O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *J. ACM*, 33(4):792–807, August 1986.

32 O. Goldreich and L. A. Levin. A Hard-Core Predicate for All One-Way Functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 25–32, New York, NY, USA, 1989. Association for Computing Machinery.

33 R. Impagliazzo and L. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, FOCS'90, pages 812–821, 1990.

34 Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography with Constant Computational Overhead. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 433–442, New York, NY, USA, 2008. Association for Computing Machinery.

35 J. Jackson, H. Lee, R. Servedio, and A. Wan. Learning random monotone DNF. *Discrete Applied Mathematics*, 159(5):259–271, 2011.

36 J. Jackson and R. Servedio. Learning Random Log-Depth Decision Trees under Uniform Distribution. *SIAM Journal on Computing*, 34(5):1107–1128, 2005.

37 A. Jain, H. Lin, and A. Sahai. Indistinguishability Obfuscation from Well-Founded Assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 60–73, New York, NY, USA, 2021. Association for Computing Machinery.

38 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability Obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part I*, volume 13275 of *Lecture Notes in Computer Science*, pages 670–699. Springer, 2022.

39 M. Kearns and L. G. Valiant. Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 433–444, New York, NY, USA, 1989. Association for Computing Machinery.

40 M. Kharitonov. Cryptographic Hardness of Distribution-Specific Learning. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 372–381, New York, NY, USA, 1993. Association for Computing Machinery.

41 E. Kushilevitz and Y. Mansour. Learning Decision Trees Using the Fourier Spectrum. *SIAM J. Comput.*, 22(6):1331–1348, December 1993.

42 N. Linial, Y. Mansour, and N. Nisan. Constant Depth Circuits, Fourier Transform, and Learnability. *J. ACM*, 40(3):607–620, July 1993.

43 Yanyi Liu and Rafael Pass. On the possibility of basing cryptography on exp≠bpp. In *Advances in Cryptology – CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I*, pages 11–40, Berlin, Heidelberg, 2021. Springer-Verlag. doi:10.1007/978-3-030-84242-0_2.

44 M. Nanashima. Extending Learnability to Auxiliary-Input Cryptographic Primitives and Meta-PAC Learning. In *Proceedings of the 33rd Conference on Learning Theory, COLT'20*, volume 125, pages 2998–3029. PMLR, 09–12 July 2020.

45 M. Nanashima. A Theory of Heuristic Learnability. In *Proceedings of the 34th Conference on Learning Theory, COLT'21*. PMLR, 2021.

46 Moni Naor and Omer Reingold. Synthesizers and Their Application to the Parallel Construction of Pseudo-Random Functions. *J. Comput. Syst. Sci.*, 58(2):336–375, 1999.

47 Moni Naor and Guy N. Rothblum. Learning to impersonate. In William W. Cohen and Andrew W. Moore, editors, *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 649–656. ACM, 2006.

48 R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, New York, NY, USA, 2014.

49 R. O'Donnell and D. Witmer. Goldreich's PRG: Evidence for Near-Optimal Polynomial Stretch. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 1–12, 2014.

50 I. Oliveira and R. Santhanam. Conspiracies between Learning Algorithms, Circuit Lower Bounds, and Pseudorandomness. In *Proceedings of the 32nd Computational Complexity Conference*, CCC'17, Dagstuhl, DEU, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

51 Igor Carboni Oliveira, Rahul Santhanam, and Roei Tell. Expander-Based Cryptography Meets Natural Proofs. *Comput. Complex.*, 31(1):4, 2022.

52 L. Pitt and L. Valiant. Computational Limitations on Learning from Examples. *J. ACM*, 35(4):965–984, October 1988.

53 O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *J. ACM*, 56(6), September 2009.

54 H. Ren and R. Santhanam. Hardness of KT Characterizes Parallel Cryptography. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:58, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2021.35.

55 R. Santhanam. Pseudorandomness and the Minimum Circuit Size Problem. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*, volume 151 of *LIPIcs*, pages 68:1–68:26, 2020.

56 L. Sellie. Learning Random Monotone DNF Under the Uniform Distribution. In *Proceedings of the 21st Annual Conference on Learning Theory*, COLT'08, pages 181–192. Omnipress, 2008.

57 L. Sellie. Exact Learning of Random DNF over the Uniform Distribution. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC'09, pages 45–54, New York, NY, USA, 2009. ACM.

58 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

59 S. Vadhan. On learning vs. refutation. In *Proceedings of the 2017 Conference on Learning Theory (COLT'17)*, volume 65 of *Proceedings of Machine Learning Research*, pages 1835–1848, Amsterdam, Netherlands, 07–10 July 2017. PMLR.

60 L. Valiant. A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142, 1984. doi:10.1145/1968.1972.

61 A. Yao. Theory and Application of Trapdoor Functions. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, FOCS'82, pages 80–91, November 1982.