Electronic Colloquium on Computational Complexity, Report No. 164 (2022)

# Learning versus Pseudorandom Generators in Constant Parallel Time

Shuichi Hirahara[*] Mikito Nanashima[†]

November 20, 2022

**Abstract**

A polynomial-stretch pseudorandom generator (PPRG) in $\mathsf{NC}^0$ (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 $\mathsf{NC}^0$ by the existence of logspace-computable one-way functions, but characterizing PPRGs in $\mathsf{NC}^0$ seems out of reach at present. Therefore, it is natural to ask which sort of hardness notion is *essential* for constructing PPRGs in $\mathsf{NC}^0$. Particularly, to the best of our knowledge, all the previously known candidates for PPRGs in $\mathsf{NC}^0$ follow only one framework based on Goldreich's one-way function.

In this paper, we present a new learning-theoretic characterization for PPRGs in $\mathsf{NC}^0$ 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 $\mathsf{NC}^0$.
- The average-case hardness of learning sparse $\mathbb{F}_2$-polynomials on a sparse example distribution and an $\mathsf{NC}^0$-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 $\mathsf{NC}^0$-samplable target distribution.
- The average-case hardness of learning constant-depth parity decision trees on a sparse example distribution and an $\mathsf{NC}^0$-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 $\mathsf{NC}^0$ and related classes under a hardness assumption on a natural learning problem. An important property of PPRGs in $\mathsf{NC}^0$ 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.

[*] National Institute of Informatics, Japan. `s_hirahara@nii.ac.jp`

[†] Tokyo Institute of Technology, Japan. `nanashima.m.aa@is.c.titech.ac.jp`

ISSN 1433-8092

# 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 [Val84], 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 $\mathsf{AC}^0$ under several cryptographic or deeply related assumptions [KV89; Kha93; AK95; AR16; Dan16; DS16; CIKK16; Vad17; DV21]. The opposite implication from hardness of learning to cryptography is relatively less understood and first studied by Impagliazzo and Levin [IL90] and Blum, Furst, Kearns, and Lipton [BFKL94]. Particularly, Blum, Furst, Kearns, and Lipton [BFKL94] 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 [NR06; OS17; San20; Nan20; Nan21].

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 [BFKL94]. 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 [ABGKR14; BCGIKS21]. 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., KV89; Kha93; AR16; DV21].

In this work, we study a dichotomy between learning and polynomial-stretch pseudorandom generators (PPRGs) computable in constant-depth circuits (i.e., $\mathsf{NC}^0$), 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 $\mathsf{NC}^0$ is one of the most studied primitives in parallel cryptography [cf. CM01; App13] because of its remarkable applications, such as highly efficient cryptography [IKOS08] and a recent breakthrough on indistinguishability obfuscation ($i\mathcal{O}$) based on well-founded assumptions [JLS21; JLS22]. Despite its importance, to the best of our knowledge, the only known framework for constructing PPRGs in $\mathsf{NC}^0$ is one based on Goldreich's OWF [Gol11]. For example, the celebrated work by Applebaum, Ishai, and Kushilevitz [AIK06] on randomized encodings only yields the characterization of *sublinear-stretch* PRGs in $\mathsf{NC}^0$, but characterizing PPRGs in $\mathsf{NC}^0$ seems out of reach at present. Therefore, it is natural to inquire into a new candidate for PPRGs in $\mathsf{NC}^0$ 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 $\mathsf{NC}^0$) selected in the preprocessing [cf. Gol06, Section 2.4.2]. This relaxed setting is standard, especially when we discuss a PPRG in $\mathsf{NC}^0$ [cf. App13, 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 $\mathsf{NC}^0$. Applebaum, Barak, and Wigderson [ABW10] proved the hardness of learning $O(\log n)$-junta functions under the existence of PPRGs in $\mathsf{NC}^0$ with an additional assumption on input-output connections. Applebaum and Raykov [AR16] and Daniely and Vardi [DV21] proved the hardness of learning for central classes such as depth-3 $\mathsf{AC}^0$ 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 [OST22] 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 [CIKK16; CIKK17].

Since the equivalence between pseudorandomness and unpredictability follows from the well-known result by Yao [Yao82], a reader might expect a correspondence between PPRG in $\mathsf{NC}^0$ and hardness of learning $\mathsf{NC}^0$. However, this intuition seems incorrect because while a PPRG in $\mathsf{NC}^0$ is conjectured to exist, learning $\mathsf{NC}^0$ (i.e., functions with constant locality) is trivially feasible by applying Occam's razor [BEHW87]. In this sense, there seems to exist a gap between pseudorandomness and hardness of learning when we consider considerably low complexity classes such as $\mathsf{NC}^0$. Nevertheless, can we obtain some learning-theoretic characterization for a collection of PPRG in $\mathsf{NC}^0$? 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 Section 3.2.

We consider a distribution-specific average-case learning model, introduced by Blum, Furst, Kearns, and Lipton [BFKL94]. 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 $\gamma$ 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. AKL09]. 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 $k_1, \dots, k_c$ on a concept class $\mathscr{C}$ and classes of example distributions and target distributions, we say that $\mathscr{C}$ is learnable with $(k_1, \dots, k_c)$-FPT samples if $\mathscr{C}$ is learnable with $f(k_1, \dots, k_c) \cdot n^{\Theta(1)}$ 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 [AKL09], but feasible with FPT samples (with $k2^k \cdot n^{\Theta(1)}$ samples and in $O(n^k)$ time) by Occam's razor [BEHW87]. 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. SB14].[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 $\mathsf{NC}^0$ is characterized by the learnability of various central classes with FPT samples with respect to a sparse example distribution and an $\mathsf{NC}^0$-samplable target distribution.

**Theorem 1** (informal)**.** *The following are equivalent:*

1. *There exists a collection of (infinitely-often secure[2]) PPRGs in $\mathsf{NC}^0$.*

2. *$c$-sparse $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with respect to a target distribution samplable by a depth-$d$ $\mathsf{NC}^0$ 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-$d$ $\mathsf{NC}^0$ 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-$d'$ $\mathsf{NC}^0$ circuit and a samplable distribution on $c$-sparse example distributions with $(d, c, d')$-FPT samples.*

Informally, Theorem 1 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 1 are natural.

**Concept classes.** For the formal descriptions of each parameterized concept class, see Section 3.1. 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. ODo14]. The fourth item above concerns the extensions of decision trees, containing the well-studied class of parity decision trees[3] [e.g.,

[1] We can also 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 (see Lemma 6).

[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 (see also Remark 1).

[3] In fact, the equivalence between constant-depth parity decision trees and constant-Fourier-sparse functions follows from the work by Kushilevitz and Mansour [KM93]. However, it is unclear whether these learning settings are equivalent when we restrict the target distributions to $\mathsf{NC}^0$-samplable because the transformation between these representations may be infeasible in $\mathsf{NC}^0$.

KM93]. 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 $\mathsf{NC}^0$.

**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 Definition 7 for the formal description). Note that this captures more general settings of learning than the previous distribution-specific setting in [BFKL94]; 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 $\mathsf{NC}^0$-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 $\mathsf{NC}^0$-samplable [JS05; Sel08; Sel09; JLSW11; AC15].

We also remark that Theorem 1 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 [JLS21; JLS22]. Particularly, Jain, Lin, and Sahai [JLS21] assumed (i) the hardness of learning problems LWE and LPN, (ii) the existence of a collection of PPRGs in $\mathsf{NC}^0$, 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 1.

**On removing sparsity conditions.** Although Theorem 1 shows one characterization of a collection of PPRGs in $\mathsf{NC}^0$ 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 1 still characterizes a collection of PPRGs in superclasses of $\mathsf{NC}^0$.

**Theorem 2** (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 $\mathsf{NC}^0$ 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 $\oplus$-$\mathsf{NC}^0$), then a single PPRG in $\oplus$-$\mathsf{NC}^0$ is characterized by the hardness of learning constant-degree $\mathbb{F}_2$-polynomials on the *uniform* example distribution.

**Theorem 3** (informal)**.** *For any polynomial $r(n)$, the following are equivalent:*

1. *There exists a PPRG in $\oplus$-$\mathsf{NC}^0$.*

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 (see Section 5.1). 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 [NR99]. Thus, Theorem 3 can also be regarded as the equivalence between PPRG and WPRF within the class $\oplus$-$\mathsf{NC}^0$.

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 3 is extended to a collection of PPRGs in $\oplus$-$\mathsf{NC}^0$. In other words, we can characterize the difference between a single PPRG and a collection of PPRGs in $\oplus$-$\mathsf{NC}^0$ by the difference in the generality of example distributions on the hardness of learning.

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

1. *There exists a collection of PPRGs in $\oplus$-$\mathsf{NC}^0$.*

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 2–4 also hold in super-polynomial regimes with polynomial security loss.

Theorems 1–4 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

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

Theorems 1–4 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 2** (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 1, if $d$-LRPDT is hard with FPT samples on some parametrized sparse example distribution, then a collection of PPRGs exists in $\mathsf{NC}^0$. 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 1.** *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 $\mathsf{NC}^0$ exists.*
- *If $D$ is the uniform distribution, then a PPRG in $\oplus$-$\mathsf{NC}^0$ exists.*
- *If $D$ is samplable, then a collection of PPRGs in $\oplus$-$\mathsf{NC}^0$ exists.*

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

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 2.** *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 $\mathsf{NC}^0$ 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 [GL89; KM93]. However, it is

[5] For the requirement for the advantage of learning, see Section 6.

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 [JS05], but it is unclear whether this algorithm can be extended to the case of random parity decision trees. From Corollary 1, 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 1 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 2.

## 1.3 Related Work

Applebaum, Barak, and Wigderson [ABW10] proved the hardness of learning $O(\log n)$-junta functions under the existence of PRGs in $\mathsf{NC}^0$ with an additional assumption that (roughly speaking) a small subset of output bits can be embedded indistinguishably with good local expansion. Applebaum and Raykov [AR16] proved the hardness of learning depth-3 $\mathsf{AC}^0$ circuits under the assumption related to polynomial-stretch Goldreich's PRGs, which matches the unconditional upper bound presented in [LMN93]. 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 [App13; AR16], where they essentially use the structures of Goldreich's OWFs. Daniely and Vardi [DV21] 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 $\mathsf{NC}^0$) 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 [DV21], while we can also obtain the opposite direction from the hardness of learning to cryptography. The result of [OST22] on natural properties also differs in the learning setting, particularly natural properties essentially correspond to learning with membership queries on the uniform example distribution [CIKK16].

Blum, Furst, Kearns, and Lipton [BFKL94] 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 [NR99], and we apply the same technique to show one direction in Theorem 3. 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 $\mathsf{NC}^0$ because if a concept class has the evaluation performed in $\mathsf{NC}^0$, then such a class is trivially learnable. The followup studies [NR06; OS17; San20; Nan20; Nan21] discussed relationships between cryptography and hardness of learning in $\mathsf{P}$ and $\mathsf{P/poly}$. Other studies [e.g., Reg09] developed various cryptographic schemes based on the hardness of learning linear functions with noise, but it is not clear whether PPRGs in $\mathsf{NC}^0$ 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 [FGKP06].

With regard to parallel cryptography, the constructions of PRGs in $\mathsf{NC}^0$ were presented by Applebaum, Ishai, and Kushilevitz [AIK06] (sublinear-stretch) and Applebaum, Ishai, and Kushilevitz [AIK08] (linear-stretch). Recently, Ren and Santhanam [RS21] and Liu and Pass [LP21] characterized the existence of OWF in $\mathsf{NC}^0$ based on the average-case meta-complexity notion, which only yields sublinear-stretch PRGs in $\mathsf{NC}^0$, and PPRGs in $\mathsf{NC}^0$ seem out of reach at present. Some candidates for a collection of PPRGs in $\mathsf{NC}^0$ were studied by Cook, Etesami, Miller, and Trevisan

[CEMT09], Bogdanov and Qiao [BQ12], Applebaum, Bogdanov, and Rosen [ABR12], Applebaum [App13], O'Donnell and Witmer [OW14], Applebaum and Lovett [AL18], and Couteau, Dupin, Méaux, Rossi, and Rotella [CDMRR18] based on the framework of Goldreich's OWF [Gol11]. 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 [App13].[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 [FI19] and Filmus [Fil22] 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 1 and 2

The key notion to show Theorems 1 and 2 is the dualization of concept classes, which was explicitly discussed independently by Applebaum, Barak, and Wigderson [ABW10] and Vadhan [Vad17] and applied (implicitly or explicitly) in recent studies on the hardness of learning [DS16; Dan16; Nan20; Nan21; DV21]. 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 $y = G(x)$ 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 [Yao82], 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 $\mathscr{C} \circ \mathscr{C}'$.

[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. PV88].

At a high level, we will use the argument above to show Theorems 1 and 2. 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, $\mathsf{NC}^0$ cannot be dualized. Intuitively, for an $\mathsf{NC}^0$-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 Appendix A, we formally show the impossibility of the dualization of $\mathsf{NC}^0$ 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 $\mathsf{NC}^0$ by applying the original generator constant times based on the GGM construction [GGM86]. 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 3** (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 $f \in \mathscr{C}^k_n$, 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 $\mathsf{NC}^0$ and focusing on its substructure, i.e., the correspondence between $\mathsf{NC}^0$ 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 $\mathsf{NC}^0$ 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 4** (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 $wt(f^*) \le h(k)$.*

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 $wt(x) \le c$, it holds that $x^{*(k)}$ is $g(c,k)$-junta.*

### Meta-Theorem

The proof of Theorem 1 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 $\mathsf{NC}^0$ and it satisfies the junta-sparse condition, then the existence of a collection of PPRGs in $\mathsf{NC}^0$ 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 $\mathsf{NC}^0$-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 (cf. Section 4.3). 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 $\mathsf{NC}^0$ and the average-case hardness of learning $\mathscr{C}$ with sparse data based on our meta-theorem. As the second step to show Theorem 1, we solve this puzzle, i.e., demonstrate that concept classes in Theorem 1 (i.e., $c$-sparse $\mathbb{F}_2$-polynomials, $c$-Fourier-sparse functions, and depth-$d$ $\{\text{OR},\text{Mod}_m\}$-decision trees) are FPT dualizable by $\mathsf{NC}^0$-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 [GGM86] within $\mathsf{NC}^0$, 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 $\mathsf{NC}^0$, 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 $\mathsf{NC}^0$, the target distribution of the dual of the random seed $x$ is $\mathsf{NC}^0$-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* $\mathsf{NC}^0$: Suppose that learning $\mathscr{C}$ is hard on average with FPT samples. Since the target distribution is $\mathsf{NC}^0$-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 $\mathsf{NC}^0$. Technically, when we 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 [AK19].

[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.

Theorem 2 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. See Section 4.4 for the formal argument.

## 2.2 Proof Techniques for Theorem 3

Theorem 3 shows the equivalence between the existence of a (single) PPRG in $\oplus\text{-}\mathsf{NC}^0$ 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, $\oplus\text{-}\mathsf{NC}^0$ 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 $\oplus\text{-}\mathsf{NC}^0$ circuit that first computes monomials in parallel and takes the summation of them by the top-most XOR gate, and (ii) any $\oplus\text{-}\mathsf{NC}^0$ 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 3. 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 [BHL12]. 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 [Yao82], 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 [NR99]. 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 3. Let $R = n + r(n)$. Then, $G$ selects $R^2$ examples $x^{(1)}, \dots, x^{(R^2)}$ and $R^2$ target functions $f^{(1)}, \dots, f^{(R^2)}$ 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 $i, j \in \{1, \dots, R^2\}$ as a pseudorandom string. We can prove the pseudorandomness of $G$ using the hybrid argument and the equivalence between unpredictability and pseudorandomness [Yao82]. 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. For details, see Section 5.1.

## 2.3 Proof Ideas for Theorem 4

Theorem 4 shows the equivalence of a collection of PPRGs in $\oplus\text{-}\mathsf{NC}^0$ 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 $c$. We will present the details in Section 5.2.

### Organization of The Paper

The remainder of this paper is organized as follows. In Section 3, we present preliminaries for formal arguments. In Section 4, we introduce FPT dualization and show Theorems 1 and 2 by proving the meta-theorem. In Section 5, we present the formal proofs of Theorems 3 and 4. In Section 6, we verify Corollary 1. In Appendix A, we show the impossibility of dualization of $\mathsf{NC}^0$.

## 3 Preliminaries

For each $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. For any $x \in \{0,1\}^*$, let $wt(x)$ denote the Hamming weight of $x$. For any $x \in \{0,1\}^n$ and $i \in [n]$, let $x_i$ denote the $i$-th bit of $x$. For any $x \in \{0,1\}^n$ and $i, j \in [n]$ with $i < j$, let $x_{[i,j]} = x_i \circ x_{i+1} \circ \cdots \circ x_j$. For any $f \colon \{0,1\}^n \to \{0,1\}$ and any $k \in \mathbb{N}$ with $k \leq n$, we say that $f$ is $k$-junta if $f$ depends on only at most $k$ out of $n$ coordinates in the input. We say that a multi-output function $f \colon \{0,1\}^n \to \{0,1\}^m$ has locality $k$ if each output bit of $f$ is computed by a $k$-junta. For any $m \in \mathbb{N}$, we define a symmetric function $\mathrm{Mod}_m \colon \{0,1\}^* \to \{0,1\}$ as $\mathrm{Mod}_m(x) = 1$ iff $x \equiv 0 \mod m$.

We use the notation $\mathsf{negl}$ to represent some negligible function, i.e., for any polynomial $p$ and sufficiently large $n \in \mathbb{N}$, it holds that $\mathsf{negl}(n) < 1/p(n)$. We also use the notation $\mathsf{poly}$ to refer to some polynomial.

For a distribution $D$, we use the notation $x \leftarrow D$ to denote random sampling $x$ according to $D$. For a finite set $S$, we use the notation $x \leftarrow_u S$ to denote the uniform sampling from $S$. For each $n \in \mathbb{N}$, let $U_n$ denote the uniform distribution over $\{0,1\}^n$. In this paper, we abuse the notation for distribution to refer to a random variable selected according to the same distribution. For two distributions $D_1$ and $D_2$, we let $D_1 \equiv_s D_2$ denote that $D_1$ and $D_2$ are statistically indistinguishable, i.e., for any Boolean-valued function $f$, it holds that $|\Pr[f(D_1) = 1] - \Pr[f(D_2) = 1]| \leq \mathsf{negl}(n)$.

For any $n, d \in \mathbb{N}$ with $d \leq n$, let $\binom{n}{\leq d} = \sum_{i=0}^{d} \binom{n}{i} = O(n^d)$. We use the following lemma.

**Lemma 1** ([BHL12, Claim 2.4])**.** *For any $\beta \in (0,1)$, there exists a constant $\gamma \in (0,1)$ such that for any $m, d \in \mathbb{N}$ and for any sufficiently large $n \in \mathbb{N}$, $\binom{m}{\leq d} \leq \beta \cdot \binom{n}{\leq d}$ implies $m \leq n(1 - \gamma/d)$.*

Let $\mathscr{C} = \{\mathscr{C}_n\}_{n \in \mathbb{N}}$ be an arbitrary class of functions (i.e., a complexity class), where $\mathscr{C}_n \subseteq \{f \colon \{0,1\}^n \to \{0,1\}\}$ for each $n \in \mathbb{N}$. When we discuss the computability in $\mathscr{C}$ in this paper, we implicitly assume its uniformity, i.e., we say that a family of multi-output functions $f = \{f_n\}_{n \in \mathbb{N}}$, where $f \colon \{0,1\}^n \to \{0,1\}^{m(n)}$, is computable in $\mathscr{C}$ if there exists a polynomial-time algorithm $A$ such that for any $n \in \mathbb{N}$ and $i \in [m(n)]$, the algorithm $A(1^n, i)$ outputs the description of a function $g_i \in \mathscr{C}_n$ such that $g_i(x)$ is the $i$-th bit of $f_n(x)$ for any input $x \in \{0,1\}^n$. Let $\mathsf{NC}^0$ (resp. $\oplus$-$\mathsf{NC}^0$) be the complexity class of constant-depth circuits (resp. constant-depth circuits in which the top-most gate can be a $\oplus$-gate with unbounded fan-in).

### 3.1 Boolean Functions and Representations

In this paper, we consider a distribution on functions samplable in low complexity. In such cases, the choice of binary encodings of the functions may affect the results because the translation between two different representations may be infeasible in low complexity. Thus, we specify the binary representations for concept classes in a natural manner as follows.

#### 3.1.1 $\mathbb{F}_2$-polynomials

Any Boolean-valued function $f\colon \{0,1\}^n \to \{0,1\}$ has a unique representation as a polynomial in $\mathbb{F}_2$ obtained by expanding $f(x) = \sum_{a\in\mathbb{F}_2^n} f(a)\mathbb{1}(x = a) = \sum_{a\in\mathbb{F}_2^n} f(a)\prod_{i\in[n]}(x_i + a_i + 1)$ under operations of $\mathbb{F}_2$.

For each $S \subseteq [n]$ and $x \in \mathbb{F}_2^n$, let $x^S = \prod_{i\in S} x_i$. For each $\mathbb{F}_2$-polynomial $p\colon \mathbb{F}_2^n \to \mathbb{F}_n$ and $S \subseteq [n]$, we use the notation $p_S$ to refer to the coefficient of $p$ on $S$, i.e., $p(x) = \sum_S p_S x^S$. We define the degree of an $\mathbb{F}_2$-polynomial $p$ as the maximum number $d$ such that there exists a subset $S$ of coordinates such that $|S| = d$ and $p_S = 1$. Then, we specify the binary representation of degree-$d$ $\mathbb{F}_2$-polynomials naturally by a string of length $\binom{n}{\le d}$ concatenating all coefficients on $S$ with $|S| \le d$ in some canonical order.

The following lemma plays a key role in the proof of Theorem 3.

**Lemma 2** ([BHL12, Lemma 1.4]). *For any $n, m \in \mathbb{N}$ and any $2^m$ distinct points $x_1, \dots, x_{2^m} \in \mathbb{F}_2^n$, the following set is a linear subspace of $\mathbb{F}_2^{2^m}$ and the dimension is at least $\binom{m}{\le d}$:*

$$\left\{v^p \in \mathbb{F}_2^{2^m} : p \textit{ is a degree-}d\ \mathbb{F}_2\textit{-polynomial and } v_i^p = p(x_i) \textit{ for each } i \in [2^m]\right\}.$$

#### 3.1.2 Fourier Representations

When we consider the Fourier representation of Boolean functions, we regard any Boolean-valued function $f\colon \{0,1\}^n \to \{0,1\}$ as a function mapping from $\{0,1\}^n$ to $\{-1,1\}$ by considering $(-1)^{f(x)}$. For each $\alpha \in \{0,1\}^n$, we define a function $\chi_\alpha\colon \{0,1\}^n \to \{-1,1\}$ as $\chi_\alpha(x) = (-1)^{\langle x,\alpha\rangle}$, where $\langle\cdot,\cdot\rangle$ denotes the inner product in $\mathbb{F}_2^n$. Then, any Boolean function $f\colon \{0,1\}^n \to \{-1,1\}$ has a unique representation of the form $f(x) = \sum_{S\subseteq[n]} \widehat{f}(\alpha)\chi_\alpha(x)$, where $\widehat{f}(\alpha) = \mathrm{E}_{x\leftarrow_u\{0,1\}^n}[f(x)\chi_\alpha(x)]$ and is called a Fourier coefficient of $f$ on $\alpha$. For further background on Fourier analysis, refer to the textbook by O'Donnell [ODo14].

For any function $f\colon \{0,1\}^n \to \{-1,1\}$, the Fourier sparsity of $f$ is defined as $|\{S \subseteq [n] : \widehat{f}(S) \neq 0\}|$. For any $s \in \mathbb{N}$ and any function $f$ of Fourier sparsity $s$, each Fourier coefficient $\widehat{f}(\alpha)$ takes the form of $M_\alpha/2^{\lceil\log s\rceil}$, where $M_\alpha \in \{-2^{\lceil\log s\rceil}, \dots, 0, \dots, 2^{\lceil\log s\rceil}\}$ [GOSSW11; ODo14, Exercise 3.32]. Thus, we assume that each $n$-input function $f$ of Fourier sparsity $s$ is represented by an $O(ns\log s)$-bit string, where each term in $f$ is represented by a tuple of $\lceil\log s\rceil + 1$ bits indicating the coefficient (i.e., $M_\alpha$ above) and $n$ bits indicating the coordinates that are contained in the term (i.e., $\alpha$ above). For instance, $f(x_1,\dots,x_n) = x_1 \vee x_2$ is 4 Fourier-sparse function and represented in this form as $((-2, 0^n), (2, 10^{n-1}), (2, 010^{n-2}), (2, 110^{n-2}))$.

#### 3.1.3 Decision Trees and Extensions

A decision tree (DT) is a representation of Boolean functions and is defined as a rooted binary tree in which the internal nodes are labeled by a variable $x_i$, and the leaves are labeled by $\{0,1\}$. For an $n$-input DT $T$ and input $x \in \{0,1\}^n$, the value of $T(x)$ is determined as follows: $T$ queries the value in $x$ according to the label at the root, and if the answer is true (resp. false), then $T$ looks at the right (resp. left) subtree and repeats the same process for the subtree. $T$ repeats this until it reaches some leaf and then outputs the binary label of the reached leaf. We define the depth of DT as the maximum length of a path from the root to the leaves.

For any (family of) symmetric function $f$ (e.g., OR and $\mathrm{Mod}_m$), we define an $f$-decision tree ($f$-DT) in the same manner as above except that each internal node is labeled by the query of the form $f(x_{i_1}, \ldots, x_{i_k})$ for some $k \in [n]$ and $\{i_1, \ldots, i_k\} \subseteq [n]$ (instead of $x_i$).

Without loss of generality, we can assume that the number of internal nodes of any depth-$d$ $f$-DT is exactly $2^d - 1$ by adding dummy nodes, where nothing is queried, and a configuration automatically proceeds to the false subtree. We also assume a standard canonical ordering in $2^d - 1$ nodes (root to leaves) and $2^d$ leaves (left to right). Then, for any (family of) symmetric function $f$, we naturally specify the binary representation of $n$-input $f$-decision trees of depth $f$ as a $(2^d - 1) \cdot n + 2^d$-bit string consisting of $2^d - 1$ strings in $\{0,1\}^n$ that represent the sets of relevant coordinates in $[n]$ for $2^d - 1$ internal nodes and $2^d$ binary labels on leaves.

### 3.2 Learning

We define a concept class as a subset of Boolean-valued functions. For any concept class $\mathscr{C}$ and $n \in \mathbb{N}$, we use the notation $\mathscr{C}_n$ to represent a subset of $\mathscr{C}$ restricted to the input size $n$, i.e., $\mathscr{C}_n = \mathscr{C} \cap \{f \colon \{0,1\}^n \to \{0,1\}\}$. We also define a parameterized class $\mathscr{C} = \{\mathscr{C}^k\}_{k \in \mathbb{N}}$ as a family of concept classes such that $\mathscr{C}^k \subseteq \mathscr{C}^{k+1}$ for each $k \in \mathbb{N}$. Note that we often use a subscript and a superscript to refer to input size and a parameter, respectively.

Following the study by Blum, Furst, Kearns, and Lipton [BFKL94], we mainly discuss the average-case learnability based on the following prediction model. Note that the prediction model has the same capability as the standard PAC learning model with no assumption on the hypothesis class [HKLW88]. For convenience, we regard the time bound of a learning algorithm as a function in the input length of a target function (i.e., the example size) instead of a function in the input length of learning algorithms.

**Definition 5** (average-case learning)**.** *Let $\mathscr{C}$ be a concept class. Let $D = \{D_n\}_{n \in \mathbb{N}}$ and $F = \{F_n\}_{n \in \mathbb{N}}$ be families of distributions, where $D_n$ is a distribution on $\{0,1\}^n$ and $F_n$ is a distribution on $\mathscr{C}_n$. For any functions $t, m \colon \mathbb{N} \to \mathbb{N}$ and $\gamma \colon \mathbb{N} \to (0, 1/2)$, we say that $\mathscr{C}$ is $(t, m, \gamma)$-learnable on average with respect to $D$ and $F$ if there exists a randomized algorithm $L$ such that for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr_{L, f, x^{(1)}, \ldots, x^{(m(n))}, x_c}\left[L((x^{(1)}, f(x^{(1)})), \ldots, (x^{(m(n))}, f(x^{(m(n))})), x_c) \text{ outputs } f(x_c) \text{ in time } t(n)\right] \geq \frac{1}{2} + \gamma(n),$$

*where $x^{(1)}, \ldots, x^{(m(n))}, x_c \leftarrow D_n$ and $f \leftarrow F_n$.*

*We refer to $D$ (resp. $F$) as an example (resp. a target) distribution. We also refer to $f$, $x_c$, and $\gamma$ above as a target function, a challenge, and an advantage, respectively.*

Without loss of generality, we ignore the cases in which $m(n) > t(n)$. Next, we define the key notion of this work, i.e., FPT sample complexity.

**Definition 6** (FPT samples)**.** *For $c \in \mathbb{N}$, let $k_1, \ldots, k_c$ be parameters on a concept class $\mathscr{C}$ and classes of example distributions and target distributions. For any functions $t \colon \mathbb{N} \to \mathbb{N}$ and $\gamma \colon \mathbb{N} \to (0, 1/2)$, we say that $\mathscr{C}$ is $(t, \gamma)$-learnable on average with $(k_1, \ldots, k_c)$-FPT samples if there exists a function $m(n, k_1, \ldots, k_c) = f(k_1, \ldots, k_c) \cdot n^{O(1)}$ for some $f \colon \mathbb{N}^c \to \mathbb{N}$ such that for any choice of $k_1, \ldots, k_c \in \mathbb{N}$ and any choice of an example distribution $D$ and target distribution $F$ that satisfy the settings of the parameters, $\mathscr{C}$ is $(t, m_{k_1, \ldots, k_c}, \gamma)$-learnable on average with respect to $D$ and $F$, where $m_{k_1, \ldots, k_c}(n) := m(n, k_1, \ldots, k_c)$.*

We define distributions on example distributions as example distributions samplable with shared randomness to introduce the average-case variant of distribution specifc learning.

**Definition 7** (samplable with shared randomness)**.** *We say that an example distribution is samplable with shared randomness if there exists a polynomial-time sampling algorithm $S$ such that for any example size $n \in \mathbb{N}$, examples in $\{0,1\}^n$ are selected identically and independently according to $S(U_{\mathsf{poly}(n)}; r)$, where $r$ is an auxiliary random string selected uniformly at random from $\{0,1\}^{\mathsf{poly}(n)}$ at the initiation and shared through sampling processes.*

Note that learning on example distribution samplable with shared randomness is the notion sandwiched between distribution-free learning and distribution-specific learning in the following sense. Any distribution-free learner that succeeds on all (unknown) example distributions also succeeds on any example distribution samplable with shared randomness regardless of the choice of shared randomness. In addition, if there exists a learner that succeeds on $D$ for each example distribution $D$ samplable with shared randomness, then there exists a distribution-specific learner that succeeds on $D'$ for each samplable example distribution $D'$.

We also discuss another formulation of learning, which was introduced explicitly by Vadhan [Vad17].

**Definition 8** (RRHS-refutation)**.** *Let $\mathscr{C}$ be a concept class, $D$ be an example distribution, and $F$ be a target distribution on $\mathscr{C}$. For functions $t, m\colon \{0,1\}^n \to \mathbb{N}$ and $\gamma\colon \mathbb{N} \to (0,1/2)$, we say that $\mathscr{C}$ is $(t,m,\gamma)$-random-right-hand-side-refutable (RRHS-refutable) on average with respect to $D$ and $F$ if there exists a randomized $t(n)$-time algorithm $A$ such that for any $n \in \mathbb{N}$,*

$$\Pr_{A,x,f}\left[A((x^{(1)}, f(x^{(1)})), \ldots, (x^{(m(n))}, f(x^{(m(n))}))) = 1\right] - \Pr_{A,x,b}\left[A((x^{(1)}, b^{(1)}), \ldots, (x^{(m(n))}, b^{(m(n))})) = 1\right] \geq \gamma(n),$$

*where $f \leftarrow F_n$, $x^{(i)} \leftarrow D_n$, and $b^{(i)} \leftarrow_u \{0,1\}$ for each $i \in [m(n)]$.*

Vadhan [Vad17] observed that RRHS-refutability is equivalent to learnability. In this work, we use one direction from the hardness of learning to the hardness of RRHS-refuting, which follows from Yao's next-bit generator [Yao82].

**Theorem 5** ([Yao82; Vad17])**.** *Let $m\colon \mathbb{N} \to \mathbb{N}$ and $\gamma\colon \mathbb{N} \to (0,1/2)$ be any polynomial-time computable functions. Let $D$ be an arbitrary example distribution and $F$ be an arbitrary target distribution on a concept class $\mathscr{C}$. Then, there exists a polynomial $q$ such that for any time-bound function $t(n)$, if $\mathscr{C}$ is not $(t(n), m(n), \gamma(n))$-learnable on average with respect to $D$ and $F$, then $\mathscr{C}$ is not $(t(n)/q(n), m(n), m(n)\gamma(n))$-RRHS-refutable on average with respect to $D$ and $F$.*

We introduce the following useful fact, which follows from the XOR lemma. For the formal argument, see the work by Nanashima [Nan21].

**Fact 1.** *For any polynomial $m^{\oplus}, p$, there exist polynomials $m, \ell, q$ and a randomized algorithm* **Boost** *such that for any example distribution $D_{ex}$ and any samplable target distribution $D_{targ}$ on a concept class $\mathscr{C}$, the following hold.*

- *$\ell$ is determined by only $p$ (i.e., independent of $m^{\oplus}$).*
- **Boost** *is given $m(n)$ samples and a challenge according to $D_{ex}$ and $D_{targ}$ with a description of a randomized algorithm $L^{\oplus}$ and outputs a prediction for the challenge.*

- *We define a concept class $\mathscr{C}^{\oplus}$ by $\mathscr{C}_n^{\oplus} = \mathcal{C}^{\oplus}_{n'\ell(n')}$, where $n'$ is the maximum integer satisfying $n'\ell(n') \leq n$ and*

$$\mathcal{C}^{\oplus}_{n\ell(n)} = \left\{ f(x^{(1)} \circ \cdots \circ x^{(\ell(n))}) := \bigoplus_{i,j \in [\ell(n)]} f^{(i)}(x^{(j)}) \,\middle|\, f^{(i)} \in \mathscr{C}_n, x^{(j)} \in \{0,1\}^n \right\}.$$

*Let $D^{\oplus}_{ex}$ and $D^{\oplus}_{targ}$ be families of distributions, where $(D^{\oplus}_{ex})_n$ and $(D^{\oplus}_{targ})_n$ are the distributions of $x^{\oplus} := x^{(1)} \circ \cdots \circ x^{(\ell(n'))}$ and $f^{\oplus}(x^{\oplus}) := \bigoplus_{i,j} f^{(i)}(x^{(j)})$ for $x^{(1)}, \ldots, x^{(\ell(n'))} \leftarrow (D_{ex})_{n'}$ and $f^{(1)}, \ldots, f^{(\ell(n'))} \leftarrow (D_{targ})_{n'}$, respectively (where $n'$ is the maximum integer satisfying $n'\ell(n') \leq n$). If $L^{\oplus}$ $(t(n), m^{\oplus}(n), 1/p^{\oplus}(n))$-learns $\mathscr{C}^{\oplus}$ on average with respect $D^{\oplus}_{ex}$ and $D^{\oplus}_{targ}$ for some polynomial $p^{\oplus}$, then* **Boost** *$(t(n\ell(n))q(n), m(n), 1/2 - 1/p(n))$-learns $\mathscr{C}$ on average with respect to $D_{ex}$ and $D_{targ}$ for any sufficiently large $n$.*

### 3.3 Pseudorandom Generator

We define a pseudorandom generator that stretches a short random seed to a long pseudorandom string indistinguishable from a random string by time-bounded adversaries. For convenience, in this paper, we regard the time-bound of adversaries as a function in the input length of a generator (i.e., the length of the hidden random seed) instead of a function in the input length of adversaries.

**Definition 9** (pseudorandom generator)**.** *Let $t \colon \mathbb{N} \to \mathbb{N}$ be any time-bound function. We say that a family $G = \{G_n\}_{n \in \mathbb{N}}$, where $G_n \colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ for some function $\ell \colon \mathbb{N} \to \mathbb{N}$, is an (infinitely often) pseudorandom generator (PRG) against $t(n)$-time adversaries if $\ell(n) > n$ and for any randomized $t(n)$-time algorithm $A$, there exist infinitely many $n \in \mathbb{N}$ such that*

$$\left| \Pr_{A,U_n}[A(1^n, G_n(U_n)) = 1] - \Pr_{A,U_{\ell(n)}}[A(1^n, U_{\ell(n)}) = 1] \right| \leq \mathsf{negl}(n).$$

*In addition, we say that a PRG $G$ is a polynomial-stretch PRG (PPRG) if $\ell(n) > n^{1+\epsilon}$ holds for some constant $\epsilon > 0$.*

*For any polynomial $p(n)$, we say that $G$ is a weak PRG with indistinguishable parameter $1/p(n)$ against $t(n)$-time adversaries if $\ell(n) > n$ and for any randomized $t(n)$-time algorithm $A$, there exist infinitely many $n \in \mathbb{N}$ such that*

$$\left| \Pr[A(1^n, G_n(U_n)) = 1] - \Pr[A(1^n, U_{\ell(n)}) = 1] \right| \leq 1/p(n).$$

We usually omit the subscript $n$ from the notation above. Instead, we use the notation $G_i$ for $i \in [n]$ to refer to the function computing the $i$-th bit of $G$. We also often omit $1^n$ from the input to adversaries.

Note that any generator in $\mathsf{NC}^0$ has a constant locality because any depth-$d$ circuit only depends on at most $2^d$ coordinates of the input.

**Remark 1.** *In this paper, we mainly discuss the equivalence between learnability for all input sizes as in Section 3.2 and PPRGs with infinitely often security as above. However, our results are easily extended to the equivalence between learnability for infinitely many input sizes and PPRGs with sufficiently large security based on the following observation. In reductions from adversaries to learners (resp. from learners to adversaries) we discuss in this paper, it is not hard to verify that each seed length (resp. input size) $n$ is mapped some distinct interval $I_n$ on input sizes (resp. seed lengths) of size $\mathsf{poly}(n)$ such that $\cup_{n \in \mathbb{N}} I_n = \mathbb{N}$.*

We also extend the definition above to a collection of pseudorandom generators.

**Definition 10** (a collection of PRGs)**.** *We say that a family $G = \{G_{n,z}\}_{n\in\mathbb{N}, z\in\{0,1\}^{\mathsf{poly}(n)}}$, where $G_{n,z}\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ for some function $\ell\colon \mathbb{N}\to\mathbb{N}$, is a collection of PRGs against $t(n)$-time adversaries if (i) $\ell(n) > n$, (ii) for any $(n, z)$, the binary representation[8] of $G_{n,z}$ is computable from $(1^n, z)$ in time $\mathsf{poly}(n)$, and (iii) for any randomized $t(n)$-time algorithm $A$, there exist infinitely many $n \in \mathbb{N}$ such that*

$$\left| \Pr_{z\leftarrow_u\{0,1\}^{\mathsf{poly}(n)},A,U_n}[A(G_{n,z}, G_{n,z}(U_n)) = 1] - \Pr_{z\leftarrow_u\{0,1\}^{\mathsf{poly}(n)},A,U_{\ell(n)}}[A(G_{n,z}, U_{\ell(n)}) = 1] \right| \leq \mathsf{negl}(n),$$

*where the input $G_{n,z}$ refers to the binary representation of $G_{n,z}$. Moreover, if $\ell(n) > n^{1+\epsilon}$ holds for some constant $\epsilon > 0$, then we say that $G$ is a collection of PPRGs.*

*We also define a collection of weak PRGs in the same manner as Definition 9.*

We often omit the subscripts $n$ and $z$ from the notation above and refer to a choice of $z$ as a choice of $G$.

We introduce two useful theorems shown in earlier studies. The first theorem shows the way to amplify the stretch of PPRG by applying the original PPRG repeatedly constant time.

**Theorem 6** ([GGM86])**.** *For any function $G = \{G_n\}_{n\in\mathbb{N}}$, where $G\colon \{0,1\}^n \to \{0,1\}^{n^{1+\epsilon}}$ for some constant $\epsilon > 0$, and for any constants $c \in [\lceil\epsilon^{-1}\rceil]$ and $d \in \mathbb{N}$, we define functions $G^{(0,c)} = \{G_n^{(0,c)}\}_{n\in\mathbb{N}}$ and $G^{(d)} = \{G_n^{(d)}\}_{n\in\mathbb{N}}$, where $G^{(0,c)}\colon \{0,1\}^n \to \{0,1\}^{n^{1+c\epsilon}}$ and $G^{(d)}\colon \{0,1\}^n \to \{0,1\}^{n^{d+1}}$, inductively as follows:*

$$\begin{aligned}
G^{(0,1)}(x) &= G(x)\\
G_n^{(0,c)}(x) &= G_n(G_n^{(0,c-1)}(x)_{[1,n]}) \circ G_n(G_n^{(0,c-1)}(x)_{[n+1,2n]}) \circ \cdots \circ G_n(G_n^{(0,c-1)}(x)_{[n^{1+(c-1)\epsilon}-n+1,n^{1+(c-1)\epsilon}]})\\
G^{(1)}(x) &= G_n^{(0,\lceil\epsilon^{-1}\rceil)}(x)_{[1,n^2]}\\
G_n^{(d)}(x) &= G_n^{(1)}(G_n^{(d-1)}(x)_{[1,n]}) \circ G_n^{(1)}(G_n^{(d-1)}(x)_{[n+1,2n]}) \circ \cdots \circ G_n^{(1)}(G_n^{(d-1)}(x)_{[n^d-n+1,n^d]}).
\end{aligned}$$

*For each $d \in \mathbb{N}$, there exists a polynomial $q$ such that for any time-bound function $t$, if $G$ is a PPRG against $t(n)$-time adversaries, then $G^{(d)}$ is also a PPRG against $t(n)/q(n)$-time adversaries. Furthermore, this also holds for a collection of PRGs.*

The second theorem shows that any weak PPRG with indistinguishable parameter $n^{-\Theta(1)}$ can be converted into a PPRG (with negligible indistinguishable parameter) without loss of constant locality.

**Theorem 7** ([AK19])**.** *For any constant $d \in \mathbb{N}$, $a > 0$, and $\epsilon, \epsilon' > 0$, there exist $d' \in \mathbb{N}$, a polynomial $q$, and $\delta \in (0, 1)$ such that any collection of weak PPRGs of stretch $n^{1+\epsilon}$ and indistinguishable parameter $1/n^a$ computable in depth-$d$ $\mathsf{NC}^0$ against $t(n)$-time adversaries can be converted into a collection of PPRGs of stretch $n^{1+\epsilon'}$ in depth-$d'$ $\mathsf{NC}^0$ against $t(n^\delta)/q(n)$-time adversaries.*

# 4 Learning vs. PPRGs in Constant-Parallel Time

In this section, we show Theorems 1 and 2.

[8] Specifically, when we discuss a collection of PPRGs in a class $\mathcal{C}$ of Boolean functions, this is the binary representation for $\mathcal{C}$.

**Theorem 1.** *For any $a > 1$, the following are equivalent:*

1. *There exists a collection of PPRGs in* $\mathsf{NC}^0$.
2. *$c$-sparse $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with advantage $n^{-a}$ with respect to a $c'$-sparse example distribution samplable with shared randomness a target distribution samplable by a depth-$d$ $\mathsf{NC}^0$ circuit with $(c, c', d)$-FPT samples.*
3. *$c$-Fourier-sparse functions are not polynomial-time learnable on average with advantage $n^{-a}$ with respect to a $c'$-sparse example distribution samplable with shared randomness and a target distribution samplable by a depth-$d$ $\mathsf{NC}^0$ circuit 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 advantage $n^{-a}$ with respect to a $c$-sparse example distribution samplable with shared randomness and a target distribution samplable by a depth-$d'$ $\mathsf{NC}^0$ circuit with $(d, c, d')$-FPT samples.*

**Theorem 2.** *For any $a > 1$, 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 advantage $n^{-a}$ with respect to an example distribution samplable with shared randomness and a target distribution samplable by a $c'$-sparse $\mathbb{F}_2$-polynomial 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 advantage $n^{-a}$ with respect to an example distribution samplable with shared randomness and a target distribution samplable by a $c'$-Fourier sparse functions with $(c, c')$-FPT samples.*

## 4.1 FPT Dualization and Junta-Sparse Condition

First, we review the key notions for showing Theorem 1.

**Definition 4** (FPT dualizable)**.** *Let $\mathscr{C}^k$ be a parameterized concept class. We say that $\mathscr{C}$ is 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) $g(k,x) \in \mathscr{C}_{f_1(k)\cdot p_{dual}(n)}^{f_2(k)}$, (ii) $h(k,f) \in \{0,1\}^{f_1(k)\cdot p_{dual}(n)}$, and (iii) $(g(k,x))(h(k,f)) = f(x)$.*

*Moreover, for parameterized classes $\mathscr{C}^k$ and $\mathscr{D}^\ell$, we say that $\mathscr{C}$ is FPT dualizable in $\mathscr{D}$ if (i) $\mathscr{C}$ is FPT dualizable and (ii) there exists a computable function $l : \mathbb{N} \to \mathbb{N}$ such that for any $k \in \mathbb{N}$, it holds that $g(k, \cdot)$ and $h(k, \cdot)$ are computable in $\mathscr{D}^{l(k)}$.*

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. For example, the third condition above can be rewritten as $x^*(f^*) = f(x)$ for any $f \in \mathscr{C}$ and $x \in \{0,1\}^*$.

**Definition 5** (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 $wt(f^*) \le h(k)$.*
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 $wt(x) \le c$, it holds that $x^{*(k)}$ is $g(c,k)$-junta.*

### 4.2 Meta-Theorems

We present the meta-theorems for Theorem 1.

**Theorem 8** (PPRG in $\mathsf{NC}^0 \Rightarrow$ hardness of learning)**.** *Let $p(n)$ be an arbitrary polynomial and $\mathscr{C}^k$ be a parameterized class that is FPT dualizable in $\mathsf{NC}^0$ (parameterized by depth) and satisfies the junta-sparse condition. There exist a polynomial $q(n)$ and a constant $\epsilon > 0$ such that for any time-bound function $t(n)$, if there exists a collection of PPRGs in $\mathsf{NC}^0$ against $t(n)$-time adversaries, then $\mathscr{C}$ is not $(t(n^\epsilon)/q(n), 1/p(n))$-learnable on average with respect to a $c$-sparse example distribution samplable with shared randomness and a target distribution samplable by a depth-$d$ $\mathsf{NC}^0$ circuit with $(k, c, d)$-FPT samples.*

*Proof.* Let $G$ be a collection of PPRGs with locality $d_0$, and let $\mathcal{G}$ be its generator, i.e., $\mathcal{G}(1^n; r)$ outputs a description of $G$ in polynomial time for a random seed $r \in \{0,1\}^{\mathsf{poly}(n)}$. Let $f_1, f_2, p_{dual}$ be the functions in Definition 3 for the FPT dualization of $\mathscr{C}$. Then, we select a constant $\epsilon \in (0, 1]$ such that $(\log n \cdot p_{dual}(n))^\epsilon \le n$, i.e., $\log n \cdot p_{dual}(n) \le n^{1/\epsilon}$.

We fix an FPT sample-complexity function $m_{k,c,d}(n) = f_m(k, c, d) \cdot p_m(n)$ arbitrarily, where $f_m \colon \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, and $p_m$ is a polynomial. Then, we select a sufficiently large $D \in \mathbb{N}$ such that $n^D \ge p(\log n \cdot p_{dual}(n)) \cdot p_m(\log n \cdot p_{dual}(n))$. We construct a collection of PPRGs $G^{(D)}$ in Theorem 6 based on $G$. It is easily verified that the locality of $G^{(D)}$ is at most some constant $D'$, i.e., each output of $G^{(D)}$ is computable by a $D'$-junta function. By the junta-sparse condition, any $D'$-junta function is contained in $\mathscr{C}^k$ for some $k \in \mathbb{N}$. The description of $G_i^{(D)}$ for each $i \in [n^{D+1}]$ as a function in $\mathscr{C}^k$ is computable in polynomial time by using $\mathcal{G}$.

Now, we introduce the hard problem for learning $\mathscr{C}$. We specify the example distribution $D_{ex}$ by the following sampling algorithm $S$ using shared randomness. On input $1^n$ and shared randomness $r \in \{0,1\}^{\mathsf{poly}(n)}$, the sampling algorithm $S$ generates the description of $G$ by executing $\mathcal{G}(1^n; r)$. Then, $S$ selects $i \leftarrow_u [n^{D+1}]$ by an (unshared) random seed, and computes $f \in \mathscr{C}^k$ corresponding to the $D'$-junta function $G_i^{(D)}$. Finally, $S$ outputs the dual $f^* \in \{0,1\}^{f_1(k) \cdot p_{dual}(n)}$ of $f$ as an example. We also define the target distribution $D_{targ}$ as the distribution of $x^* \in \mathscr{C}^{f_2(k)}_{f_1(k) \cdot p_{dual}(n)}$ for randomly selected $x \leftarrow_u \{0,1\}^n$.

By the junta-sparse condition, $D_{ex}$ is $c$-sparse for some constant $c \in \mathbb{N}$. Since $\mathscr{C}$ is FPT dualizable in $\mathsf{NC}^0$, the target distribution $D_{targ}$ is samplable by a depth-$d$ $\mathsf{NC}^0$ circuit for some constant $d \in \mathbb{N}$. Therefore, if we assume that $\mathscr{C}$ is $(t(n^\epsilon)/q(n), 1/p(n))$-learnable on average with sample complexity $m_{k,c,d}$ for contradiction, there exists an algorithm that succeeds in $(t(n^\epsilon)/q(n), m_{k,c,d}(n), 1/p(n))$-learning $\mathscr{C}^k$ on average with respect to $D_{ex}$ and $D_{targ}$.

By selecting sufficiently large $q(n)$, we will show that for any time-bound function $T$, any learner $L$ that $(T(n), m_{k,c,d}(n), 1/p(n))$-learns $\mathscr{C}^k$ (on average with respect to $D_{ex}$ and $D_{targ}$) can be converted into a $T(n^{1/\epsilon}) \cdot q(n^{1/\epsilon})$-time adversary that breaks $G$. Since $(t((n^{1/\epsilon})^\epsilon)/q(n^{1/\epsilon})) \cdot q(n^{1/\epsilon}) = t(n)$, any algorithm that succeeds in $(t(n^\epsilon)/q(n), m_{k,c,d}(n), 1/p(n))$-learning $\mathscr{C}^k$ yields a $t(n)$-time adversary $G$. This contradicts that $G$ is a PRG against $t(n)$-time adversaries.

First, we construct an adversary $A$ for $G^{(D)}$ as follows: On input $w \in \{0,1\}^{n^{D+1}}$ and the description of $G^{(D)}$ (note that $w$ is a pseudorandom string generated by $G^{(D)}$ or a truly random string), $A$ simulates the example distribution $D_{ex}$ by selecting a random index $i \leftarrow_u [n^{D+1}]$, computing the $D'$-junta function corresponding to $G_i^{(D)}$ and its dual (for simplicity, we let $G_i^*$ denote this dual string of length $N := f_1(k) \cdot p_{dual}(n)$), and generating a sample $(G_i^*, w_i)$. After generating $m_{k,c,d}(N)$ samples, $A$ also generates a challenge $G_{i_c}^*$ for $i_c \leftarrow_u [n^{D+1}]$ and feeds it to $L$. If $L$ outputs some prediction $b \in \{0,1\}$, then $A$ checks whether $b = w_{i_c}$. If so, $A$ outputs 1; otherwise, it outputs 0. We remark that the running time of $A$ is bounded above by $\mathsf{poly}(n) \cdot T(N)$.

In the case in which $w \leftarrow G^{(D)}(x)$ for $x \leftarrow_u \{0,1\}^n$, we have $w_i = G_i^{(D)}(x) = x^*(G_i^*)$ for all $i$. Therefore, the simulated samples are valid for the target function $x^*$. Furthermore, it is not hard to verify that $A$ executes $L$ on the example distribution $D_{ex}$ and the target distribution $D_{targ}$. Therefore, we have

$$\Pr_{A,U_n,G}[A(G^{(D)}, G^{(D)}(U_n)) = 1] = \Pr_{L,D_{ex},D_{targ}}[L \text{ succeeds in learning}] \geq \frac{1}{2} + \frac{1}{p(N)}.$$

By contrast, in the case in which $w \leftarrow_u \{0,1\}^{n^{D+1}}$, the labels in the simulated samples are selected truly at random. Because any learning algorithm cannot guess a random label not contained in the given samples better than a random guess, i.e., with success probability $1/2$, we have

$$\begin{aligned}\Pr_{A,U_n,G}[A(G^{(D)}, U_{n^{D+1}})) = 1] &= \Pr_{L,D_{ex}}[L \text{ succeeds in learning}] \\ &\leq \frac{1}{2} \cdot \left(1 - \frac{m_{k,c,d}(N)}{n^{D+1}}\right) + 1 \cdot \frac{m_{k,c,d}(N)}{n^{D+1}} \\ &= \frac{1}{2} + \frac{m_{k,c,d}(N)}{2n^{D+1}} \\ &\leq \frac{1}{2} + \frac{f_m(k,c,d) \cdot p_m(N)}{2n \cdot p(\log n \cdot p_{dual}(n)) \cdot p_m(\log n \cdot p_{dual}(n))}\end{aligned}$$

Therefore, for sufficiently large $n$,

$$\begin{aligned}\Pr_{A,U_n,G}[A(G^{(D)}, U_{n^{D+1}})) = 1] &\leq \frac{1}{2} + \frac{f_m(k,c,d) \cdot p_m(N)}{2n \cdot p(N) \cdot p_m(N)} \\ &\leq \frac{1}{2} + \frac{1}{2p(N)}\end{aligned}$$

and the advantage of $A$ is at least

$$\left(\frac{1}{2} + \frac{1}{p(N)}\right) - \left(\frac{1}{2} + \frac{1}{2p(N)}\right) \geq \frac{1}{2p(N)} \geq \frac{1}{2p(n \cdot p_{dual}(n))}.$$

Thus, $A$ successfully breaks $G^{(D)}$.

By Theorem 6, the adversary $A$ for $G^{(D)}$ can be converted into an adversary $A'$ for $G$ such that the running time of $A$ is bounded above by $\mathsf{poly}(n) \cdot T(N) \leq q(n^{1/\epsilon}) \cdot T(\log n \cdot p_{dual}(n)) \leq q(n^{1/\epsilon}) \cdot T(n^{1/\epsilon})$ for a sufficiently large polynomial $q$. □

Next, we prove the opposite direction.

**Theorem 9** (hardness of learning ⇒ PPRG in $\mathsf{NC}^0$)**.** *Let $p(n) = n^{\Theta(1)}$ be a polynomial, and let $\mathscr{C}^k$ be a parameterized class that is FPT dualizable in $\mathsf{NC}^0$ (parameterized by depth). Assume that for any $k \in \mathbb{N}$ and sufficiently large $n \in \mathbb{N}$, the length of the representation for $\mathscr{C}^k$ is at most $p(n)^{1-\epsilon}$ for some constant $\epsilon \in (0,1)$. Then, there exist a polynomial $q(n)$ and a constant $\delta > 0$ such that for any time-bound function $t(n)$, if $\mathscr{C}$ is not $(t(n), 1/p(n))$-learnable on average with respect to a $c$-sparse example distribution samplable with shared randomness and a target distribution samplable by a depth-$d$ $\mathsf{NC}^0$ circuit with $(k,c,d)$-FPT samples, then there exists a collection of PPRGs in $\mathsf{NC}^0$ against $t(n^\delta)/q(n)$-time adversaries.*

*Proof.* We use the hardness assumption for the sample complexity function $m_{k,c,d}(n) = m(n) := p(n)^{1-\epsilon/2}$ (i.e., independent of parameters). Then, there exist constants $k, c, d \in \mathbb{N}$, an example distribution $D_{ex}$, and a target distribution $D_{targ}$ for the hard problem of learning $\mathscr{C}^k$, where $D_{ex}$ is $c$-sparse and samplable with shared randomness, and $D_{targ}$ is samplable by a depth-$d$ $\mathsf{NC}^0$ circuit. We remark that the length of the representation for $\mathscr{C}$ is at most $p(n)^{1-\epsilon}$. Since $D_{targ}$ is samplable by a depth-$d$ $\mathsf{NC}^0$ circuit, each bit of such a representation is determined by a constant number of random seeds for $D_{targ}$. Therefore, without loss of generality, we assume that the length of random bits for $D_{targ}$ is at most $\ell(n) = p(n)^{1-3\epsilon/4} (= n^{\Theta(1)})$ for sufficiently large $n \in \mathbb{N}$.

We construct a collection of weak PPRGs in $\mathsf{NC}^0$, where the indistinguishable parameter is $p(\ell^{-1}(n))^{-\epsilon/2} = n^{-\Theta(1)}$. Then, we apply Theorem 7 to obtain a collection of (standard) PPRGs in $\mathsf{NC}^0$.

By Theorem 5, the hardness assumption implies that $\mathscr{C}^k$ is not RRHS-refutable on average with respect to $D_{ex}$ and $D_{targ}$ with $m(n)$ samples and advantage $p(n)^{-\epsilon/2}$. Now, we introduce the generator $\mathcal{G}$ of PPRGs. On input $1^{\ell(n)}$, the generator $\mathcal{G}$ first generates $m(n)$ examples $x^{(1)}, \ldots, x^{(m(n))} \leftarrow D_{ex}$. Since $D_{ex}$ is samplable with shared randomness, $\mathcal{G}$ can perfectly simulate $D_{ex}$ in polynomial time. Then, $\mathcal{G}$ computes their duals $(x^{(1)})^*, \ldots, (x^{(m(n))})^*$, where the input to each $(x^{(m(n))})^*$ is the dual of the target function selected according to $D_{targ}$. By the junta-sparse condition, these duals $(x^{(1)})^*, \ldots, (x^{(m(n))})^*$ are $O(1)$-junta except with negligible probability when the dual of a target function is given as input. Since $D_{targ}$ is samplable by a depth-$d$ $\mathsf{NC}^0$ circuit whose input is the random seed $r \in \{0,1\}^{\ell(n)}$, and the dual of the target function is computable in $\mathsf{NC}^0$, by considering the composition of $(x^{(1)})^*, \ldots, (x^{(m(n))})^*$, the $\mathsf{NC}^0$ circuit computing the dual, and the $\mathsf{NC}^0$ circuit sampling the target function, we make $m(n)$ $\mathsf{NC}^0$ circuits $G_1(r), \ldots, G_{m(n)}(r)$, where each $G_i$ corresponds to $(x^{(i)})^*$. Finally, $\mathcal{G}$ outputs $G(r) := G_1(r) \circ \cdots \circ G_{m(n)}(r)$ as the description of the $\mathsf{NC}^0$-computable generator.

We remark that the above-mentioned generator is only defined for input size $\ell(n)$. This can be converted into a generator defined for all input sizes $n$ by the standard technique, i.e., for a given $n$-bit random seed, the generator uses only $\ell(n')$ bits, where $n'$ is the maximum integer such that $\ell(n') \leq n$ (for details, refer to the textbook by Goldreich [Gol06]). Let $G$ denote the generator. Then, the length $N$ of the output of $G$ is at least

$$N = m(n') = p(n')^{1-\epsilon/2} = \ell(n')^{1+\frac{\epsilon}{4-3\epsilon}} > \ell(n'+1)^{1+\epsilon'} > n^{1+\epsilon'}$$

for some $\epsilon' \in (0, \frac{\epsilon}{4-3\epsilon})$ and any sufficiently large $n$. Thus, $G$ has polynomial-stretch.

Next, we show that $G$ satisfies the security condition of a weak pseudorandom generator by contradiction. We assume that there exists a $T(n)$-time adversary $A$ such that

$$\left| \Pr_{\mathcal{G},A,U_n}[A(G, G(U_n)) = 1] - \Pr_{\mathcal{G},A,U_N}[A(G, U_N) = 1] \right| > 1/p(\ell^{-1}(n))^{\epsilon/2}. \tag{1}$$

Now, we construct a refuting algorithm $R$ for $\mathscr{C}^k$ as follows. On input $(x^{(1)}, b^{(1)}), \ldots, (x^{(m(n))}, b^{(m(n))})$, the algorithm $R$ constructs the generator $G$ in the same way as $\mathcal{G}$, i.e., $R$ computes $(x^{(i)})^*$ and the composed function $G_i$ for each $i \in [m(n)]$. Then, $R$ executes $A(G, b^{(1)} \circ \cdots \circ b^{(m(n))})$ and returns the same answer. We remark that each $x^{(i)}$ is selected according to $D_{ex}$. Thus, $R$ correctly simulates the distribution of the generator $G$. In the case in which $f \leftarrow D_{targ}$ and $b^{(i)} = f(x^{(i)})$ for each $i$, we have $b^{(i)} = f(x^{(i)}) = (x^{(i)})^*(f^*) = G_i(r)$, where $r$ is the seed for selecting $f$, and the distribution of $b^{(1)} \circ \cdots \circ b^{(m(n))}$ corresponds to $G(U_{\ell(n)})$. By contrast, in the case in which $b^{(i)} \leftarrow_u \{0,1\}$ for each $i$, the distribution of $b^{(1)} \circ \cdots \circ b^{(m(n))}$ corresponds to a uniform distribution. Therefore, by (1), $R$ refutes $\mathscr{C}^k$ on $D_{ex}$ and $D_{targ}$ with $m(n)$ samples and advantage grater than $1/p(\ell^{-1}(\ell(n)))^{\epsilon/2} = p(n)^{-\epsilon/2}$.

By Theorem 5, the refuter $R$ can be converted to a learner with advantage $1/p(n)$. By selecting a sufficiently large polynomial $q(n)$ and a sufficiently large constant $a > 1$, the running time of the learner is bounded above by $q(n) \cdot T(\ell(n)) \leq q(n) \cdot T(n^a)$. Thus, by letting $\delta = 1/a$, any $t(n^\delta)/q(n)$-time adversary for $G$ is converted into a learning algorithm that works in time $q(n) \cdot t(n^{\delta \cdot a})/q(n^a) \leq t(n)$ with advantage $1/p(n)$, which is a contradiction. □

### 4.3 FPT Dualizable Classes with Junta-Sparse Condition

In this section, we present FPT dualization in $\mathsf{NC}^0$ with the junta-sparse condition for $c$-sparse $\mathbb{F}_2$-polynomials, $c$-Fourier-sparse functions, and depth-$d$ $\{\mathrm{OR}, \mathrm{Mod}_m\}$-decision trees. Then, we can show Theorem 1 by applying Theorems 8 and 9 for all polynomial time-bounds $t(n)$. To apply Theorem 9, we leverage the fact that for any $a > 0$ and the parameter of the class, the length of the binary representations of target functions is at most $n^{1+a}$ for sufficiently large input size $n \in \mathbb{N}$.

#### 4.3.1 c-Sparse $\mathbb{F}_2$-Polynomials

For each $c$-sparse $\mathbb{F}_2$-polynomial $f = M_1 + \cdots + M_c$, where each $M_i$ represents a monomial, and for each input $x \in \{0,1\}^n$, we define their duals as a binary string $f^* \in \{0,1\}^{cn+c}$ and a $2c$-sparse $\mathbb{F}_2$-polynomial $x^*$. For simplicity, we assume that $f^*$ is indexed by $\{0, \cdots, n\} \times [c]$ instead of $[cn+c]$. Then, $f^*$ and $x^*$ is determined as follows.

$$f^*_{(i,j)} = \begin{cases} \mathbb{1}(x_i \in M_j) & \text{if } i \in [n] \\ \mathbb{1}(M_j \equiv 1) & \text{if } i = 0 \end{cases}$$
$$x^*(f^*) = \sum_{j \in [c]} \prod_{i : x_i = 1} f^*_{(i,j)} + \sum_{j \in [c]} f^*_{(0,j)}.$$

The dualization above is trivially computable in $\mathsf{NC}^0$. The correctness is verified as follows:

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

In addition, the junta-sparse condition is verified as follows:

**Lemma 3.** *$c$-sparse $\mathbb{F}_2$-polynomials satisfy the junta-sparse condition by the FPT dualization in $\mathsf{NC}^0$ above.*

*Proof.* (1.) Any $n$-input $k$-junta function is represented as an $n$-input $\mathbb{F}_2$-polynomial of degree $k$ and sparsity $2^k$. It is not hard to verify that for any degree-$k$ $2^k$-sparse $\mathbb{F}_2$-polynomial, the Hamming weight of its dual $f^*$ is at most $2^k \cdot k$.

(2.) For any $n, c, c' \in \mathbb{N}$ and $x \in \{0,1\}^n$ with $wt(x) \leq c'$, the dual $x^{*c}$ depends on only $c \cdot wt(x) + c \leq cc' + c$ coordinates. □

### 4.3.2 c-Fourier-Sparse Functions

For each $x \in \{0,1\}^n$ and each $c$-Fourier-sparse function $f = M_1 2^{-\lceil \log c \rceil} \chi_{\alpha_1} + \cdots + M_c 2^{-\lceil \log c \rceil} \chi_{\alpha_c}$, where $M_i \in \{-2^{\lceil \log c \rceil}, \ldots, 2^{\lceil \log c \rceil}\}$ and $\alpha_i \in \{0,1\}^n$ for each $i \in [c]$, we define their duals as a binary string $f^* \in \{0,1\}^{(\lceil \log c \rceil + n + 1)c}$ and a function $x^*$ of Fourier sparsity $c' := 2c\lceil \log c \rceil$.

For each $i \in [c]$, let $b^i \in \{0,1\}^{\lceil \log c \rceil}$ be the binary representation of the absolute value of $M_i$. Then, $f^*$ consists of $c$ triples of $b^i$, $\alpha_i$, and $b^i_{neg} \in \{0,1\}$, where $b^i_{neg} = 1$ iff $M_i < 0$. We also specify $x^*$ as $x^* = \sum_{(i,j) \in [c] \times [\lceil \log c \rceil]} N_{i,j} 2^{-\lceil \log c' \rceil} \chi_{i,j} + N'_{i,j} 2^{-\lceil \log c' \rceil} \chi'_{i,j}$, where $N_{i,j}, N'_{i,j} \in \{-2^{\lceil \log c' \rceil}, \ldots, 2^{\lceil \log c' \rceil}\}$ and $\chi'_{i,j}, \chi'_{i,j} \colon \{0,1\}^{(\lceil \log c \rceil n + 1)c} \to \{-1,1\}$ are determined as follows:

$$\begin{aligned}
N_{i,j} &= 2^{\lceil \log c' \rceil + j - 1 - \lceil \log c \rceil} \; (\leq 2^{\lceil \log c' \rceil - 1}) \\
N'_{i,j} &= -N_{i,j} = -2^{\lceil \log c' \rceil + j - 1 - \lceil \log c \rceil} \\
\chi_{i,j}(f^*) &= (-1)^{b^i_{neg} + \sum_{k : x_k = 1} (\alpha_i)_k} \\
\chi'_{i,j}(f^*) &= (-1)^{b^i_j} \chi_{i,j}(f^*) = (-1)^{b^i_j + b^i_{neg} + \sum_{k : x_k = 1} (\alpha_i)_k}.
\end{aligned}$$

It is not hard to verify that the dualization above is computable in $\mathsf{NC}^0$. The correctness is verified as follows:

$$\begin{aligned}
x^*(f^*) &= \sum_{(i,j) \in [c] \times [\lceil \log c \rceil]} N_{i,j} 2^{-\lceil \log c' \rceil} \chi_{i,j}(f^*) + N'_{i,j} 2^{-\lceil \log c' \rceil} \chi'_{i,j}(f^*) \\
&= \sum_{i \in [c]} \sum_{j \in [\lceil \log c \rceil]} 2^{j - 1 - \lceil \log c \rceil} (-1)^{b^i_{neg} + \sum_{k : x_k = 1} (\alpha_i)_k} - 2^{j - 1 - \lceil \log c \rceil} (-1)^{b^i_j + b^i_{neg} + \sum_{k : x_k = 1} (\alpha_i)_k} \\
&= \sum_{i \in [c]} (-1)^{\sum_{k : x_k = 1} (\alpha_i)_k} 2^{-\lceil \log c \rceil} (-1)^{b^i_{neg}} \sum_{j \in [\lceil \log c \rceil]} 2^j \cdot (1 - (-1)^{b^i_j})/2 \\
&= \sum_{i \in [c]} (-1)^{\langle x, \alpha_i \rangle} 2^{-\lceil \log c \rceil} \cdot (-1)^{b^i_{neg}} \sum_{j \in [\lceil \log c \rceil]} 2^j \cdot b^i_j \\
&= \sum_{i \in [c]} \chi_{\alpha_i}(x) \cdot 2^{-\lceil \log c \rceil} M_i \\
&= f(x).
\end{aligned}$$

In addition, the junta-sparse condition is verified as follows:

**Lemma 4.** *$c$-Fourier sparse functions satisfy the junta-sparse condition by the FPT dualization in* $\mathsf{NC}^0$ *above.*

*Proof.* (1.) Based on the unique Fourier representation, any $n$-input $k$-junta function is represented as a degree-$k$ function of Fourier sparsity at most $2^k$. It is not hard to verify that for any degree-$k$ $2^k$-Fourier sparse function, the Hamming weight of its dual $f^*$ is at most $2^k \cdot (\lceil \log 2^k \rceil + k + 1) = 2^k \cdot (2k+1)$.

(2.) For any $n, c, c' \in \mathbb{N}$ and $x \in \{0,1\}^n$ with $wt(x) \leq c'$, the dual $x^{*c}$ depends on only $2c\lceil \log c \rceil \cdot (2 + wt(x)) \leq 2c\lceil \log c \rceil (2 + c')$ coordinates. □

### 4.3.3 Degree-d $\mathrm{Mod}_\mathrm{m}$-Decision Trees and OR-Decision Trees

In this subsection, we present the FPT dualization for $\mathrm{Mod}_m$-DT that satisfies the junta-sparse condition. Note that the case of OR-DT follows in the same way.

For each $x \in \{0,1\}^n$ and depth-$d$ $\mathrm{Mod}_m$-DT $T$, we define their duals as a binary string $T^* \in \{0,1\}^{(2^d-1)n+2^d}$ and a depth-$(d+1)$ $\mathrm{Mod}_m$-DT $x^*$. For simplicity, we assume that $T^*$ consists of a tuple $t \in \{0,1\}^{(2^d-1)n}$ and $\ell \in \{0,1\}^{2^d}$, and $t$ is indexed by $[2^d-1] \times [n]$ instead of $[(2^d-1)n]$. Let $\{j_1, \dots, j_c\} = \{j \in [n] : x_j = 1\}$, where $c := wt(x)$. Then, $T^*$ (i.e., $t$ and $\ell$) and $x^*$ are defined as follows:

$$\begin{aligned} t_{i,j} &= \mathbb{1}(x_j \text{ is relevant to the query at node } i) \\ \ell_i &= (\text{the label at leaf } i), \end{aligned}$$

and for any $i \in [2^{d+1}-1]$ and $j \in [2^{d+1}]$,

$$(\text{the query at node } i \text{ in } x^*) = \begin{cases} \mathrm{Mod_m}(t_{i,j_1}, \dots, t_{i,j_c}) & i \leq 2^d - 1 \\ \mathrm{Mod_m}(\ell_{i-(2^d-1)}) & i \geq 2^d \end{cases}$$

$$(\text{the label at leaf } j \text{ in } x^*) = \mathbb{1}(\text{leaf } j \text{ is the false subtree of its parent node}).$$

The dualization above is computable in $\mathsf{NC}^0$. We also verify the correctness. On evaluating $x^*(T^*)$, any answer to the query at node $i \in [2^d-1]$ is consistent with the answer to the query at node $i$ in $T(x)$ because

$$\mathrm{Mod_m}(t_{i,j_1}, \dots, t_{i,j_c}) = \mathrm{Mod_m}(x_1 \wedge t_{i,1}, \dots, x_n \wedge t_{i,n}) = \mathrm{Mod_m}(x_{k^i_1}, \dots, x_{k^i_\cdot}),$$

where

$$\{k^i_1, \dots, k^i_\cdot\} = \{k \in [n] : x_k \text{ is relevant to the query at node } i \text{ in } T\}.$$

For any $i \in [2^{d+1}-1] \setminus [2^d-1]$, the answer to the query at node $i$ is $\mathrm{Mod_m}(\ell_{i-(2^d-1)}) = \neg \ell_{i-(2^d-1)}$ for any $m \geq 2$. Note that $x^*$ outputs 1 (i.e., true) iff the answer to the query at degree $d+1$ is false. Thus, $x^*(T^*)$ is consistent with $T(x)$.

Furthermore, the junta-sparse condition is verified as follows:

**Lemma 5.** *For any $m \geq 2$, degree-$d$ $\mathrm{Mod}_m$-DT satisfies the junta-sparse condition by the FPT dualization in $\mathsf{NC}^0$ above.*

*Proof.* (1.) Any $n$-input $k$-junta function is represented as a degree-$k$ $\mathrm{Mod}_m$-DT, where each query is represented as $\mathrm{Mod}_m(x_i) = \neg x_i$ for some $i \in [n]$. It is not hard to verify that the Hamming weight of the dual of such a $\mathrm{Mod}_m$-DT is at most $(2^d-1) + 2^d$.

(2.) For any $n, d, c \in \mathbb{N}$ and $x \in \{0,1\}^n$ with $wt(x) \leq c$, the dual $x^{*d}$ depends on only $(2^d-1) \cdot wt(x) + 2^d \leq (2^d-1)c + 2^d$ coordinates. □

## 4.4 Relaxed Hardness Assumption

In this section, we present the meta-theorem for Theorem 2. First, we introduce a natural condition of parameterized concept classes.

**Definition 11** (junta-composition condition)**.** *Let $\mathscr{C}^k$ be a parameterized class. We say that $\mathscr{C}$ satisfies the junta-composition condition if the following hold:*

1. *For any $k, n', n \in \mathbb{N}$ with $n' \leq n$, it holds that $\mathscr{C}^k_{n'} \subseteq \mathscr{C}^k_n$ (i.e., paddable with dummy inputs).*

2. *There exists a computable $g \colon \mathbb{N} \to \mathbb{N}$ such that any $k$-junta function is contained in $\mathscr{C}^{g(k)}$ for each $k \in \mathbb{N}$.*

3. *There exists a computable $g\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ such that for any $k,k',n,n'\in\mathbb{N}$, $f^{(1)},\dots,f^{(n)}\in\mathscr{C}^k_{n'}$, and $f'\in\mathscr{C}^{k'}_n$, the composite function $f''\colon\{0,1\}^{n'}\to\{0,1\}$ defined as $f''(x)=f'(f^{(1)}(x),\dots,f^{(n)}(x))$ is contained in $\mathscr{C}^{g(k,k')}$. In addition, the representation of $f''$ is computable from $f^{(1)},\dots,f^{(n)}$, and $f'$ in polynomial time.*

It is easily verified that $c$-sparse $\mathbb{F}_2$-polynomials and $c$-Fourier-sparse functions satisfy the junta-composition condition.

Suppose that $G$ is a weak PPRG of output length $n^{1+\epsilon}$, where each bit is computable in $\mathscr{C}^k$ satisfying the junta-composition condition. For the translation to a (standard) PPRG of output length $n^{1+\epsilon'}$ in Theorem 7, we only need the following operations [for details, refer to App13; AK19]: Let $f^{(1)},\dots,f^{(n^{1+\epsilon})}$ be the functions computing each bit of $G$. Then, the operations are either of

- $f^{(i)}(x):=f^{(i_0)}(f^{(i_1)}(x),\dots,f^{(i_n)}(x))$ for some $i_0\in[n^{1+\epsilon}]$ and $i_1,\dots,i_n<i$ (for amplifying the stretch);
- $f^{(i)}(x^{(1)},\dots,x^{(t)}):=f^{(i_1)}(x^{(1)})\oplus\cdots\oplus f^{(i_t)}(x^{(t)})$ for some $t\in\mathbb{N}$ and $i_1,\dots,i_t<i$ (for amplifying the unpredictability); or
- $f^{(i)}(x^{(1)},\dots,x^{(\mathsf{poly}(n))},r):=g(f^{(i_1)}(x^{(1)}),\dots,f^{(i_{\mathsf{poly}(n)})}(x^{(\mathsf{poly}(n))}),r)$ for some $i_1,\dots,i_{\mathsf{poly}(n)}<i$, $r\in\{0,1\}^{\mathsf{poly}(n)}$, and some $O(1)$-junta function $g$ [for applying the extractor presented in AK19],

and the resulting PPRG is computable by $f^{(i_1)},\dots,f^{(i_{n^{1+\epsilon'}})}$ for some indices $i_1,\dots,i_{n^{1+\epsilon'}}$.

If the $f^{(1)},\dots,f^{(n^{1+\epsilon})}\in\mathscr{C}^k$ and $\mathscr{C}^k$ satisfies the junta-composition condition, it is not hard to verify that each $f^{(i)}$ is contained in $\mathscr{C}^{k'}$ for some $k'$ by induction. Therefore, we have the following analog of Theorem 7.

**Theorem 10** ([AK19]). *Let $\mathscr{C}^k$ be a parameterized class satisfying the junta-composition condition. For any $k\in\mathbb{N}$, $a>0$, and $\epsilon,\epsilon'>0$, there exist $k'\in\mathbb{N}$ (computable from $k$), a polynomial $q$, and $\delta\in(0,1)$ such that any collection of weak PPRGs in $\mathscr{C}^k$ of stretch $n^{1+\epsilon}$ and indistinguishable parameter $1/n^a$ against $t(n)$-time adversaries can be converted into a collection of PPRGs in $\mathscr{C}^{k'}$ of stretch $n^{1+\epsilon'}$ against $t(n^\delta)/q(n)$-time adversaries.*

Now, we present the meta-theorem for Theorem 2, where we only assume the FPT dualization and the junta-composition condition.

**Theorem 11** (PPRG in $\mathscr{C}\Rightarrow$ hardness of learning $\mathscr{C}$). *Let $p(n)$ be an arbitrary polynomial, and let $\mathscr{C}^k$ be a parameterized class that is FPT dualizable in a parameterized class $\mathscr{F}^\ell$ and satisfies the junta-composition condition. There exist a polynomial $q(n)$ and a constant $\epsilon>0$ such that for any time-bound function $t(n)$, if there exist $k\in\mathbb{N}$ and a collection of PPRGs in $\mathscr{C}^k$ against $t(n)$-time adversaries, then $\mathscr{C}^k$ is not $(t(n^\epsilon)/q(n),1/p(n))$-learnable on average with respect to an example distribution samplable with shared randomness and an $\mathscr{F}^\ell$-samplable target distribution with $(k,\ell)$-FPT samples.*

*Proof.* (sketch.) The theorem follows in the same way as Theorem 8. The proof is outlined as follows:

First, we assume a collection of PPRGs in $\mathscr{C}^k$ for some $k\in\mathbb{N}$. Then, for each FPT sample complexity $m$, we amplify the stretch sufficiently by applying Theorem 10 so that any learning algorithm with sample complexity $m$ cannot read all the output bits of the generator (where we use the junta-composition condition to apply Theorem 10). We remark that each bit of the generator is computable in $\mathscr{C}^{k'}$ for some $k'$. Next, we specify the hard learning problem where the example distribution $D_{ex}$ is the uniform distribution over the duals of the functions computing the generator, and the target distribution $D_{targ}$ is the distribution of $x^{*k'}$ for $x \leftarrow_u \{0,1\}^n$. It is not hard to verify that $D_{ex}$ is samplable with shared randomness. By FPT dualization for $\mathscr{F}^\ell$, this $D_{targ}$ is a distribution on $\mathscr{C}^{k''}$ for some $k'' \in \mathbb{N}$ and $\mathscr{F}^\ell$-samplable for some $\ell \in \mathbb{N}$. Therefore, this is a valid case for the hardness of learning, and any learner succeeds in learning on average with respect to $D_{ex}$ and $D_{targ}$ with sample complexity $m$ can be converted into the adversary for the collection of PPRGs, as in the proof of Theorem 8. This is a contradiction. □

Next, we show the opposite direction.

**Theorem 12** (hardness of learning $\mathscr{C} \Rightarrow$ PPRG in $\mathscr{C}$)**.** *Let $p(n) = n^{\Theta(1)}$ be a polynomial and $\mathscr{C}^k$ be a parameterized class that is FPT dualizable in $\mathscr{C}^k$ and satisfies the junta-composition condition. Assume that for any $k \in \mathbb{N}$ and sufficiently large $n \in \mathbb{N}$, the length of the representation for $\mathscr{C}^k$ is at most $p(n)^{1-\epsilon}$ for some constant $\epsilon \in (0,1)$. Then, there exists a polynomial $q(n)$ and a constant $\delta > 0$ such that for any time-bound function $t(n)$, if $\mathscr{C}^k$ is not $(t(n), 1/p(n))$-learnable on average with respect to an example distribution samplable with shared randomness and a $\mathscr{C}^{k'}$-samplable target distribution with $(k,k')$-FPT samples, then there exists a collection of PPRGs in $\mathscr{C}$ against $t(n^\delta)/q(n)$-time adversaries.*

*Proof.* (sketch.) The theorem follows in the same manner as Theorem 9. The proof is outlined as follows:

First, we construct a collection of weak PPRGs in $\mathscr{C}^k$ (for some $k \in \mathbb{N}$) based on the hardness assumption of learning. Then, we apply Theorem 10 to convert the weak PPRG into a standard PPRG in $\mathscr{C}$, where we use the junta-composition condition to apply Theorem 10. For the collection of weak PPRGs, we apply the same construction as Theorem 9, i.e., each bit of the generator takes the form of $x^*$ for some $x \in \{0,1\}^*$, where $x$ is an example selected according to the example distribution in the hard learning problem (note that the difference with Theorem 9 is that $x$ is not always sparse in this case). By the FPT dualization in $\mathscr{C}^k$, each bit of the generator is computable in $\mathscr{C}^k$ for some $k \in \mathbb{N}$ when the description of the dual of a target function is given as the input. Remember that a target function in the hard learning problem is samplable in $\mathscr{C}^{k'}$ for some $k' \in \mathbb{N}$, and the dual of the target function is computable in $\mathscr{C}^{k''}$ for some $k'' \in \mathbb{N}$. Therefore, by considering the composite functions of these three types of functions, we can construct a generator whose input is the random seed for selecting a target function. By the junta-composition condition, each bit of the generator is computable in $\mathscr{C}^{k'''}$ for some $k''' \in \mathbb{N}$. □

Theorem 2 holds by applying Theorems 11 and 12 for all polynomial time-bounds $t(n)$.

## 5 Learning vs. PPRG in Constant-Degree Polynomials

We show Theorem 3 in Section 5.1 and Theorem 4 in Section 5.2.

**Theorem 3.** *For any polynomial $p(n), r(n)$, the following are equivalent:*

1. *There exists a PPRG in $\oplus$-$\mathsf{NC}^0$.*
2. *Degree-$d$ $\mathbb{F}_2$-polynomials are not polynomial-time learnable on average with advantage $1/2 - 1/p(n)$ 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.*

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

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

## 5.1 PPRG vs. Learning on Uniform Example Distribution

In this section, we show the equivalence between a PPRG in constant-degree $\mathbb{F}_2$-polynomials (i.e., $\oplus$-$\mathsf{NC}^0$) and average-case hardness of learning constant-degree $\mathbb{F}_2$-polynomials with respect to the uniform example distribution and a target distribution samplable by constant-degree $\mathbb{F}_2$-polynomials. First, we show the following key lemma.

**Lemma 6.** *For any $d \in \mathbb{N}$, let $p \colon \mathbb{F}_2^n \to \mathbb{F}_2$ denote a random degree-$d$ $\mathbb{F}_2$-polynomial, i.e., each coefficient of $p$ is selected uniformly at random from $\{0,1\}$. Let $m(n) = 1/2 \cdot \binom{n}{\le d}$. Then,*

$$\left((U_n^{(1)}, p(U_n^{(1)})), \ldots, (U_n^{(m(n))}, p(U_n^{(m(n))}))\right) \equiv_s \left((U_n^{(1)}, U_1^{(1)}), \ldots, (U_n^{(m(n))}, U_1^{(m(n))})\right).$$

*Proof.* We identify $\left[\binom{n}{\le d}\right]$ with $\{S \subseteq [n] : |S| \le d\}$ in some canonical ordering. For $\ell \in \mathbb{N}$ and $x^{(1)}, \ldots, x^{(\ell)} \in \mathbb{F}_2^n$, we define a matrix $A[x^{(1)}, \ldots, x^{(\ell)}] \in \mathbb{F}_2^{\ell \times \binom{n}{\le d}}$ by $A[x^{(1)}, \ldots, x^{(\ell)}]_{i,S} = \prod_{j \in S} x_j^{(i)}$ for each $i \in [\ell]$ and $S \subseteq [n]$ with $|S| \le d$.

We identify a degree-$d$ $\mathbb{F}_2$-polynomial $p$ with a string in $\mathbb{F}_2^{\binom{n}{\le d}}$ consisting of coefficients. Then, for each $x^{(1)}, \ldots, x^{(m)} \in \mathbb{F}_2^n$ and each $n$-input degree-$d$ $\mathbb{F}_2$ polynomial $p$, the vector $[p(x^{(1)}), \ldots, p(x^{(m)})]^T$ is represented as $A[x^{(1)}, \ldots, x^{(m)}] \cdot p$.

Now, we assume that $x^{(1)}, \ldots, x^{(m)}$ satisfies that $A[x^{(1)}, \ldots, x^{(m)}]$ has full rank. Then, there exists a full-rank matrix $B \in \mathbb{F}_2^{m \times m}$ such that

$$I := [e^1, \ldots, e^m, *, \ldots, *] = B \cdot A[x^{(1)}, \ldots, x^{(m)}],$$

where $e^1, \ldots, e^m \in \mathbb{F}_2^m$, and each $e^i$ is the unit vector (i.e., $e^i_j = 1$ iff $i = j$). In this case, we have

$$A[x^{(1)}, \ldots, x^{(m)}] \cdot p = B^{-1} I \cdot p = B^{-1} \cdot \left[p_1 + f^1\left(p_{m+1}, \ldots, p_{\binom{n}{\le d}}\right), \ldots, p_m + f^m\left(p_{m+1}, \ldots, p_{\binom{n}{\le d}}\right)\right]^T,$$

for some functions $f^1, \ldots, f^m$. Since $B^{-1}$ has full rank, if $p$ is selected uniformly at random, then $[p(x^{(1)}), \ldots, p(x^{(m)})]^T$ is also distributed uniformly at random over the choice of $p$. Thus, it is sufficient to show that the probability that $A[x^{(1)}, \ldots, x^{(m)}]$ does not have full rank is negligible over the choices of $x^{(1)}, \ldots, x^{(m)}$.

Fix $i \in [m]$ arbitrarily. Suppose that we have selected $x^{(1)}, \ldots, x^{(i-1)}$ such that $A[x^{(1)}, \ldots, x^{(i-1)}]$ has full rank, and we select a new $x^{(i)} \in \mathbb{F}_2^n$ uniformly at random. Then, we show that the conditional probability that $A[x^{(1)}, \ldots, x^{(i)}]$ also has full rank with probability at least $1 - 2^{-\Omega(n)}$. If this is correct, then by the union bound, the probability that $A[x^{(1)}, \ldots, x^{(m)}]$ does not have full rank is bounded above by $m \cdot 2^{-\Omega(n)} = O(n^d) \cdot 2^{-\Omega(n)} = \mathsf{negl}(n)$ because there must exist $i \in [m]$ such that $A[x^{(1)}, \ldots, x^{(i)}]$ does not have full rank in such a case.

Let $V \le \mathbb{F}_2^{\binom{n}{\le d}}$ be the linear subspace spanned by the rows in $A[x^{(1)}, \ldots, x^{(i-1)}]$. Note that $\mathrm{dim} V \le i - 1$. For each $x \in \mathbb{F}_2^n$, we define $\tilde{x} \in \mathbb{F}_2^{\binom{n}{\le d}}$, where $\tilde{x}_S = \prod_{j \in S} x_j$ for each $S \subseteq [n]$ with

$|S| \leq d$. Let $U = \{\tilde{x} : x \in \mathbb{F}_2^n\}$. Then, it is not hard to verify that

$$\Pr_{x^{(i)}}\left[A[x^{(1)}, \ldots, x^{(i)}] \text{ does not have full rank} \middle| x^{(1)}, \ldots, x^{(i-1)}\right] = \frac{|V \cap U|}{|U|}.$$

Let $k = \lfloor \log |V \cap U| \rfloor$. Then, we have $2^k \leq |V \cap U| \leq 2^{k+1}$. We fix $2^k$ distinct elements $y^{(1)}, \ldots, y^{(2^k)} \in \mathbb{F}_2^n$ such that $\tilde{y}^{(1)}, \ldots, \tilde{y}^{(2^k)} \in V \cap U$. Let $V' = \text{span}\{y^{(1)}, \ldots, y^{(2^k)}\}$. Then, $V'$ is a linear subspace of $V$. By Lemma 2, we have

$$\binom{k}{\leq d} \leq \text{dim}V' \leq \text{dim}V \leq i - 1 \leq m = \frac{1}{2}\binom{n}{\leq d}.$$

By Lemma 1, there exists a constant $\gamma \in (0, 1)$ such that $k \leq n(1 - \gamma/d)$ for any sufficiently large $n \in \mathbb{N}$. Therefore, we conclude that

$$\begin{aligned}\Pr_{x^{(i)}}\left[A[x^{(1)}, \ldots, x^{(i)}] \text{ does not have full rank} \middle| x^{(1)}, \ldots, x^{(i-1)}\right] &= \frac{|V \cap U|}{|U|} \\ &\leq \frac{2^{k+1}}{2^n} \\ &\leq \frac{2 \cdot 2^{n(1-\gamma/d)}}{2^n} \\ &= 2^{-\frac{\gamma}{d}n+1} \\ &= 2^{-\Omega(n)}\end{aligned}$$

□

We remark that Lemma 6 implies that learning degree-$d$ $\mathbb{F}_2$-polynomials is infeasible with $2^{-1} \cdot \binom{n}{\leq d} = \Omega(n^d)$ samples and non-negligible advantage even for time-unbounded learners with respect to the uniform example distribution and the uniform target distribution over degree-$d$ $\mathbb{F}_2$-polynomials. In this sense, the upper bound on the seed length for a target distribution is essential in Theorems 3 and 4.

We now show one direction from PPRGs to the average-case hardness of learning.

**Theorem 13.** *For any polynomial $p(n)$, there exists a polynomial $q$ such that for any time-bound function $t(n)$, if there exists a PPRG computable by constant-degree $\mathbb{F}_2$-polynomials against $t(n)$-time adversaries, then degree-$d$ $\mathbb{F}_2$-polynomials are not $(t(n)/q(n), 1/p(n))$-learnable on average with respect to a uniform example distribution and a target distribution samplable by a degree-$d'$ $\mathbb{F}_2$-polynomial using an $n$-bit random seed with $(d, d')$-FPT samples.*

*Proof.* We assume that $G$ is a PPRG computable in degree-$d_0$ polynomials for some $d_0 \in \mathbb{N}$. Fix an FPT sample-complexity function $m_{d,d'}(n) = f(d, d') \cdot p_m(n)$ arbitrarily, where $p_m(n)$ is a polynomial. We select $d \in \mathbb{N}$ such that $1/2 \cdot \binom{n}{\leq d} > \log n \cdot p_m(n)$.

We consider a pseudorandom degree-$d$ $\mathbb{F}_2$ polynomial $p_{PR} : \mathbb{F}_2^n \to \mathbb{F}_2$, where each coefficient is selected by a pseudorandom string $G^{(d)}(U_n)$, where $G^{(d)}$ is the PPRG in Theorem 6. Then, we specify a target distribution $D_{targ}$ for the hard problem as the distribution of $p_{PR}$. Since each pseudorandom bit of $G^{(d)}(U_n)$ is computable by a degree-$d'$ $\mathbb{F}_2$-polynomial for some $d' \in \mathbb{N}$, the target distribution $D_{targ}$ is samplable by the degree-$d'$ $\mathbb{F}_2$-polynomial using an $n$-bit random seed.

We prove the theorem by contradiction. Suppose that there exists a $t(n)$-time algorithm $L$ that learns degree-$d$ $\mathbb{F}_2$-polynomial with respect to the uniform example distribution and $D_{targ}$ with

$m_{d,d'}(n)$ samples. Because no learning algorithm can guess a random function non-negligibly better than a random guess, $L$ is converted into a distinguisher $D$ with advantage $1/p(n) - \mathsf{negl}(n)$ for the following two distributions: (1) $(U_n^{(1)}, p_{PR}(U_n^{(1)})), \ldots, (U_n^{(m_{d,d'}(n))}, p_{PR}(U_n^{(m_{d,d'}(n))}))$ and (2) $(U_n^{(1)}, U_1^{(1)}), \ldots, (U_n^{(m_{d,d'}(n))}, U_1^{(m_{d,d'}(n))})$. Since $f(d,d') \cdot p_m(n) < \log n \cdot p_m(n) < 1/2 \cdot \binom{n}{\leq d}$ for sufficiently large $n \in \mathbb{N}$, by Lemma 6, $D$ distinguishes (1) $(U_n^{(1)}, p_{PR}(U_n^{(1)})), \ldots, (U_n^{(m_{d,d'}(n))}, p_{PR}(U_n^{(m_{d,d'}(n))}))$ and (2') $(U_n^{(1)}, p_R(U_n^{(1)})), \ldots, (U_n^{(m_{d,d'}(n))}, p_R(U_n^{(m_{d,d'}(n))}))$, where $p_R$ is a truly random degree-$d$ $\mathbb{F}_2$-polynomial. Then, we can construct an $O(m_{d,d'}(n) + t(n))$-time adversary $A$ for $G^{(d)}$ that is given a pseudorandom or random string $r$, selects a degree-$d$ polynomial $p_r$ by using $r$, makes $m_{d,d'}(n)$ samples for $p_r$ for random inputs, and feeds them to $D$. By Theorem 6, $A$ is converted into an adversary $A'$ breaking $G$. It is not hard to verify that the running time is bounded above by $\mathsf{poly}(n) \cdot O(m_{d,d'}(n) + t(n)) \leq q(n) \cdot t(n)$ for some polynomial $q$. □

Next, we show the opposite direction, i.e., from the average-case hardness of learning to PPRGs. Theorem 3 is obtained by applying Theorems 13 and 14 for all polynomial time-bounds $t(n)$.

**Theorem 14.** *For any polynomial $p(n), r(n)$, there exist a polynomial $q$ and a constant $\epsilon > 0$ such that for any time-bound function $t(n)$, if degree-$d$ polynomials are not $(t(n), 1/2 - 1/p(n))$-learnable on average with respect to a uniform example distribution and a target distribution samplable by a degree-$d'$ $\mathbb{F}_2$-polynomial using an $r(n)$-bit random seed with $(d,d')$-FPT samples, then there exists a PPRG computable by a constant-degree $\mathbb{F}_2$-polynomial against $t(n^\epsilon)/q(n)$-time adversaries.*

*Proof.* Let $m(n)$ and $\ell(n)$ be the polynomials obtained by applying Fact 1 for $p(n)$ and $m^{\oplus}(n) = \ell(n)^2(n + r(n))^2$ (note that these are well-defined because $\ell(n)$ is determined by only $p(n)$). Then, based on the hardness assumption, there exist constants $d, d' \in \mathbb{N}$ and a target distribution $D_{targ}$ on degree-$d$ $\mathbb{F}_2$-polynomials for the hard learning problem with $m(n)$ samples and advantage $1/2 - 1/p(n)$, where $D_{targ}$ is samplable by a degree-$d'$ $\mathbb{F}_2$-polynomial using an $r(n)$-bit random seed. Let $p_{targ} \colon \{0,1\}^{r(n)} \to \{0,1\}^{\binom{n}{\leq d}}$ be the degree-$d'$ $\mathbb{F}_2$-polynomials that select a target function according to $D_{targ}$.

For $R(n) = \ell(n)(n + r(n))$, we construct a PPRG $G \colon \{0,1\}^{R(n)^3} \to \{0,1\}^{R(n)^4}$. We remark that $G$ is defined on only the input length $R(n)^3$. For a PPRG on any input length, we apply the following simple technique. For a given $n$-bit random seed, we find the maximum integer $n'$ such that $R(n')^3 \leq n$, separate the seed into blocks of size $n'$, and apply the original PRG to each block (where the remaining seed of length $n - n' \cdot \lfloor n/n' \rfloor$ is outputted directly). For the security proof, we apply the standard hybrid argument [for details, refer to Gol06]. Since $R(n)^3$ is a polynomial in $n$, it is not hard to verify that the resulting PRG still preserves polynomial-stretch.

Now, we present the construction of $G$ for input length $R(n)^3$. For convenience, we assume that the random seed $R(n)^3$ is separated into $\ell(n)R(n)^2$ strings $x^{i,j} \in \{0,1\}^n$ and $y^{i,j} \in \{0,1\}^{r(n)}$ indexed by $(i,j) \in [R^2(n)] \times [\ell(n)]$. First, $G(x,y)$ generates target functions $f^{i,j} = p_{targ}(y^{i,j})$ for each $i,j$. Then, $G$ computes $b^{i_1,i_2} := \bigoplus_{j_1,j_2 \in [\ell(n)]} f^{i_1,j_1}(x^{i_2,j_2})$ for all $i_1, i_2 \in [R(n)^2]$ and outputs them as a pseudorandom string of length $R(n)^2 \cdot R(n)^2 = R(n)^4$.

In the following, we show that (i) each $b^{i_1,i_2}$ is computed by a constant-degree $\mathbb{F}_2$-polynomial, and (ii) the above-mentioned $G$ is a pseudorandom generator, which implies the theorem.

For statement (i), we remark that for any depth-$d$ $\mathbb{F}_2$-polynomial $f^{i_1,j_1}$ and the input $x^{i_2,j_2}$, the value of $f^{i_1,j_1}(x^{i_2,j_2})$ is computable by depth-$(d+1)$ $\mathbb{F}_2$-polynomial given $f^{i_1,j_1}$ and $x^{i_2,j_2}$ as input because $f^{i_1,j_1}(x^{i_2,j_2}) = \sum_{S:|S| \leq d} f_S^{i_1,j_1} x_S^{i_2,j_2}$. Furthermore, each bit of $f^{i_1,j_1}$ is computable by degree-$d'$ $\mathbb{F}_2$-polynomials in $y^{i_1,j_1}$. Therefore, each $f^{i_1,j_1}(x^{i_2,j_2})$ is computable by an $\mathbb{F}_2$-polynomial of degree $d'(d+1)$, and so is $b^{i_1,i_2} := \bigoplus_{j_1,j_2 \in [n]} f^{i_1,j_1}(x^{i_2,j_2})$.

Next, we show statement (ii) by contradiction. The outline of the proof is as follows. First, by assuming that there exists an adversary $A$ that breaks $G$ with non-negligible advantage, we show that degree-$d$ $\mathbb{F}_2$-polynomials are learnable on $D^{\oplus}_{ex}$ and $D^{\oplus}_{targ}$ with $R(n)^2 - 1$ samples and non-negligible advantage, where $D^{\oplus}_{ex}$ and $D^{\oplus}_{targ}$ are the distributions of $x^{\oplus} = x^{(1)} \circ \cdots \circ x^{(\ell(n))}$ and $f^{\oplus}(x^{\oplus}) := \bigoplus_{i,j} f^{(i)}(x^{(j)})$ for $x^{(1)}, \ldots, x^{(\ell(n))} \leftarrow U_n$ and $f^{(1)}, \ldots, f^{(\ell(n))} \leftarrow D_{targ}$, respectively. Then, by applying the XOR lemma (i.e., Fact 1), we show that degree-$d$ $\mathbb{F}_2$-polynomials are learnable with respect to $U_n$ and $D_{targ}$ with $m(n)$ samples and an advantage of $1/2 - 1/p(n)$, which contradicts the hardness assumption of learning.

For sufficiently large $n \in \mathbb{N}$, we assume that[9]

$$\Pr\left[A(G(U_{R(n)^3})) = 1\right] - \Pr\left[A(U_{R(n)^4}) = 1\right] \geq 1/\mathsf{poly}(n). \tag{2}$$

We construct a learner $L^{\oplus}$ on $D^{\oplus}_{ex}$ and $D^{\oplus}_{targ}$ as follows. On input $(x^{(1)}, b^{(1)}), \ldots, (x^{(R(n)^2)}, b^{(R(n)^2)})$ and a challenge $x_c$, where each $x^{(i)}$ consists of $x^{i,1}, \ldots, x^{i,\ell(n)} \leftarrow U_n$, the learner $L^{\oplus}$ randomly selects $I_1, I_2 \leftarrow_u [R(n)^2]$ and $f^{i,j} \leftarrow D_{targ}$ for each $i < I_1$ and $j \in [\ell(n)]$. For simplicity, $L^{\oplus}$ changes the indices as $x^{(I_2)} := x_c$ and $(x^{(i)}, b^{(i)}) := (x^{(i+1)}, b^{(i+1)})$ for each $i > I_2$, i.e., $L^{\oplus}$ inserts the challenge in the $I_2$-th position in samples. Then, $L^{\oplus}$ executes $A(b^{1,1}, \ldots, b^{R(n)^2, R(n)^2})$, where each $b^{i_1,i_2}$ is defined by

$$b^{i_1,i_2} = \begin{cases} \bigoplus_{j_1,j_2 \in [\ell(n)]} f^{i_1,j_1}(x^{i_2,j_2}) & \text{if } i_1 < I_1 \\ b^{(i)} & \text{if } (i_1 = I_1) \wedge (i_2 < I_2) \\ r^{i_1,i_2} & \text{if } (i_1 = I_1) \wedge (i_2 \geq I_2) \\ r^{i_1,i_2} & \text{if } i_1 > I_1, \end{cases}$$

where $r^{i_1,i_2} \leftarrow_u \{0,1\}$. If $A$ outputs 1, then $L^{\oplus}$ outputs $r^{I_1,I_2}$ as a prediction; otherwise, $1 - r^{I_1,I_2}$.

We show the correctness of $L^{\oplus}$. For each $I_1$ and $I_2$, we define the hybrid distribution $H_{I_1,I_2}$ as the distribution of $b^{1,1}, \ldots, b^{R(n)^2,R(n)^2}$ selected as

$$b^{i_1,i_2} = \begin{cases} \bigoplus_{j_1,j_2 \in [n]} f^{i_1,j_1}(x^{i_2,j_2}) & \text{if } i_1 < I_1 \\ \bigoplus_{j_1,j_2 \in [n]} f^{i_1,j_1}(x^{i_2,j_2}) & \text{if } (i_1 = I_1) \wedge (i_2 \leq I_2) \\ r^{i_1,i_2} & \text{if } (i_1 = I_1) \wedge (i_2 > I_2) \\ r^{i_1,i_2} & \text{if } i_1 > I_1, \end{cases}$$

where $f^{i_1,j_1} \leftarrow D_{targ}$, $x^{i_2,j_2} \leftarrow D_{ex}$, and $r^{i_1,i_2} \leftarrow_u \{0,1\}$. Then, it is easily verified that $H_{1,0} \equiv U_{R(n)^4}$, $H_{R(n)^2,R(n)^2} \equiv G(U_{R(n)^3})$, and $H_{I_1,0} \equiv H_{I_1-1,R(n)^2}$ for each $I_1 \in [R(n)^2]$. Therefore, by inequality 2, we have

$$\Pr[A(H_{R(n)^2,R(n)^2}) = 1] - \Pr[A(H_{1,0}) = 1] \geq 1/\mathsf{poly}(n).$$

For each $I_1, I_2$ selected by $L^{\oplus}$, the probability that $L^{\oplus}$ outputs the correct prediction $b_c$ is

$$\begin{aligned} &\Pr[r^{I_1,I_2} = b_c] \Pr[A(H_{I_1,I_2}) = 1] + \Pr[r^{I_1,I_2} = 1 - b_c] \Pr[A(\bar{H}_{I_1,I_2}) = 0] \\ &\qquad = \frac{1}{2} + \frac{1}{2} \Pr[A(H_{I_1,I_2}) = 1] - \frac{1}{2} \Pr[A(\bar{H}_{I_1,I_2}) = 1], \end{aligned}$$

where $\bar{H}_{I_1,I_2}$ is the same distribution as $H_{I_1,I_2}$ except that the $(I_1, I_2)$-th bit is flipped.

[9] Strictly speaking, we test the behavior of the adversary first, then take a negation according to the result to remove the vertical bars for an absolute value.

We remark that

$$\Pr[A(H_{I_1,I_2-1})=1]=\frac{1}{2}\Pr[A(H_{I_1,I_2})=1]+\frac{1}{2}\Pr[A(\bar{H}_{I_1,I_2})=1].$$

Thus, the probability that $L^{\oplus}$ succeeds in predicting $b_c$ conditioned on $I_1, I_2$ is

$$\begin{aligned}&\frac{1}{2}+\frac{1}{2}\Pr[A(H_{I_1,I_2})=1]-\frac{1}{2}\Pr[A(\bar{H}_{I_1,I_2})=1]\\&\quad=\frac{1}{2}+\frac{1}{2}\Pr[A(H_{I_1,I_2})=1]-\left(\Pr[A(H_{I_1,I_2-1})=1]-\frac{1}{2}\Pr[A(H_{I_1,I_2})=1]\right)\\&\quad=\frac{1}{2}+\Pr[A(H_{I_1,I_2})=1]-\Pr[A(H_{I_1,I_2-1})=1].\end{aligned}$$

Therefore, the success probability of $L^{\oplus}$ is at least

$$\begin{aligned}\sum_{I_1,I_2\in[R(n)^2]}&\frac{1}{R(n)^4}\left(\frac{1}{2}+\Pr[A(H_{I_1,I_2})=1]-\Pr[A(H_{I_1,I_2-1})=1]\right)\\&=\frac{1}{2}+\frac{1}{R(n)^4}(\Pr[A(H_{R(n)^2,R(n)^2})=1]-\Pr[A(H_{1,0})=1])\\&\geq\frac{1}{2}+\frac{1}{\mathsf{poly}(n)R(n)^4}\\&\geq\frac{1}{2}+\frac{1}{\mathsf{poly}(n)},\end{aligned}$$

where the first equality holds because $H_{I_1,0}\equiv H_{I_1-1,R(n)^2}$ for each $I_1\in[R(n)^2]$.

Since $L^{\oplus}$ succeeds in learning on $D^{\oplus}_{ex}$ and $D^{\oplus}_{targ}$ with $R(n)^2-1<R(n)^2=m^{\oplus}(n)$ $(<m^{\oplus}(n\ell(n)))$ samples, **Boost** in Lemma 1 succeeds in learning on $U_n$ and $D_{targ}$ with $m(n)$ samples and advantage $1/2-1/p(n)$ (for sufficiently large $n$). It is not hard to verify that if the running time of $A$ is bounded by $T(n)$, then the running time of the learner is at most $\mathsf{poly}(n)\cdot T(R^3(n))\leq q(n)\cdot T(n^a)$ for a sufficiently large polynomial $q$ and a sufficiently large constant $a\geq 1$. By letting $\epsilon=1/a$, the theorem follows. □

## 5.2 A Collection of PPRGs vs. Learning on Example Distribution Samplable with Shared Randomness

In this section, we show the equivalence between a collection of PPRGs in constant-degree $\mathbb{F}_2$-polynomials and average-case hardness of learning constant-degree $\mathbb{F}_2$-polynomials with respect to an example distribution samplable with shared randomness and a target distribution samplable by constant-degree $\mathbb{F}_2$-polynomials. First, we introduce a useful lemma.

**Lemma 7.** *For $n,d,d'\in\mathbb{N}$, let $p_1,\ldots,p_n\colon\{0,1\}^n\to\{0,1\}$ be degree-$d$ $\mathbb{F}_2$-polynomials, and let $p'\colon\{0,1\}^n\to\{0,1\}$ be a degree-$d'$ $\mathbb{F}_2$-polynomial. We define a degree-$dd'$ $\mathbb{F}_2$-polynomial $q\colon\{0,1\}^n\to\{0,1\}$ as $q(x):=p'(p_1(x),\ldots,p_n(x))$. Then, the representation of $q$ is computed by degree-$(d'+1)$ $\mathbb{F}_2$-polynomials given $p_1,\ldots,p_n,p'$ as the input.*

*Proof.* For each $S\subseteq[n]$ with $|S|\leq dd'$, we show that $q_S$ is computed by a degree-$(d'+1)$ $\mathbb{F}_2$-polynomial given $p_1,\ldots,p_n,p'$ as the input.

We consider the expansion of the following formula:

$$q(x)=\sum_{I\subseteq[n]:|I|\leq d'}p'_I\prod_{i\in I}p_i(x)=\sum_{I\subseteq[n]:|I|\leq d'}p'_I\prod_{i\in I}\sum_{J\subseteq[n]:|J|\leq d}(p_i)_Jx^J.$$

For convenience, let $r^I(x) := p'_I \prod_{i \in I} \sum_{J \subseteq [n]} (p_i)_J x^J$ for each $I$, i.e., $q(x) = \sum_{I:|I| \leq d'} p'_I \cdot r^I(x)$.

Fix $I \subseteq [n]$ with $k := |I| \leq d'$ arbitrarily, and let $I = \{i_1, \ldots, i_k\}$. Since $x_i^2 = x_i$ for each $i \in [n]$, by expanding $r^I$, it is not hard to verify that

$$(r^I)_S = \sum_{\substack{J_1, \ldots, J_k \subseteq [n]: \\ J_1 \cup \cdots \cup J_k = S}} (p_{i_1})_{J_1} \cdot (p_{i_2})_{J_2} \cdot \cdots \cdot (p_{i_k})_{J_k}.$$

Therefore, the degree of $(r^I)_S$ as an $\mathbb{F}_2$-polynomial in $p_1, \ldots, p_n, p'$ is at most $k = |I|$. Since $q_S = \sum_{I:|I| \leq d'} p'_I \cdot (r^I)_S$, the degree of $q_S$ is at most $d' + 1$ as an $\mathbb{F}_2$-polynomial in $p_1, \ldots, p_n, p'$. □

Now, we show one direction from a collection of PPRGs to the average-case hardness of learning.

**Theorem 15.** *For any polynomial $p(n)$, there exist a polynomial $q$ and a constant $\epsilon \in (0,1)$ such that for any time-bound function $t(n)$, if there exists a collection of PPRGs in constant-degree $\mathbb{F}_2$-polynomials against $t(n)$-time adversaries, then degree-$d$ $\mathbb{F}_2$-polynomials are not $(t(n)/q(n), 1/p(n))$-learnable on average with respect to an example distribution samplable with shared randomness and a target distribution samplable by a degree-$d'$ $\mathbb{F}_2$-polynomial using an $n$-bit random seed with $(d, d')$-FPT samples.*

*Proof.* We assume that $G$ is a collection of PPRGs in degree-$d_0$ polynomials for some $d_0 \in \mathbb{N}$. Without loss of generality, we can assume that the output length of $G$ is $n^2$; otherwise, we regard $G^{(1)}$ in Theorem 6 as $G$. Fix an FPT sample-complexity function $m_{d,d'}(n) = f(d, d') \cdot p_m(n)$ arbitrarily, where $p_m(n)$ is a polynomial. We select $d \in \mathbb{N}$ such that $n^{d+1} > \log n \cdot p_m(n^{d_0+1} \cdot \log n) \cdot p(n^{d_0+1} \cdot \log n)$.

Now, we specify an example distribution $D_{ex}$ and a target distribution $D_{targ}$ for the hardness of learning with sample complexity $m_{d,d'}$. We define $D_{ex}$ as the distribution of the concatenation of binary representations $G_{i_1 \cdot n+1}, \ldots, G_{i_1 \cdot n+n}, \ldots, G_{i_{d-1} \cdot n+1}, \ldots, G_{i_{d-1} \cdot n+n}, G_{i_d}$, where $i_1, \ldots, i_{d-1} \leftarrow_u [n-1] \cup \{0\}$ and $i_d \leftarrow_u [n^2]$ (where the choice of $G$ is simulated by shared randomness). We remark that the following composition of the selected polynomials

$$G_{i_1, \ldots, i_d}(x) := G_{i_d}(G_{i_{d-1}+1}(\cdots G_{i_2 \cdot n+1}(G_{i_1 \cdot n+1}(x), \ldots, G_{i_1 \cdot n+n}(x)), \ldots, ), \ldots, G_{i_{d-1}+n}(\ldots))$$

is distributed according to the uniform distribution over $G_1^{(d)}, \ldots, G_{n^{d+1}}^{(d)}$.

The outline of the remaining proof is as follows. First, we show that (i) for any $x \in \{0,1\}^n$, there exists a constant-degree $\mathbb{F}_2$-polynomial $x^*$ such that $x^*(G_{i_1 \cdot n+1}, \ldots, G_{i_d}) = G_{i_1, \ldots, i_d}(x)$ for any choice of $i_1, \ldots, i_d$. Then, we define the distribution $D_{targ}$ as the distribution of $x^*$ for $x \leftarrow_u \{0,1\}^n$ and show that (ii) $D_{targ}$ is samplable by constant-degree $\mathbb{F}_2$-polynomials that takes $x$ as the input (i.e., a random seed). Finally, by a similar proof as Theorem 8, we show that (iii) learning constant-degree $\mathbb{F}_2$-polynomials is hard with respect to $D_{ex}$ and $D_{targ}$.

By induction in $d$, we show that for any $x \in \{0,1\}^n$, there exists a degree-$(d_0+1)^{d-1}$ $\mathbb{F}_2$-polynomial that is given $G_{i_1 \cdot n+1}, \ldots, G_{i_d}$ and outputs $G_{i_1, \ldots, i_d}$. The base step (i.e., $d = 1$) is trivial. For the inductive step, we assume that the claim holds in the case of $d-1$. Then, for any $i \in [n]$, there exists a degree-$(d_0+1)^{d-2}$ $\mathbb{F}_2$-polynomial $q_i$ that is given $G_{i_1 \cdot n+1}, \ldots, G_{i_{d-2} \cdot n+n}, G_{i_{d-1} \cdot n+i}$ and outputs $G_{i_1, \ldots, i_{d-1}+i}$ for each $i_1, \ldots, i_{d-1}$. Now, we apply Lemma 7 for $p_1 = G_{i_1, \ldots, i_{d-1}+1}, \ldots, p_n = G_{i_1, \ldots, i_{d-1}+n}$, and $p' = G_{i_d}$. Then, the degree-$(d_0+1)$ polynomial $q$ in Lemma 7 outputs $G_{i_1, \ldots, i_d}$ for given $G_{i_1, \ldots, i_{d-1}+1}, \ldots, G_{i_d}$. Since each bit of $G_{i_1, \ldots, i_{d-1}+i}$ is computed by $q_i$, this $q$ is implemented by an $\mathbb{F}_2$-polynomial in $G_{i_1 \cdot n+1}, \ldots, G_{i_d}$, where the degree is at most $(d_0+1)^{d-2} \cdot (d_0+1) = (d_0+1)^{d-1}$.

This claim implies statement (i) for the following reason. Since the degree of $G_{i_1, \ldots, i_d}(x)$ is at most $d_0^d$, it is written as

$$G_{i_1, \ldots, i_d}(x) = \sum_{S:|S| \leq d_0^d} (G_{i_1, \ldots, i_d})_S x^S.$$

We can regard this expression as a degree-1 $\binom{n}{\le d_0^d}$-input $\mathbb{F}_2$-polynomial in $G_{i_1,\dots,i_d}$. By the claim above, each bit of the representation of $G_{i_1,\dots,i_d}$ is computed by degree-$(d_0+1)^{d-1}$ $\mathbb{F}_2$-polynomials in $G_{i_1\cdot n+1},\dots,G_{i_d}$. Thus, $G_{i_1,\dots,i_d}(x)$ is computed by a degree-$(d_0+1)^{d-1}$ $\mathbb{F}_2$-polynomial $x^*(G_{i_1\cdot n+1},\dots,G_{i_d})$, which is determined only by $x$. We remark that, by the argument above, we reduce the input size from $O(n^{d_0^d})$ to $O(d\cdot n^{d_0}) \le n^{d_0}\cdot \log n$ at the expense of the degree of the target function.

Next, we show statement (ii), i.e., the target distribution $D_{targ}$ of $x^*$ for $x \leftarrow \{0,1\}^n$ is samplable by degree-$d_0^d$ $\mathbb{F}_2$-polynomials. Based on the argument above, the polynomial $x^*$ is represented as

$$x^*(G_{i_1\cdot n+1},\dots,G_{i_d}) = \sum_{S:|S|\le d_0^d} p^S(G_{i_1\cdot n+1},\dots,G_{i_d})x^S,$$

where $p^S$ is some polynomial of degree $(d_0+1)^{d-1}$ for each $S\subseteq [n]$ with $|S|\le d_0^d$. Thus, for each $T\subseteq \left[((d-1)n+1)\binom{n}{\le d_0}\right]$ (note that $((d-1)n+1)\binom{n}{\le d_0}$ is the input length of $x^*$), the coefficient $(x^*)_T$ is written as

$$(x^*)_T = \sum_{S:|S|\le d_0^d} (p^S)_T\cdot x^S.$$

Therefore, $x^*$ is computed by an $\mathbb{F}_2$-polynomial (given $x$ as the input) of degree $d_0^d$, and $D_{targ}$ is samplable by $\mathbb{F}_2$-polynomials of degree $d_0^d$, where the seed length is $|x|=n$.

Finally, we prove statement (iii) by contradiction. Suppose that there exists an algorithm $L$ that succeeds in $(t(n),1/p(n))$-learning $\mathbb{F}_2$-polynomials with sample complexity $m(n) := m_{(d_0+1)^{d-1},d_0^d}(n)$. Then, we construct an adversary $A$ for $G^{(d)}$ based on $L$, which also yields an adversary for $G$.

On input $w\in\{0,1\}^{n^{d+1}}$ and a description of $G$, where $w$ is a pseudorandom string generated by $G^{(d)}$ or a truly random string, $A$ simulates the example distribution $D_{ex}$ by selecting $G_{i_1\cdot n+1},\dots,G_{i_d}$ for $i_1,\dots,i_{d-1}\leftarrow_u [n-1]\cup\{0\}$ and $i_d \leftarrow_u [n^2]$. For convenience, we let $N$ denote the size of each example, i.e., $N := ((d-1)n+1)\binom{n}{\le d_0} = O(d\cdot n^{d_0+1})$. After generating $m(N)$ samples, $A$ also generates a challenge according to $D_{ex}$ and feeds them to $L$. Let $i_c\in[n^{d+1}]$ be the position in $G^{(d)}$ that corresponds to the challenge. If $L$ outputs some prediction $b\in\{0,1\}$, then $L'$ checks whether $b=w_{i_c}$. If so, $L'$ outputs 1; otherwise, it outputs 0.

In the case in which $w\leftarrow G^{(d)}(x)$ for $x\leftarrow_u \{0,1\}^n$, we have $x^*(G_{i_1\cdot n+1},\dots,G_{i_d}) = G_i^{(d)}(x) = w_i$ for each $(i_1,\dots,i_d)$ and the corresponding position $i\in[n^{d+1}]$. Therefore, the simulated samples are valid for the target function $x^*$. Since $A$ executes $L$ on the example distribution $D_{ex}$ and target distribution $D_{targ}$, we have

$$\Pr_{A,G,U_n}[A(G^{(d)}(U_n))=1] = \Pr_{L,D_{ex},D_{targ}}[L \text{ succeeds in learning}] \ge \frac{1}{2}+\frac{1}{p(N)}.$$

By contrast, in the case in which $w\leftarrow_u \{0,1\}^{n^{d+1}}$, the labels in the simulated samples are selected truly at random. Because no learning algorithm can guess a random label not contained in the given samples better than a random guess, i.e., with a success probability of 1/2, we have

$$\begin{aligned}\Pr_{A,U_{n^{d+1}}}[A(U_{n^{d+1}})=1] &= \Pr_{L,D_{ex}}[L \text{ succeeds in learning}]\\ &\leq \frac{1}{2}\cdot\left(1-\frac{m(N)}{n^{d+1}}\right)+1\cdot\frac{m(N)}{n^{d+1}}\\ &\leq \frac{1}{2}+\frac{m(n^{d_0+1}\log n)}{2n^{d+1}}\\ &\leq \frac{1}{2}+\frac{f((d_0+1)^{d-1},d_0^d)\cdot p_m(n^{d_0+1}\log n)}{2n^{d+1}}\\ &\leq \frac{1}{2}+\frac{n^{d+1}\cdot f((d_0+1)^{d-1},d_0^d)}{2\log n\cdot n^{d+1}\cdot p(n^{d_0+1}\cdot\log n)}\\ &\leq \frac{1}{2}+\frac{1}{2p(n^{d_0+1}\cdot\log n)}\\ &\leq \frac{1}{2}+\frac{1}{2p(N)},\end{aligned}$$

for sufficiently large $n$. Therefore, the advantage of $A$ is at least

$$\left(\frac{1}{2}+\frac{1}{p(N)}\right)-\left(\frac{1}{2}+\frac{1}{2p(N)}\right)=\frac{1}{2p(N)},$$

and $A$ succeeds in breaking $G^{(d)}$.

By Theorem 6, we can construct an adversary for $G$. It is not hard to verify that the running time is bounded above by $t(N)\cdot q(n)\leq t(n^a)\cdot q(n)$ for a sufficiently large polynomial $q$ and a sufficiently large constant $a\geq 1$. For the theorem, we let $\epsilon=1/a$. We remark that for input size $N=((d-1)n+1)\binom{n}{\leq d_0}$, the length of the random seeds for $D_{targ}$ is at most $n\leq N$. □

Next, we show the opposite direction from the average-case hardness of learning to a collection of PPRGs. We obtain Theorem 4 by applying Theorems 15 and 16 for all polynomial time-bounds $t(n)$.

**Theorem 16.** *For any polynomial $p(n), r(n)$, there exists a polynomial $q$ and a constant $\epsilon>0$ such that for any time-bound function $t(n)$, if degree-$d$ $\mathbb{F}_2$-polynomials are not $(t(n),1/2-1/p(n))$-learnable on average with respect to an example distribution samplable with shared randomness and a target distribution samplable by degree-$d'$ $\mathbb{F}_2$-polynomials using an $r(n)$-bit random seed with $(d,d')$-FPT samples, then there exists a collection of PPRGs computable by a constant-degree $\mathbb{F}_2$-polynomial against $t(n^\epsilon)/q(n)$-time adversaries.*

*Proof.* (sketch.) The construction of a collection of PPRGs mainly follows the construction of the PPRG in the proof of Theorem 14.

Let $m(n)$ and $\ell(n)$ be the polynomials obtained by applying Fact 1 for $p(n)$ and $m^{\oplus}(n)=(\ell(n)r(n))^{1+\delta}$, where $\delta>0$ is an arbitrary constant. Then, based on the hardness assumption, there exist constants $d,d'\in\mathbb{N}$, an example distribution $D_{ex}$, and a target distribution $D_{targ}$ on degree-$d$ $\mathbb{F}_2$-polynomial for the hard learning problem with $m(n)$ samples and advantage $1/2-1/p(n)$, where $D_{ex}$ is samplable with shared randomness, and $D_{targ}$ is samplable by a degree-$d'$ $\mathbb{F}_2$-polynomial using $r(n)$-bit random seeds. Let $p_{targ}\colon\{0,1\}^{r(n)}\to\{0,1\}^{\binom{n}{\leq d}}$ be the degree-$d'$ $\mathbb{F}_2$-polynomial selecting a target function according to $D_{targ}$.

For $R(n)=\ell(n)r(n)$, we construct a collection of PPRGs $G\colon\{0,1\}^{R(n)}\to\{0,1\}^{R(n)^{1+\delta}}$. For convenience, we assume that the random seed of length $R(n)$ is separated into $\ell(n)$ strings $y^j\in\{0,1\}^{r(n)}$

indexed by $j \in [\ell(n)]$. Our generator $G$ is specified with $m(n)\ell(n)$ strings $x^{i,j} \in \{0,1\}^n$ indexed by $(i,j) \in [m(n)] \times [\ell(n)]$, which are selected according to $D_{ex}$ in the selection of $G$. First, $G(y)$ generates target functions $f^j = p_{targ}(y^j)$ for each $j$. Then, $G$ computes $b^i := \bigoplus_{j_1,j_2 \in [\ell(n)]} f^{j_1}(x^{i,j_2})$ for each $i \in [m(n)]$ and outputs them as a pseudorandom string of length $m(n) = R(n)^{1+\delta}$.

The security proof for $G$ is almost the same as Theorem 14. Next, we verify that the generator is implemented as a collection of generators in constant-degree $\mathbb{F}_2$-polynomials. For each $i \in [m(n)]$, the $i$-th output bit of $G$ is

$$b^i := \bigoplus_{j_1,j_2 \in [\ell(n)]} f^{j_1}(x^{i,j_2}) = \sum_{j_1,j_2} \sum_{S:|S|\leq d} f^{j_1}_S (x^{i,j_2})^S. = \sum_{j_1,j_2} \sum_{S:|S|\leq d} p^S_{targ}(y^{j_1})(x^{i,j_2})^S,$$

where $p^S_{targ}$ represents the degree-$d'$ polynomial computing the coefficient of $f$ on $S$. Since the selector of $G$ can select a shared randomness, $G$ can simulate the example distribution perfectly. By hardwiring the values of $x^{i,j_2}$ in the expression above (as a part of $G$), each output bit of $G$ is computable by a degree-$d'$ $\mathbb{F}_2$-polynomial given $y$ as the input. Thus, we conclude that $G$ is a collection of PPRGs in constant-degree $\mathbb{F}_2$-polynomials. □

# 6 PPRGs based on Hardness of d-LRPDT

In this section, we verify Corollary 1 based on the proofs of Theorems 9, 14, and 16. For each $d \in \mathbb{N}$, we use the notation $\ell_d$ to refer to the length of the binary representation of degree-$d$ parity decision trees, i.e., $\ell_d(n) = (2^d - 1) \cdot n + 2^d$.

**Corollary 3** (The first item of Corollary 1)**.** *For any $\epsilon \in (0,1)$, if $d$-LRPDT is $(n^{1+\epsilon}, n^{-(1+\epsilon)})$-hard on an $O(1)$-sparse example distribution samplable with shared randomness for some $d \in \mathbb{N}$, then a collection of PPRGs in $\mathsf{NC}^0$ exists.*

*Proof.* Let $p(n) = n^{1+\epsilon}$. Then, for each $d \in \mathbb{N}$, it holds that $\ell_d(n) \leq n^{1+\epsilon(1-\epsilon)/2} = p(n)^{1-\epsilon/2}$ for sufficiently large $n \in \mathbb{N}$. In the proof of Theorem 9, we use the hardness assumption of learning for the sample complexity $m(n) = p(n)^{1-\frac{(\epsilon/2)}{2}} = n^{(1+\epsilon)(1-\frac{\epsilon}{4})} = n^{1+\frac{3\epsilon}{4}-\frac{\epsilon^2}{4}} \leq n^{1+\epsilon}$. Thus, by the FPT dualization of parity decision trees (i.e., Mod$_2$-DTs), the corollary holds. □

**Corollary 4** (The second item of Corollary 1)**.** *For any $\epsilon \in (0,1)$, if $d$-LRPDT is $(n^{1+\epsilon}, n^{-(2+\epsilon)})$-hard on the uniform example distribution for some $d \in \mathbb{N}$, then a PPRG in $\oplus$-$\mathsf{NC}^0$ exists.*

*Proof.* First, we observe that any depth-$d$ parity decision tree is represented by a degree-$d$ $\mathbb{F}_2$-polynomial as follows: Let $T$ be an arbitrary depth-$d$ parity decision tree. For each path $p \in \{0,1\}^d$ in $T$, let $\chi^p_1, \ldots, \chi^p_d$ be the queried linear functions at the internal nodes on the path $p$, and let $b_p$ the binary label at the leaf corresponding to $p$. Then, it is not hard to verify that for each input $x$,

$$T(x) = \sum_{p \in \{0,1\}^d} b_p \prod_{i=1}^{d} \mathbb{1}\{\chi^p_i(x) = p_i\} = \sum_{p \in \{0,1\}^d} b_p \prod_{i=1}^{d} (\chi^p_i(x) + 1 + p_i).$$

Since the degree of $\chi^p_i$ is at most 1, the depth-$d$ parity decision tree $T$ is expressed as a degree-$d$ $\mathbb{F}_2$-polynomial. Furthermore, a random degree-$d$ $\mathbb{F}_2$-polynomial corresponding to a random depth-$d$ parity decision tree is selected by a degree-$(d+1)$ $\mathbb{F}_2$-polynomial according to the expanded expression of the above by using a $\ell_d(n)$-bit random seed. Let $r(n) = n^{1+\epsilon/16} + n$. Then, for any sufficiently large $n \in \mathbb{N}$, we can bound the seed length above by $\ell_d(n) \leq n^{1+\epsilon/16} = r(n) - n$.

Let $p(n) = n^{2+\epsilon}$. Now, we apply the proof of Theorem 14, where we slightly change the construction of the PPRG $G$ and do not use Fact 1 (i.e., the XOR lemma). Specifically, we let $G$ select $r(n)^{1+\epsilon/4}$ examples $x^1, \dots, x^{r(n)^{1+\epsilon/4}} \in \{0,1\}^n$ uniformly at random and $r(n)^{1+\epsilon/4}$ random degree-$d$ parity decision trees $f^1, \dots, f^{r(n)^{1+\epsilon/4}}$ as degree-$d$ $\mathbb{F}_2$-polynomials, and output $b^{i_1,i_2} := f^{i_1}(x^{i_2})$ for each $i_1, i_2 \in [r(n)^{1+\epsilon/4}]$. We remark that by the hybrid argument in the proof of Theorem 14, we can convert the hardness of $d$-LRPDT with advantage $1/p(n)$ and sample complexity $r(n)$ into a weak PPRG $G$ that stretches an $r(n)^{2+\epsilon/4}$-bit random seed to an $r(n)^{2+\epsilon/2}$-bit pseudorandom string, and the indistinguishable parameter is $r(r^{-1}(n^{1/(2+\epsilon/4)}))^{2+\epsilon/2}/p(r^{-1}(n^{1/(2+\epsilon/4)})) = n^{(2+\epsilon/2)/(2+\epsilon/4)}/p(r^{-1}(n^{1/(2+\epsilon/4)}))$, where we interpret the upper bound on the advantage $r(n)^{2+\epsilon/2}/p(n)$ of distinguishers as a function in the seed length $r(n)^{2+\epsilon/4}$. For sufficiently large $n \in \mathbb{N}$, we have $r(n) \leq n^{1+\epsilon/8} \leq n^{1+\epsilon}$. Thus, the hardness assumption of learning satisfies the requirement on the sample complexity in Corollary 4. In addition, we have $r^{-1}(n) \geq n^{1/(1+\epsilon/8)}$, and the indistinguishable parameter is at most

$$\frac{n^{\frac{2+\frac{\epsilon}{2}}{2+\frac{\epsilon}{4}}}}{p(r^{-1}(n^{\frac{1}{2+\frac{\epsilon}{4}}}))} \leq \frac{n^{\frac{8+2\epsilon}{8+\epsilon}}}{n^{\frac{32(2+\epsilon)}{(8+\epsilon)^2}}} = n^{-\frac{32(2+\epsilon)-(8+2\epsilon)(8+\epsilon)}{(8+\epsilon)^2}} = n^{-\frac{8\epsilon-2\epsilon^2}{(8+\epsilon)^2}} = n^{-\Omega(1)}.$$

Thus, we can translate the weak PPRG $G$ into a standard PPRG by Theorem 10, where we use the junta-composition condition of degree-$d$ $\mathbb{F}_2$-polynomials. □

**Corollary 5** (The third item of Corollary 1)**.** *For any $\epsilon \in (0,1)$, if $d$-LRPDT is $(n^{1+\epsilon}, n^{-(1+\epsilon)})$-hard on an example distribution samplable with shared randomness for some $d \in \mathbb{N}$, then a collection of PPRGs in $\oplus\text{-}\mathsf{NC}^0$ exists.*

*Proof.* Let $r(n) = n^{1+\epsilon/4}$ and $p(n) = n^{1+\epsilon}$. Again, we use the observation that $d$-LRPDT is regarded as learning degree-$d$ $\mathbb{F}_2$-polynomials selected by a degree-$(d+1)$ $\mathbb{F}_2$-polynomial by using an $r(n)$-bit random seed. Thus, we can apply the proof of Theorem 16 (without the XOR lemma) and convert the hardness of $d$-LRPDT with advantage $1/p(n)$ and sample complexity $r(n)^{1+\epsilon/4}$ into a collection of weak PPRGs that stretches an $r(n)$-bit random seed to an $r(n)^{1+\epsilon/4}$-bit pseudorandom string, and the indistinguishable parameter is $r(r^{-1}(n))^{1+\epsilon/4}/p(r^{-1}(n)) = n^{1+\epsilon/4}/p(r^{-1}(n))$, where we interpret the upper bound on the advantage $r(n)^{1+\epsilon/4}/p(n)$ of distinguishers as a function in the seed length $r(n)$. For sufficiently large $n \in \mathbb{N}$, we have $r(n)^{1+\epsilon/4} = n^{(1+\epsilon/4)^2} \leq n^{1+\epsilon/2+(\epsilon/2)^2} \leq n^{1+\epsilon}$. Thus, the hardness assumption of learning satisfies the requirement on the sample complexity in Corollary 5. Furthermore, the indistinguishable parameter is at most

$$\frac{n^{1+\frac{\epsilon}{4}}}{p(r^{-1}(n))} = \frac{n^{1+\frac{\epsilon}{4}}}{n^{\frac{1+\epsilon}{1+\frac{\epsilon}{4}}}} = \frac{n^{1+\frac{\epsilon}{4}}}{n^{1+\frac{3\epsilon}{4+\epsilon}}} = n^{-\frac{8\epsilon-\epsilon^2}{4(4+\epsilon)}} = n^{-\Omega(1)}.$$

Thus, we can translate the collection of weak PPRGs into a collection of PPRGs by Theorem 10, where we use the junta-composition condition of degree-$d$ $\mathbb{F}_2$-polynomials. □

## Acknowledgment

We thank the anonymous ITCS reviewers for providing helpful comments and suggestions. Shuichi Hirahara is supported by JST, PRESTO Grant Number JPMJPR2024, Japan. This work was done when the second author was supported by JST, ACT-X Grant Number JPMJAX190M, Japan.

## References

[ABGKR14] 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. Princeton, New Jersey, USA: Association for Computing Machinery, 2014, pp. 251–260.

[ABR12] B. Applebaum, A. Bogdanov, and A. Rosen. “A Dichotomy for Local Small-Bias Generators”. In: *Theory of Cryptography*. Ed. by Ronald Cramer. Springer Berlin Heidelberg, 2012, pp. 600–617.

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. “Public-key cryptography from different assumptions”. In: *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. Ed. by Leonard J. Schulman. ACM, 2010, pp. 171–180. DOI: 10.1145/1806689.1806715. URL: https://doi.org/10.1145/1806689.1806715.

[AC15] 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. Springer International Publishing, 2015, pp. 119–133.

[AIK06] B. Applebaum, Y. Ishai, and E. Kushilevitz. “Cryptography in $NC^0$”. In: *SIAM Journal on Computing* 36.4 (2006), pp. 845–888.

[AIK08] B. Applebaum, Y. Ishai, and E. Kushilevitz. “On Pseudorandom Generators with Linear Stretch in NC0”. In: *Comput. Complex.* 17.1 (Apr. 2008), pp. 38–69.

[AK19] 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)*. 2019, pp. 171–179.

[AK95] D. Angluin and M. Kharitonov. “When Won’t Membership Queries Help?” In: *Journal of Computer and System Sciences* 50.2 (1995), pp. 336–355.

[AKL09] V. Arvind, J. Köbler, and W. Lindner. “Parameterized Learnability of Juntas”. In: *Theor. Comput. Sci.* 410.47-49 (Nov. 2009), pp. 4928–4936.

[AL18] B. Applebaum and S. Lovett. “Algebraic Attacks against Random Local Functions and Their Countermeasures”. In: *SIAM Journal on Computing* 47.1 (2018), pp. 52–79.

[App13] Benny Applebaum. “Pseudorandom Generators with Long Stretch and Low Locality from Random Local One-Way Functions”. In: *SIAM J. Comput.* 42.5 (2013), pp. 2008–2037.

[AR16] Benny Applebaum and Pavel Raykov. “Fast Pseudorandom Functions Based on Expander Graphs”. In: *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*. Ed. by Martin Hirt and Adam D. Smith. Vol. 9985. Lecture Notes in Computer Science. 2016, pp. 27–56.

[BCGIKS21] 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*. Vol. 12828. Lecture Notes in Computer Science. Springer, 2021, pp. 487–516.

[BEHW87] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. "Occam's Razor". In: *Inf. Process. Lett.* 24.6 (Apr. 1987), pp. 377–380.

[BFKL94] 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. 1994, pp. 278–291.

[BHL12] I. Ben-Eliezer, R. Hod, and S. Lovett. "Random low-degree polynomials are hard to approximate". In: *Comput. Complex.* 21.1 (2012), pp. 63–81. DOI: 10.1007/s00037-011-0020-6. URL: https://doi.org/10.1007/s00037-011-0020-6.

[BQ12] A. Bogdanov and Y. Qiao. "On the Security of Goldreich's One-Way Function". In: *Comput. Complex.* 21.1 (Mar. 2012), pp. 83–127.

[CDMRR18] G. Couteau, A. Dupin, P. Méaux, M. Rossi, and Y. Rotella. "On the Concrete Security of Goldreich's Pseudorandom Generator". In: *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.* Ed. by Thomas Peyrin and Steven D. Galbraith. Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 96–124.

[CEMT09] J. Cook, O. Etesami, R. Miller, and L. Trevisan. "Goldreich's One-Way Function Candidate and Myopic Backtracking Algorithms". In: *Theory of Cryptography*. Ed. by Omer Reingold. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 521–538.

[CIKK16] 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. Tokyo, Japan: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[CIKK17] 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)*. Vol. 81. LIPIcs. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 35:1–35:19.

[CM01] M. Cryan and P. B. Miltersen. "On Pseudorandom Generators in $\mathrm{NC}^0$". In: *Mathematical Foundations of Computer Science 2001, 26th International Symposium, MFCS 2001 Marianske Lazne, Czech Republic, August 27-31, 2001, Proceedings*. Vol. 2136. Lecture Notes in Computer Science. Springer, 2001, pp. 272–284.

[Dan16] A. Daniely. "Complexity Theoretic Limitations on Learning Halfspaces". In: *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*. STOC'16. New York, NY, USA: ACM, 2016, pp. 105–117.

[DS16] A. Daniely and S. Shalev-Shwartz. "Complexity Theoretic Limitations on Learning DNF's". In: *Proceedings of 29th Conference on Learning Theory*. Vol. 49. COLT'16. Columbia University, New York, USA: PMLR, 23–26 Jun 2016, pp. 815–830.

[DV21] 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*. Vol. 134. Proceedings of Machine Learning Research. PMLR, 2021, pp. 1358–1394.

[FGKP06] 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)*. Oct. 2006, pp. 563–574.

[FI19] Yuval Filmus and Ferdinand Ihringer. "Boolean constant degree functions on the slice are juntas". In: *Discret. Math.* 342.12 (2019). DOI: 10.1016/j.disc.2019.111614. URL: https://doi.org/10.1016/j.disc.2019.111614.

[Fil22] Yuval Filmus. "Junta threshold for low degree Boolean functions on the slice". In: *CoRR* abs/2203.04760 (2022). DOI: 10.48550/arXiv.2203.04760. arXiv: 2203.04760. URL: https://doi.org/10.48550/arXiv.2203.04760.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. "How to Construct Random Functions". In: *J. ACM* 33.4 (Aug. 1986), pp. 792–807. ISSN: 0004-5411.

[GL89] 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. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 25–32.

[Gol06] O. Goldreich. *Foundations of Cryptography: Volume 1*. New York, NY, USA: Cambridge University Press, 2006. ISBN: 0521035368.

[Gol11] O Goldreich. "Candidate One-Way Functions Based on Expander Graphs". In: Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 76–87.

[GOSSW11] P. Gopalan, R. O'Donnell, R. A. Servedio, A. Shpilka, and K. Wimmer. "Testing Fourier Dimensionality and Sparsity". In: *SIAM Journal on Computing* 40.4 (2011), pp. 1075–1100.

[HKLW88] D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. "Equivalence of Models for Polynomial Learnability". In: *Proceedings of the First Annual Workshop on Computational Learning Theory*. COLT'88. 1988, pp. 42–55.

[IKOS08] 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. Victoria, British Columbia, Canada: Association for Computing Machinery, 2008, pp. 433–442.

[IL90] 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. 1990, pp. 812–821.

[JLS21] 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. Virtual, Italy: Association for Computing Machinery, 2021, pp. 60–73.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability Obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in $\mathrm{NC}^0$". In: *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*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13275. Lecture Notes in Computer Science. Springer, 2022, pp. 670–699.

[JLSW11] J. Jackson, H. Lee, R. Servedio, and A. Wan. "Learning random monotone DNF". In: *Discrete Applied Mathematics* 159.5 (2011), pp. 259–271.

[JS05] J. Jackson and R. Servedio. "Learning Random Log-Depth Decision Trees under Uniform Distribution". In: *SIAM Journal on Computing* 34.5 (2005), pp. 1107–1128.

[Kha93] M. Kharitonov. "Cryptographic Hardness of Distribution-Specific Learning". In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*. STOC '93. San Diego, California, USA: Association for Computing Machinery, 1993, pp. 372–381.

[KM93] E. Kushilevitz and Y. Mansour. "Learning Decision Trees Using the Fourier Spectrum". In: *SIAM J. Comput.* 22.6 (Dec. 1993), pp. 1331–1348.

[KV89] 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. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 433–444.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. "Constant Depth Circuits, Fourier Transform, and Learnability". In: *J. ACM* 40.3 (July 1993), pp. 607–620.

[LP21] 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*. Berlin, Heidelberg: Springer-Verlag, 2021, pp. 11–40. ISBN: 978-3-030-84241-3. DOI: 10.1007/978-3-030-84242-0_2. URL: https://doi.org/10.1007/978-3-030-84242-0_2.

[Nan20] M. Nanashima. "Extending Learnability to Auxiliary-Input Cryptographic Primitives and Meta-PAC Learning". In: *Proceedings of the 33rd Conference on Learning Theory, COLT'20*. Vol. 125. PMLR, Sept. 2020, pp. 2998–3029.

[Nan21] M. Nanashima. "A Theory of Heuristic Learnability". In: *Proceedings of the 34th Conference on Learning Theory, COLT'21*. PMLR, 2021.

[NR06] Moni Naor and Guy N. Rothblum. "Learning to impersonate". In: *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*. Ed. by William W. Cohen and Andrew W. Moore. Vol. 148. ACM International Conference Proceeding Series. ACM, 2006, pp. 649–656.

[NR99] Moni Naor and Omer Reingold. "Synthesizers and Their Application to the Parallel Construction of Pseudo-Random Functions". In: *J. Comput. Syst. Sci.* 58.2 (1999), pp. 336–375.

[ODo14] R. O'Donnell. *Analysis of Boolean Functions*. New York, NY, USA: Cambridge University Press, 2014.

[OS17] I. Oliveira and R. Santhanam. "Conspiracies between Learning Algorithms, Circuit Lower Bounds, and Pseudorandomness". In: *Proceedings of the 32nd Computational Complexity Conference*. CCC'17. Riga, Latvia: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

[OST22] Igor Carboni Oliveira, Rahul Santhanam, and Roei Tell. "Expander-Based Cryptography Meets Natural Proofs". In: *Comput. Complex.* 31.1 (2022), p. 4.

[OW14] R. O'Donnell and D. Witmer. "Goldreich's PRG: Evidence for Near-Optimal Polynomial Stretch". In: *2014 IEEE 29th Conference on Computational Complexity (CCC)*. 2014, pp. 1–12.

[PV88] L. Pitt and L. Valiant. “Computational Limitations on Learning from Examples”. In: *J. ACM* 35.4 (Oct. 1988), pp. 965–984.

[Reg09] O. Regev. “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography”. In: *J. ACM* 56.6 (Sept. 2009).

[RS21] H. Ren and R. Santhanam. “Hardness of KT Characterizes Parallel Cryptography”. In: *36th Computational Complexity Conference (CCC 2021)*. Ed. by Valentine Kabanets. Vol. 200. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 35:1–35:58. ISBN: 978-3-95977-193-1. DOI: 10.4230/LIPIcs.CCC.2021.35. URL: https://drops.dagstuhl.de/opus/volltexte/2021/14309.

[San20] R. Santhanam. “Pseudorandomness and the Minimum Circuit Size Problem”. In: *11th Innovations in Theoretical Computer Science Conference, ITCS 2020*. Vol. 151. LIPIcs. 2020, 68:1–68:26.

[SB14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014. ISBN: 1107057132, 9781107057135.

[Sel08] L. Sellie. “Learning Random Monotone DNF Under the Uniform Distribution”. In: *Proceedings of the 21st Annual Conference on Learning Theory*. COLT’08. Omnipress, 2008, pp. 181–192.

[Sel09] 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. Bethesda, MD, USA: ACM, 2009, pp. 45–54.

[Vad17] S. Vadhan. “On Learning vs. Refutation”. In: *Proceedings of the 2017 Conference on Learning Theory (COLT’17)*. Vol. 65. Proceedings of Machine Learning Research. Amsterdam, Netherlands: PMLR, July 2017, pp. 1835–1848.

[Val84] L. Valiant. “A Theory of the Learnable”. In: *Commun. ACM* 27.11 (1984), pp. 1134–1142. ISSN: 0001-0782. DOI: 10.1145/1968.1972.

[Yao82] A. Yao. “Theory and Application of Trapdoor Functions”. In: *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*. FOCS’82. Nov. 1982, pp. 80–91.

# A Impossibility of Dualization of $\mathbf{NC^0}$

In this section, we show that dualization of $O(1)$-junta (i.e., Boolean-valued functions in $\mathsf{NC}^0$) is impossible even in the statistical setting. The formal statement is the following.

**Theorem 17.** *$O(1)$-junta is not dualizable, i.e., for every $k, k' \in \mathbb{N}$, there is no pair of functions $g$ and $h$ satisfying that for every $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and every $k$-junta function $f\colon \{0,1\}^n \to \{0,1\}$,*

1. *$x^* := g(x)$ is a $k'$-junta function of input size $n' = \mathsf{poly}(n)$;*
2. *$f^* := h(f) \in \{0,1\}^{n'}$; and*
3. *$f(x) = x^*(f^*)$.*

To show Theorem 17, we recall the notion of 2-round communication protocols.

**Definition 12** (2-round communication protocol)**.** *Let $\mathscr{C}$ be a concept class (i.e., a subset of Boolean-valued functions). A 2-round communication protocol for evaluating $\mathscr{C}$ is a pair of deterministic algorithms* $(\mathsf{Input}, \mathsf{Function})$ *(they are possibly not efficiently computable) satisfying that there exist functions $m_{\mathsf{input}}(n)$ and $m_{\mathsf{function}}(n)$ such that for every $n \in \mathbb{N}$, every $x \in \{0,1\}^n$, and every $f \in \mathscr{C}_n$,*

1. $\mathsf{Input}$ *takes $x$ as input and sends a message $a \in \{0,1\}^{m_{\mathsf{Input}}(n)}$ to* $\mathsf{Function}$.
2. $\mathsf{Function}$ *takes $f$ and the message $a$ as input and sends a message $b \in \{0,1\}^{m_{\mathsf{function}}(n)}$ to* $\mathsf{Input}$.
3. $\mathsf{Input}$ *obtains the message $b$ additionally and outputs $f(x)$.*

*For convenience, we call a 2-round communication protocol for evaluating $\mathscr{C}$ with the message-length functions $m_{\mathsf{input}}(n)$ and $m_{\mathsf{function}}(n)$ an $(m_{\mathsf{input}}(n), m_{\mathsf{function}}(n))$-protocol for evaluating $\mathscr{C}$.*

Any concept class $\mathscr{C}$ has a trivial $(n,1)$-protocol (for evaluating $\mathscr{C}$), where $\mathsf{Input}$ sends $x \in \{0,1\}^n$, and $\mathsf{Function}$ sends back $f(x) \in \{0,1\}$. Thus, nontrivial cases are when $\mathsf{Input}$ does not send the whole input $x$.

Now, we show Theorem 17 by observing that any dualization of $\mathsf{NC}^0$ yields a 2-round communication protocol for evaluating $O(1)$-junta with short messages, but such a protocol does not exist information theoretically.

*Proof of Theorem 17.* Fix $k, k' \in \mathbb{N}$ arbitrarily. Theorem 17 follows from Claims 1 and 2.

**Claim 1.** *If there exist the functions $g$ and $h$ for dualization as in Theorem 17, then there exists an $(O(\log n), O(1))$-protocol for evaluating $k$-junta.*

**Claim 2.** *For any $\epsilon > 0$, there is no $((1-\epsilon)n, o(\log n))$-protocol for evaluating $k$-junta.*

*Proof of Claim 1.* We can construct an $(O(\log n), O(1))$-protocol for evaluating $k$-junta based on $g$ and $h$ as follows: for any $n \in \mathbb{N}$, $x \in \{0,1\}^n$, and any $k$-junta function $f \colon \{0,1\}^n \to \{0,1\}$,

1. $\mathsf{Input}(x)$ computes the dual $x^* = g(x)$ and sends all indices $i_1, \dots, i_{k'} \in [n']$ of the relevant variables of $x^*$ to $\mathsf{Function}$, where the message length is at most $O(k' \log n') = O(\log n)$.
2. $\mathsf{Function}(f; i_1, \dots, i_{k'})$ computes the dual $f^* \in \{0,1\}^{n'}$ and sends $f^*_{i_1}, \dots, f^*_{i_{k'}} \in \{0,1\}$ that are relevant to computing $x^*(f^*)$ to $\mathsf{Input}$, where the message length is at most $O(k') = O(1)$.
3. $\mathsf{Input}$, given $f^*_{i_1}, \dots, f^*_{i_{k'}}$, computes and outputs $x^*(f^*) = f(x)$.

□

*Proof of Claim 2.* Suppose that there exists a $((1-\epsilon)n, m(n))$-protocol $(\mathsf{Input}, \mathsf{Function})$ for evaluating $k$-junta, where $\epsilon > 0$ and $m(n) = o(\log n)$. We derive a contradiction.

Fix a sufficiently large $n \in \mathbb{N}$ with $\epsilon n - 1 \geq 2^{m(n)} = o(n)$ arbitrarily. We can classify each input string $x \in \{0,1\}^n$ according to the massage sent by $\mathsf{Input}(x)$. Since the length of the message is $(1-\epsilon)n$, the number of possible messages is at most $2^{(1-\epsilon)n}$. Thus, there exists a message $a \in \{0,1\}^{(1-\epsilon)n}$ such that $S_a = \{x \in \{0,1\}^n : \mathsf{Input}(x) \text{ sends } a\}$ has cardinality at least $2^n/2^{(1-\epsilon)n} = 2^{\epsilon n}$.

We focus on the case in which the given input $x$ is contained in $S_a$. By the definition of $S_a$, the first message sent by $\mathsf{Input}$ is fixed to $a$. Thus, the second message sent by $\mathsf{Function}$ is determined only by a given $k$-junta function $f$. Again, we classify $k$-junta functions according to the second message as follows: for every $b \in \{0,1\}^{m(n)}$,

$$T_b = \{f \colon \{0,1\}^n \to \{0,1\} | f \text{ is } k\text{-junta and } \mathsf{Function}(f; a) \text{ sends } b\}.$$

We show that there exist $x \in S_a$ and $f, f' \in T_b$ such that $f(x) \neq f'(x)$ holds. This contradicts the correctness of $(\mathsf{Input}, \mathsf{Function})$ because, in the both cases of $\{(x, f), (x, f')\}$, the transcript is the same (i.e., the first message is $a$, and the second message is $b$); thus, Input cannot distinguish between $f$ and $f'$ and outputs the same value $y \in \{0,1\}$, and $f(x) = f'(x) = y$ must hold for the correctness. Thus, our goal is to find such $x \in S_a$ and $f, f' \in T_b$.

Let $d = |S_a| \geq 2^{\epsilon n}$. For each $j \in [n]$, we define $v^j \in \{0,1\}^d$ as $v^j = v^j_1 \cdots v^j_d$, where $v^j_i$ is the $j$-th bit of the $i$-th string $x$ in $S_a$ (in lexicographic order) for each $i \in [d]$. If there are at most $c$ distinct vectors in $v^1, \ldots, v^n$ (say, $v^{j_1}, \ldots, v^{j_c}$), then the cardinality of $S_a$ is at most $2^c$ because each $x \in S_a$ is determined only by the patterns of $(v^{j_1}_i, \ldots, v^{j_c}_i) \in \{0,1\}^c$, where $i$ is the lexicographic order of $x$ in $S_a$. Since $|S_a| \geq 2^{\epsilon n}$, there are at least $c \geq \epsilon n$ distinct vectors $v^{j_1}, \ldots, v^{j_c}$ in $v^1, \ldots, v^n$.

We consider 1-junta functions (i.e., $k$-junta functions) $\chi_{j_1}, \ldots, \chi_{j_c}$, where $\chi_{j_\ell}(x) = x_{j_\ell}$ for each $\ell \in [c]$. Remember that the number of the separation $\{T_b\}_{b \in \{0,1\}^{m(n)}}$ of $k$-junta functions is at most $2^{m(n)} \leq \epsilon n - 1 \leq c - 1$. Thus, by the pigeonhole principle, there exist $\ell, \ell' \in [c]$ and $b \in \{0,1\}^{m(n)}$ such that $\chi_{j_\ell}, \chi_{j_{\ell'}} \in T_b$. Since $v^{j_\ell}$ and $v^{j_{\ell'}}$ are distinct vectors, there exists $i \in [n]$ such that $v^{j_\ell}_i \neq v^{j_{\ell'}}_i$. Let $x \in S_a$ be the $i$-th string in $S_a$.

We verify that $x \in S_a$ and $\chi_{j_\ell}, \chi_{j_{\ell'}} \in T_b$ satisfy the condition that $\chi_{j_\ell}(x) \neq \chi_{j_{\ell'}}(x)$ for contradiction as follows:

$$\chi_{j_\ell}(x) = x_{j_\ell} = v^{j_\ell}_i \neq v^{j_{\ell'}}_i = x_{j_{\ell'}} = \chi_{j_{\ell'}}(x).$$

□

□