A Theory of Heuristic Learnability

Mikito Nanashima NANASHIMA.M.AA@IS.C.TITECH.AC.JP
Department of Mathematical and Computing Science, Tokyo Institute of Technology

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

Which concepts can we learn efficiently *on average*? In this paper, we investigate the capability of a natural average-case learning framework, heuristic PAC (heurPAC) learning to answer this and some other related questions. Roughly speaking, we say that a concept class is heurPAC learnable if there exists a learning algorithm that given $n, s \in \mathbb{N}$ and $\epsilon, \delta, \eta \in (0, 1]$ as input, learns all but η fraction of n-input target functions represented as s-bit strings in the class from passively collected examples and then outputs an ϵ-close hypothesis with failure probability at most δ in polynomial-time in n, s, $\epsilon^{-1}, \delta^{-1}$, and η^{-1}, where each example is generated according to some example distribution.

First, we establish a positive learnability result. Specifically, we show that a simple Fourier-based algorithm heurPAC learns $\Omega(\log n)$-junta functions on the uniform distribution, which is a central open question in the original PAC learning model. Our technical contribution is to introduce the notion of elusive functions that captures hard-to-learn cases and to establish a polynomial relation between the running time and the fraction of such elusive functions. Second, we present clear relations between heurPAC learnability and cryptography. Particularly, we show that for any efficiently evaluated class $\mathscr{C}$, (1) if $\mathscr{C}$ is not heurPAC learnable, then an auxiliary-input one-way function (AIOWF) exists; (2) if $\mathscr{C}$ is not heurPAC learnable on the uniform distribution, then an infinitely-often one-way function (io-OWF) exists. As a corollary, we also present new characterizations for AIOWF and io-OWF based on heurPAC learnability, which is conceptually stronger than the previous ones that are based on average-case learnability for fixed parameters. These results show that our framework might yield heuristic learners with theoretical guarantees for broader classes than the usual PAC learning framework, and *any* efficiently evaluated class has a potential for such a heuristic learner or a secure cryptographic primitive. Through this paper, we suggest further research toward the win-win "learning vs. cryptography" paradigm.

Keywords: PAC learning, computational learning theory, average-case learnability, cryptography.

1. Introduction

Which concepts can we learn efficiently from experience? This is one of the most central questions in computational learning theory (CoLT). The most studied model which captures the notion of "learning from experience" is the Probably Approximately Correct (PAC) learning model, introduced by Valiant (1984). Roughly speaking, we say that a concept class $\mathscr{C}$, defined as a set of Boolean-valued functions, is (polynomially) PAC learnable if there exists an algorithm which learns *any* unknown target function $f \in \mathscr{C}$ (which represents a "concept") under *any* example distribution D (which represents an "environment") in the following sense: (1) the learner is given an accuracy parameter $\epsilon \in (0, 1]$, a confidence parameter $\delta \in (0, 1]$, and passively collected examples of the form $(x, f(x))$ as information on f, where each x is selected independently and identically according to D, and (2) the learner is asked to generate a good hypothesis h which is ϵ-close to f (i.e., $\Pr_{x \sim D}[h(x) \neq f(x)] \leq \epsilon$) with probability at least $1 - \delta$ efficiently, or more specifically, in polynomial-time in the size of input for f, the length of binary representation for f, ϵ^{-1}, and δ^{-1}. It is worthy to note that many researchers have studied PAC learnability for several natural classes such as Boolean circuits, formulas, algebraic polynomials, automatons, decision trees, threshold functions, and so on since Valiant introduced this model. Further, there are not only theoretical interests but also a series of work that have resulted in several sophisticated strategies for learning; for instance, Occam's razor (Blumer et al., 1987) and boosting (Schapire, 1990). For further backgrounds, refer to textbooks on the subject, for instance, written by Kearns and Vazirani (1994) and Wigderson (2019, Section 17).

Although Valiant's PAC learning model provides a natural formulation for the task of learning, the condition for "learnable" seems quite strict. Despite much effort by researchers, few concept classes are known to be PAC learnable at the moment: $O(1)$-degree polynomial threshold functions (Blumer et al., 1989) and $O(1)$-degree $\mathbb{F}_2$-polynomials (Helmbold et al., 1992). Further, several natural classes are unlearnable under some reasonable assumptions (e.g., Kearns and Valiant, 1994; Daniely and Shalev-Shwartz, 2016). Thus, many researchers have sought a way to overcome this difficulty by considering a weaker condition for learnable; for instance, allowing a learner to have superpolynomial resources (e.g., quasi-polynomial and nontrivial savings) or additional query access (e.g., membership queries), or focusing on only restricted example distributions (e.g., the uniform distribution). In fact, almost all remarkable positive results in CoLT have been developed in such relaxed frameworks including the work on learning deterministic finite automatons (DFAs) (Angluin, 1987), AC^0 circuits (Linial et al., 1993), decision trees (DTs) (Kushilevitz and Mansour, 1993), disjunctive normal form formulas (DNFs) (Jackson, 1997), and $\mathsf{AC}^0[p]$ circuits for any prime number p (Carmosino et al., 2016).

One of such relaxations is by considering the average-case learnability. In other words, we try to learn *almost all* target functions in the class instead of *all* target functions. This relaxation is reasonable under the following observation: the notion of hard-to-learn is rephrased as looking "random" in the sense that any feasible algorithm cannot find crucial knowledge for learning from its behavior, and in real life, such "random" concepts seem empirically rare. In fact, this observation has been theoretically confirmed for several classes such as DNFs (Aizenstein and Pitt, 1995), and some novel techniques for average-case learning have been proposed (e.g., Jackson and Servedio, 2005; Sellie, 2009; Jackson et al., 2011; Angluin and Chen, 2015). As another line of research, cryptographic primitives based on average-case learnability were proposed by Blum et al. (1994); Nanashima (2020). Presently, such implications for (worst-case) PAC learnability are still not known.

Despite its advantage and importance, the theory of average-case learnability is less developed compared to other learning frameworks or the original average-case theory on computational complexity. In fact, the meanings of "learnable on average" are implicit and slightly vary according to contexts, and some of them seem weak for the formulation of "learnable on average" (see also Section 1.2). Our main goal is to suggest a standard and natural learning framework for discussing average-case learnability. We will then provide fundamental knowledge and advantage of the proposed framework to invoke further research for developing the theory of average-case learnability.

1.1. This Work: Heuristic PAC Learning

Our framework is informally motivated by the natural demands for heuristic learning: more resources (i.e., computational costs and examples) yield better heuristics, and for better heuristics, the number of resources needed is supposed not to increase drastically. Generally, to collect resources for learning requires more or lower costs. A heuristic learner satisfying the above demands yields an effective learning procedure through the interactive use of the learner: First, we begin with learning under a reasonable amount of resources, and then based on the result obtained, we collect and provide more resources to the learner. We repeat this until a desirable result is obtained.

However, it is not easy to reflect the above demands in the learning framework: the meaning of "good" heuristic learning is quite ambiguous. At least a good heuristic learner should be able to learn a "large" fraction of target functions "accurately" in the best possible way. The problem is that, in general, there might be a tradeoff between such accuracy and largeness, and the priority is also different depending on the situation.

Our basic idea to resolve the above problem is to introduce a new parameter, *heuristic* parameter η, to the original PAC learning framework. To be more specific, a learner takes three parameters as input: an accuracy parameter $\epsilon \in (0,1]$, a confidence parameter $\delta \in (0,1]$ (as usual PAC learning), and an additional heuristic parameter $\eta \in (0,1]$. Afterward, the learner is asked to PAC learn all but η fraction of target functions in the class instead of all targets functions with accuracy ϵ and confidence δ within a feasible number of steps, i.e., polynomial-time in ϵ^{-1}, δ^{-1}, and η^{-1}.

Now, we are ready to present the formal definition of the above framework, which we refer to as *heuristic* PAC learning. In this paper, we define a concept class by an evaluation rule $E = \{E_n : \{0,1\}^* \times \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$, where E_n is given a binary representation for an n-input target function f and its input $x \in \{0,1\}^n$ and evaluates the value of $f(x)$. Note that any evaluation rule E naturally determines a concept class $\mathscr{C} = \{\mathscr{C}_n\}_{n\in\mathbb{N}}$ as follows: for each $n \in \mathbb{N}$,

$$\mathscr{C}_n = \{f(x) := E(u,x) : u \in \{0,1\}^*\}\,.$$

For a function $s : \mathbb{N} \to \mathbb{N}$, we say that a concept class $\mathscr{C}$ is $s(n)$-represented if $\mathscr{C}_n$ is determined by an evaluation $E_n : \{0,1\}^{s(n)} \times \{0,1\}^n \to \{0,1\}$ for each $n \in \mathbb{N}$, i.e., each n-input target function has a representation of length $s(n)$. For convenience, we identify a function $f : \{0,1\}^n \to \{0,1\}$ in an $s(n)$-represented class with a binary string $f \in \{0,1\}^{s(n)}$. Our learning framework is defined for $s(n)$-represented concept classes as follows.

Definition 1 (Heuristic PAC learning) *Let $s : \mathbb{N} \to \mathbb{N}$ and $\mathscr{C}$ be an $s(n)$-represented concept class. We say that a randomized oracle machine L, herein referred to as a learner, (polynomially) heuristic PAC (heurPAC) learns $\mathscr{C}$ if L satisfies the following conditions:*

- *L is given $n, s(n) \in \mathbb{N}$, parameters $\epsilon, \delta, \eta \in (0,1]$ and access to an example oracle $EX(f,D)$ determined by a target function $f : \{0,1\}^n \to \{0,1\}$ and an example distribution D on $\{0,1\}^n$;*
- *for each access, EX(f,D) returns an example of the form $(x, f(x))$, where x is selected identically and independently with respect to D;*
- *for any $n \in \mathbb{N}$, $\epsilon, \delta, \eta \in (0,1]$, example distribution D on $\{0,1\}^n$, L outputs, for at least $(1-\eta)$-fraction of target functions $f \in \{0,1\}^{s(n)}$, a hypothesis $h : \{0,1\}^n \to \{0,1\}$ that is ϵ-close to f under D with probability at least $1-\delta$; namely, L satisfies the following condition:*

$$\Pr_{f\sim\{0,1\}^{s(n)}}\left[\Pr_{L,EX(f,D)}\left[L^{EX(f,D)} \text{ outputs } h \text{ such that } \Pr_{x\leftarrow D}[h(x)\neq f(x)] \leq \epsilon\right] \geq 1-\delta\right] \geq 1-\eta;$$

- *L halts in polynomial-time in $n, s(n), \epsilon^{-1}, \delta^{-1}$, and η^{-1} for each input.*

We say that $\mathscr{C}$ is (polynomially) heurPAC learnable if there is a learner that heurPAC learns $\mathscr{C}$.

For a family of distributions $D = \{D_n\}_{n\in\mathbb{N}}$, where D_n denotes a distribution on $\{0,1\}^n$, we say that $\mathscr{C}$ is heurPAC learnable on D if there exists a learner which heurPAC learns $\mathscr{C}$ on the example distribution D (instead of all example distributions).

It should be noted that the above definition relies on the assumption that any concept is represented in the same length $s(n)$, and the reader may think that the uniform distribution over all strings of length $s(n)$ is quite unnatural as distribution on target functions. In previous work, particularly, such a distribution on target functions is usually given as a uniform distribution over valid target functions or by some randomized procedure. However, we claim that the above formulation about a distribution on target functions is applicable for almost all such cases. For instance, as long as a natural distribution on target functions is polynomial-time sampleable in n and $s(n)$ with non-negligible success probability, we can automatically find a representation for our formulation without changing its learnability. We demonstrate this by regarding random seeds for selecting a target function as a new representation. Refer to Appendix B.1 for a formal statement on this. Therefore, in the subsequent argument, we may implicitly consider other forms of distributions on target functions when it is trivially polynomial-time sampleable with non-negligible probability.

Another natural learning framework is the prediction model introduced by Pitt and Warmuth (1990), whose capability is known to be equivalent to the PAC learning model (Haussler et al., 1988). In fact, the above heurPAC learning model can be regarded as a natural average-case extension of the PAC learning model in the sense that its equivalence to the prediction model is preserved. In other words, the heurPAC learnability is equivalent to the learnability determined by a natural average-case extension of the prediction model (Definition 21). Refer to Appendix B.2 for a formal statement on this.

1.2. Comparison with Related Models

In average-case learning on the uniform distribution, positive results have been derived for random $O(\log n)$-depth DTs (Jackson and Servedio, 2005), $\mathsf{poly}(n)$-size monotone DNFs (Sellie, 2008; Jackson et al., 2011), and $\mathsf{poly}(n)$-size DNFs (Sellie, 2009). The main difference between their work and ours is in the setting of a heuristic parameter η. Their work guarantees a polynomial-time learner only in the case where $\eta = 1/\mathsf{poly}(n)$ for the input size n, and the learnability for $\eta = n^{-\omega(1)}$ is not taken into account. Additionally, some of the learners (e.g., Jackson and Servedio, 2005) do not work in polynomial-time[1] in η^{-1}. Angluin and Chen (2015) and its related work presented a positive results for random DFAs. In addition to the difference in a heuristic parameter, their learner requires richer information as labels in examples than binary labels (i.e., states of DFAs).

Generally, the above results were shown by (1) introducing a particular learning strategy L, (2) finding a "good" property P such that L can learn any target function with P in polynomial-time, and (3) showing that a random target function satisfies P with probability $p := 1 - n^{-\omega(1)}$. However, this type of argument gives rise to a certain threshold on η depending on p. Therefore, for heurPAC learning, we may need to introduce a smoother classification of target functions on the level of hardness of learning in step (2) and establish polynomial relations among the level of hardness, the computational complexity of the learner, accuracy, and a heuristic parameter.

1. Note that $O(n^{c \log\log c})$-time or even $O(n^{2^{2^c}})$-time are polynomial-time with respect to $\eta = n^{-c}$ for a constant c.

Blum et al. (1994) constructed a cryptographic primitive (i.e., infinitely-often one-way function) based on the hardness of average-case learning on a polynomial-time sampleable distribution[2]. Their learning model corresponds to our model where parameters are fixed in advance to arbitrary functions $\epsilon = 1/\mathsf{poly}(n)$, $\eta = 1/\mathsf{poly}(n)$, and $\delta = 1/\mathsf{poly}(n)$. This condition is essentially based on the security condition for weak one-way functions, and these parameters are determined by a success probability of an adversary for the cryptographic primitive. A similar result for a weaker cryptographic primitive (i.e., auxiliary-input one-way function) was reported by Nanashima (2020) based on average-case weak learnability on all example distributions, where parameters are fixed in advance to some functions $\epsilon = 1/2 - 1/\mathsf{poly}(n)$, $\eta = 1 - 1/\mathsf{poly}(n)$, and $\delta = 1 - 1/\mathsf{poly}(n)$. In addition to the weakness of learnability, his model is unfamiliar in the sense that the distribution on target functions is also given as input. Note that it is not clear whether boosting techniques (e.g., Schapire, 1990; Boneh and Lipton, 1993) can be applied to show the equivalence between weak learnability for fixed parameters and heurPAC learnability. The main reason is that an average-case weak learner is not guaranteed to succeed even in other settings on a target function or an example distribution for boosting.

2. Our Results

As discussed in Section 1.2, a well-structured knowledge of hard-to-learn functions seems crucial for the heurPAC learnability. Therefore, the natural question is whether the heurPAC learning model is more capable than the PAC learning model. As a first result, we provide an affirmative answer to this question by presenting a heurPAC learner for junta functions on the uniform distribution.

A junta function is a function whose value is determined by only a small part of the input, formally defined as follows.

Definition 2 (Junta functions) *For a function $f : \{0,1\}^n \to \{0,1\}$, we say that a coordinate $i \in \{1, \ldots, n\}$ is relevant if $f(x) \neq f(y)$ for some points $x, y \in \{0,1\}^n$ which differ only at the coordinate i. For $k \leq n$, we say that f is a k-junta function if it has at most k relevant coordinates.*

Technically, we assume that any k-junta function is specified by a truth table of k-input function (we refer to it as a base function for convenience) and correspondence between k coordinates in $\{1, \ldots, n\}$ and input for the base function. As a natural distribution on k-junta functions, we consider a uniform distribution over pairs of a base function and correspondence to coordinates.

Learning small junta functions essentially corresponds to distinguishing relevant features from irrelevant ones in collected data. Although this task is both theoretically and practically fundamental in learning theory, an efficient PAC learner has not yet been found even for the uniform example distribution, and this problem is regarded as one of the most central challenges in CoLT since it was posed by Blum and Langley (1997).

Our first result is to show the heurPAC learnability for junta functions on the uniform distribution. This shows one advantage of our average-case relaxation for a natural learning problem unless a PAC learner for junta functions is developed.

Theorem 3 *$k(n)$-junta functions are heurPAC learnable on the uniform distribution for any $k(n) = \Omega(\log n)$.*

2. They originally considered weaker learnability for infinitely many input sizes to get a standard one-way function. Their result can be extended trivially to a usual learning setting for any input size at the expense of a security condition in cryptography, in particular, "sufficiently large" condition.

The above result guarantees a polynomial-time learner for an input size n if $k(n) = \Theta(\log n)$. In the case where $k(n) = \omega(\log n)$, the length $s(n)$ of representation for $k(n)$-junta functions is greater than $2^{k(n)} = n^{\omega(1)}$. Therefore, the above result does not indicate a polynomial-time learner in n because a heurPAC learner is allowed to work in polynomial-time in $s(n)$. Further, our result does not work in a case where $k(n) = o(\log n)$. In average-case learning, in general, a successful learner does not always work for its subclass, not as in PAC learning. This is because such a subclass could be a hard-core class consisting of hard functions for the learner. Thus, the above result on junta functions is also incomparable with results on its superclasses such as DTs and DNFs.

Therefore, several questions on heurPAC learning such as learnability for DNFs, DTs, or junta functions on other example distributions are still open at the moment. To answer these questions, we may require a deeper understanding of such classes or new learning strategies, and finding them could take several decades as other notorious open problems in CoLT. As a second result, however, we claim that even such a situation where heurPAC learnability has not been revealed is still meaningful. We will see this advantage of the heurPAC learning framework by showing that the hardness of heurPAC learning is capable of constructing secure cryptographic primitives.

Let us mention some backgrounds. The goal of cryptography is, roughly speaking, to produce protocols for protecting private information against feasible adversaries. Intuitively, by regarding adversaries as learners, learning theory and cryptography have opposite goals, i.e., finding and hiding information. Valiant (1984) showed that a standard cryptographic primitive (i.e., a pseudorandom function) yields the hardness of PAC learning in his pioneer paper. However, the opposite direction from the hardness of learning to cryptography, initiated by Impagliazzo and Levin (1990), is less understood, and an explicit barrier by oracle separation has also been found (Xiao, 2009). To the best of our knowledge, all known constructions of cryptographic primitives based on hardness of learning need "strong" hardness assumptions compared to usual PAC learning such as hardness of average-case weak learning (Blum et al., 1994; Nanashima, 2020), specific learning problem (e.g., LWE) (Regev, 2009), or query learning in subexponential-time (Oliveira and Santhanam, 2017).

Technically, our second result weakens the hardness assumptions employed by Blum et al. (1994); Nanashima (2020) from weak learning for fixed parameters to strong learning for parameters given as input. By contraposition, we strengthen the consequence for a cryptographic adversary to a strong heurPAC learner, which is quite nontrivial at present, as mentioned in Section 1.2.

First, we establish a relation between heurPAC learnability (on all example distributions) and an auxiliary-input one-way function (AIOWF), which is a weaker variant of the standard one-way function introduced by Ostrovsky and Wigderson (1993). Roughly speaking, an auxiliary-input primitive is defined as a family of primitives and has relaxed security conditions that at least one primitive in the family is required to be secure depending on each adversary. For further backgrounds and motivations of studying such primitives, refer to, e.g., the work by Vadhan (2006) and Nanashima (2021). It is worthy to note that employing AIOWF is sufficient to show the hardness of PAC learning (Applebaum et al., 2008). The formal definition is given as follows.

Definition 4 (Auxiliary-input function) *A (polynomial-time computable) auxiliary-input function is a family $f = \{f_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{\ell(|z|)}\}_{z \in \{0,1\}^*}$ which has a polynomial-time evaluation algorithm F such that for any $z \in \{0,1\}^*$ and $x \in \{0,1\}^{n(|z|)}$, $F(z,x)$ outputs $f_z(x)$.*

We may write $n(|z|)$ (resp. $\ell(|z|)$) as n (resp. ℓ) when the dependence of $|z|$ is obvious. For any $n \in \mathbb{N}$, let U_n denote a random variable following the uniform distribution over $\{0,1\}^n$.

Definition 5 (Auxiliary-input one-way function) *We say that an auxiliary-input function $f = \{f_z : \{0,1\}^n \to \{0,1\}^\ell\}_{z \in \{0,1\}^*}$ is an auxiliary-input one-way function (AIOWF) if for any polynomial p and polynomial-time randomized algorithm A, there exists an infinite subset $Z_A \subseteq \{0,1\}^*$ such that for any $z \in Z_A$,*

$$\Pr_{A,U_n}\left[A(z, f_z(U_n)) \in f_z^{-1}(f_z(U_n))\right] < \frac{1}{p(|z|)}.$$

Then our main theorem is stated as follows.

Theorem 6 *Let $\mathscr{C}$ be any concept class evaluated in polynomial-time. If $\mathscr{C}$ is not heurPAC learnable, then AIOWF exists.*

Further, if we assume a hardness of learnability on the uniform distribution, we can remove auxiliary-input from the above result. Strictly speaking, for a heurPAC learner which works on any parameters, we need to consider a cryptographic primitive that is secure "infinitely-often" on security parameter n, which is defined formally as follows:

Definition 7 (Infinitely-often one-way function) *Let $s, \ell : \mathbb{N} \to \mathbb{N}$ be polynomials. We say that a function $f = \{f_n : \{0,1\}^{s(n)} \to \{0,1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is an infinitely-often one-way function (io-OWF) if for any polynomial p and polynomial-time randomized algorithm A, f satisfies the following condition for infinitely many $n \in \mathbb{N}$,*

$$\Pr_{A,U_{\ell(n)}}\left[A(1^n, f_n(U_{\ell(n)})) \in f_n^{-1}(f_n(U_{\ell(n)}))\right] < \frac{1}{p(n)}.$$

Then we can show the following theorem.

Theorem 8 *Let $\mathscr{C}$ be any concept class evaluated in polynomial-time. If $\mathscr{C}$ is not heurPAC learnable on the uniform distribution, then io-OWF exists.*

The above is extended to a case of any fixed polynomial-time sampleable[3] example distribution. This can be roughly shown by regarding the sampling algorithm as a part of the evaluation for the concept class.

Corollary 9 *Let $\mathscr{C}$ be a polynomial-time evaluated class and D be a polynomial-time sampleable distribution. If $\mathscr{C}$ is not heurPAC learnable on D, then io-OWF exists.*

Additionally, the opposite direction from a one-way function to the hardness of heurPAC learning also holds, which is based on the famous work by Håstad et al. (1999); Goldreich et al. (1986) and a similar observation by Valiant (1984); Applebaum et al. (2008). Thus, we have characterizations of the existence of AIOWF and io-OWF based on heurPAC learnability. To avoid redundancy, we assume efficiency of evaluations for concept classes in Corollaries 10– 12.

Corollary 10 *The existence of the following notions is equivalent:*

1. AIOWF;

3. We say that $D = \{D_n\}_{n \in \mathbb{N}}$ is polynomial-time sampleable if there exists a polynomial-time randomized algorithm M such that the distribution on outputs of $M(1^n)$ is statistically identical to D_n for each $n \in \mathbb{N}$.

2. *a concept class not heurPAC learnable; and*

3. *a concept class not weakly*[4] *heurPAC learnable.*

Corollary 11 *The existence of the following notions is equivalent:*

1. *io-OWF;*

2. *a concept class not heurPAC learnable on the uniform distribution;*

3. *a concept class not heurPAC learnable with membership queries on the uniform distribution;*

4. *a concept class not weakly heurPAC learnable on the uniform distribution; and*

5. *a concept class not weakly heurPAC learnable with membership queries on the uniform distribution.*

It is worthy to note that Corollary 10 yields several new insights into the hardness of PAC learning. For instance, computational zero-knowledge in cryptography is characterized by AI cryptography (Vadhan, 2006). Thus, our result provides a closer relationship between the average-case hardness of learning and the hardness of obtaining knowledge in cryptography. Even within learning theory, this result implies that if *worst-case* PAC learning *on the uniform distribution* is easy, then *average-case* PAC learning *on any distribution* is also easy because the former is sufficient to break any AI cryptography, as observed by Applebaum et al. (2008). Such a relationship seems quite non-trivial and fundamental knowledge on two worst-case requirements in PAC learning (on target functions and example distributions), which had not been known before.

Theorems 6 and 8 also hold even for a nonuniform computational model by the same proof, where the learner is given additional advice of length $\mathsf{poly}(n, s(n), \epsilon, \delta, \eta)$ for input $(n, s(n), \epsilon, \delta, \eta)$[5]. Then under the Universality Conjecture, recently introduced by Santhanam (2020), we have the following relation between original PAC learning and heurPAC learning.

Corollary 12 *Under the Universality Conjecture, the existence of the following notions is equivalent in the nonuniform setting:*

1. *a concept class not heurPAC learnable;*

2. *a concept class not PAC learnable with membership queries on the uniform distribution; and*

3. *a concept class not heurPAC learnable on the uniform distribution.*

It should be noted that Corollary 12 is regarded as a certain equivalence between worst-case learnability and average-case learnability, which is remarkably nontrivial at present. Thus, our work also enhances the importance of the study on the Universality Conjecture and the minimum circuit size problem (MCSP). For further details, see the original paper (Santhanam, 2020) and related work on MCSP.

4. In weak heurPAC learning, parameters ϵ, δ, and η are fixed in advance to some functions $\epsilon = 1/2 - 1/\mathsf{poly}(n, s(n))$, $\eta = 1 - 1/\mathsf{poly}(n, s(n))$, and $\delta = 1 - 1/\mathsf{poly}(n, s(n))$. For the detail, see also Theorem 22.
5. This model is equivalent to the usual nonuniform model when the input $(n, s(n), \epsilon, \delta, \eta)$ is given as a 5-tuple of unary strings, because there are at most $\mathsf{poly}(N)$ ways to separate input of length N into 5-tuple.

3. Discussion and Future Directions

In this paper, we introduced a heurPAC learning model as a standard model to discuss average-case learnability. Then we showed (1) a new positive result for $\Omega(\log n)$-junta functions and (2) weak cryptographic primitives based on the hardness of heurPAC learning.

The result (1) shows hope that an infeasible task in PAC learning could turn to be feasible in heurPAC learning. Further, the result (2) provides us a win-win "learning vs. cryptography" paradigm, which is the main advantage of the heurPAC learning model. In the real world, there are many concepts we may wish to learn, but in general, finding efficient learners for such classes might require several novel techniques. In heurPAC learning, however, we can use such classes first as a secure cryptographic primitive. Of course, such a tentative primitive might be broken in the future as current cryptosystems, but then the adversary will immediately yield a heuristic learner for the original class as desired at first. Thus, *any* concept class provides an application at any stage as a touchstone for heuristic learning or a cryptographic primitive. This paradigm is quite fundamental and could be extended to other computational models such as quantum computing in future work.

Unfortunately, the "learning vs. cryptography" paradigm brought by this work is imperfect in several senses. For a heurPAC learner which works on any example distribution, our work produces only an auxiliary-input primitive, which is much weaker than a standard primitive. Even if we fix an example distribution to a specific polynomial-time sampleable distribution, the consequence is enhanced only up to the "infinitely-often" security. Thus, as a primary future direction, we suggest improving this paradigm toward the standard one-way function. This challenge is rephrased by a term in the work (Impagliazzo, 1995) as follows: *"Is it possible to change Pessiland into dreamland for learners?"* In this work, we sowed a seed by introducing heurPAC learning toward such *terraforming of Pessiland*.

The heurPAC learning model also offers a good lens to review previous work on CoLT and provides several fascinating questions, for instance, heuristic learnability for other natural classes or richer learning frameworks such as agnostic learning. For such direction, the argument on the result (1) might help lay out a proof plan. Additionally, even for the classes known to be PAC learnable, the learning cost could decrease in heurPAC learning. We also believe that such a study on heurPAC learnability will bring further interrelationship between learning theory and broad fields in the theory of computing such as algorithm theory, complexity theory, and cryptography.

4. Overview of Techniques

We present a high-level overview of ideas to prove Theorems 3, 6, and 8. In this section, we may ignore arguments on confidence δ and quantitative statements to focus only on essential ideas.

4.1. HeurPAC Learning Junta Functions

We first specify a learning strategy and then introduce the key concept classifying junta functions on the level of hardness of learning. Technically, our contribution is the latter part of analyzing hard-to-learn functions, and the learning strategy indeed follows a simple Fourier-based algorithm, introduced by Mossel et al. (2004). In fact, this algorithm learns junta functions in the exact sense, i.e., $\epsilon = 0$.

Now, we review the Fourier-based algorithm for learning junta functions. Note that a polynomial-time learning algorithm is allowed to work in $\mathsf{poly}(n, 2^k)$-time for k-junta functions because the length of representation must be larger than 2^k. To learn junta functions on the uniform distribution, the essential task is to find relevant coordinates. This is because if a learner knows all relevant coordinates, then by reading $\mathsf{poly}(n, 2^k)$ examples, the learner can determine each value in the truth table of the base function with high probability. Because the number of such entries is 2^k, the additional running time to identify the target function from information on relevant coordinates is at most $\mathsf{poly}(n, 2^k)$.

To find relevant coordinates, we apply Fourier properties of junta functions as employed by Mossel et al. (2004). For a function $f : \{0,1\}^n \to \{0,1\}$, a Fourier coefficient $\widehat{f}(\alpha)$ on $\alpha \in \{0,1\}^n$ is defined by $\widehat{f}(\alpha) = \mathrm{E}_x[(-1)^{f(x)+\chi_\alpha(x)}]$, where $\chi_\alpha(x) = x_1\alpha_1 \oplus \cdots \oplus x_n\alpha_n$ denotes a linear function. Refer to textbook (O'Donnell, 2014) for further backgrounds on Fourier analysis. We also use the notation $|\alpha| = \#\{i : \alpha_i \neq 0\}$ for any $\alpha \in \{0,1\}^*$. Then any k-junta function $f : \{0,1\}^n \to \{0,1\}$ has the following crucial properties:

1. if $\widehat{f}(\alpha) \neq 0$, then all coordinates $i \in \{1, \ldots, n\}$ satisfying $\alpha_i \neq 0$ are relevant;
2. for any relevant coordinate i, there exists $\alpha \in \{0,1\}^n$ such that $\alpha_i \neq 0$, $\widehat{f}(\alpha) \neq 0$, and $|\alpha| \leq k$;
3. for any $\alpha \in \{0,1\}^n$, $\widehat{f}(\alpha)$ is computed with access to $\mathrm{EX}(f, U_n)$ in time $\mathsf{poly}(n, 2^k)$ with high probability (we abuse the notation U_n to refer to the uniform distribution over $\{0,1\}^n$).

Therefore, for each target k-junta function $f : \{0,1\}^n \to \{0,1\}$, we can find all relevant coordinates by computing a coefficient $\widehat{f}(\alpha)$ for each α with $1 \leq |\alpha| \leq k$ and identifying all coordinates i satisfying $\alpha_i \neq 0$ when $\widehat{f}(\alpha) \neq 0$. However, since the number of such α is $\binom{n}{1} + \cdots + \binom{n}{k}$, it takes superpolynomial-time when $k = \omega_n(1)$ holds.

Now, we introduce a strategy for heurPAC learning. To reduce the above running time, we set a threshold m $(\leq k)$ to limit the scope of searching relevant coordinates according to a heuristic parameter η. Then we compute $\widehat{f}(\alpha)$ for all α satisfying $|\alpha| \leq m$ (instead of k) as above and find a subset I of relevant coordinates. Note that, for any restriction ρ on a set of at most k in size, we can simulate an example oracle for f restricted by ρ (denoted by $f|_\rho$) in time $\mathsf{poly}(n, 2^k)$ with high probability by querying $\mathrm{EX}(f, U_n)$ until it returns an example consistent with ρ. Thus, we restrict f by all of at most 2^k restrictions ρ on I, find new relevant coordinates by computing $\widehat{f|_\rho}(\alpha)$ for all α with $|\alpha| \leq m$ again, and put them into I. We repeat these processes as the main loop until I gets unchanged. Because f has at most k relevant coordinates, the main loop will stop within k times.

The above algorithm does not always find all relevant coordinates; particularly, in the case where all restricted functions become m-th order correlation-immune (introduced by Siegenthaler, 1984) at some stage, i.e., all Fourier coefficients are zero on α with $1 \leq |\alpha| \leq m$. Although an upper bound on correlation-immune functions was proposed by Schneider (1997), this bound is insufficient to guarantee the performance of the above algorithm because there are functions that turned into correlation-immune by the above procedure. Therefore, we need the following notion to capture such "bad" functions more directly.

Definition 13 (Elusive function) *For any $n, m \in \mathbb{N}$ with $m \leq n$, we say that a function $f : \{0,1\}^n \to \{0,1\}$ is m-elusive if a subset $I \subseteq \{1, \ldots, n\}$ yielded by the following procedure does not correspond to a set of relevant coordinates of f.*

(Procedure.) Repeat the following operation to $I := \emptyset$: add simultaneously all $i \in \{1,\ldots,n\}$ satisfying the condition that there exist a restriction ρ on I and $\alpha \in \{0,1\}^{n-|I|}$ such that $\alpha_i \neq 0$, $|\alpha| \leq m$, and $\widehat{f|_\rho}(\alpha) \neq 0$. If I gets unchanged, then output I.

It can be easily seen that any "bad" function must have an m-elusive base function, so we will bound the fraction $\eta_{k,m}$ of m-elusive functions in k-input base functions. Because an elusive function is defined by the above adaptive procedure, we cannot apply the enumerating technique by Schneider (1997) directly. Thus, we herein take a different approach for the upper bound on elusive functions, which was originally motivated by the technique on error-correcting codes.

Our idea is quite simple: we regard a truth table of m-elusive function as a code of length 2^k and then give a lower bound on the minimum distance. Technically, we will show that the minimum distance of m-elusive functions is greater than $m+1$ based on Fourier analysis and properties of correlation-immune functions (refer to Appendix C.1). Thus, each m-elusive function has at least a distinct Hamming ball of volume roughly $V \approx (2^{k+1}/m)^{m/2}$, and $\eta_{k,m}$ is bounded above by $1/V$.

In the case of $k = \Omega(\log n)$, the above upper bound is enough to show that $n^m \leq \mathsf{poly}(\eta_{k,m}^{-1}, 2^k)$ for sufficiently large n. Notice that the above learner with a threshold m succeeds in finding relevant coordinates for all junta functions whose base function is not m-elusive in time at most $n^m \cdot \mathsf{poly}(n, 2^k)$. Thus, by choosing a minimum threshold m satisfying the condition $\eta_{k,m} \leq \eta$, the learner satisfies the condition on a heuristic parameter. Since $\eta < \eta_{k,m-1}$ holds for such m, the running time is roughly bounded above by $n^m \cdot \mathsf{poly}(n, 2^k) \leq n^{m-1} \cdot \mathsf{poly}(n, 2^k) \leq \mathsf{poly}(n, 2^k, \eta_{k,m-1}^{-1}) \leq \mathsf{poly}(n, 2^k, \eta^{-1})$ as desired.

4.2. Weak Cryptography Based on HeurPAC Learnability

To demonstrate Theorems 6 and 8, we combine techniques employed by Blum et al. (1994); Nanashima (2020) for weak learning with a hardness amplification by Levin's version of XOR lemma (Fact 3) introduced by Levin (1987). For a target function f, define another function $f^{\oplus}$ by $f^{\oplus}(x^{(1)}, \ldots, x^{(d)}) = \bigoplus_i f(x^{(i)})$ for a proper $d \in \mathbb{N}$. From the learning perspective, the XOR lemma provides a boosting method translating a weak hypothesis for $f^{\oplus}$ into a strong hypothesis for f under a distribution D on input with access to $\mathrm{EX}(f, D)$. Although a similar method was used for boosting in PAC learning by Boneh and Lipton (1993), we need additional ideas for average-case learning.

In the work by Blum et al. (1994); Nanashima (2020), roughly speaking, they constructed a pseudorandom generator based on the hardness of weak learnability for a randomly selected f. Note that a pseudorandom generator is another cryptographic primitive whose output cannot be distinguished from truly random strings by polynomial-time adversaries with a non-negligible probability, and it must be also a one-way function (for further details, refer to Section A.1). The output of their generator corresponds to labels in examples, i.e., values of a target function f, and the representation for f is regarded as a random seed for the generator. To prove the claim by contraposition, they employed Yao's next-bit generator (Yao, 1982) to translate an adversary for the generator into a weak hypothesis for f.

To enhance the above result to strong learnability, the first attempt is to let an algorithm learn $f^{\oplus}$ instead of f. Notice that we can easily simulate examples for $f^{\oplus}$ just by XORing several examples for f. Then, we can translate an adversary for the resulting generator into a weak hypothesis for $f^{\oplus}$ and construct a strong hypothesis for the original target function f by applying the XOR lemma. A similar strategy was also used in the context of "hardness vs. randomness" (e.g., Nisan and Wigderson, 1994). However, this is insufficient for heurPAC learning because it reduces the accuracy error

ϵ but does not have any effect on the heuristic error η at all. For instance, if two-thirds of target functions in class $\mathscr{C}$ are constant, the output of the above generator is trivially distinguishable from a random string, but there is no guarantee that $\mathscr{C}$ is heurPAC learnable because the other functions could be extremely hard to learn. In other words, we need a two-dimensional boosting method not only on the accuracy parameter ϵ but also on the heuristic parameter η for heurPAC learning.

At this point, we present our new construction of the pseudorandom generator. The central idea is intuitive: taking additional XOR among various target functions selected independently and identically at random in the class. In other words, we construct the above generator in such a way that each bit corresponds to a value of a new target function $f^{\oplus\oplus} : \{0,1\}^{nd} \to \{0,1\}$ defined by

$$f^{\oplus\oplus}(x^{(1)}, \dots, x^{(d)}) := \bigoplus_{i=1}^{v} f^{(i)\oplus}(x^{(1)}, \dots, x^{(d)}) = \bigoplus_{i=1}^{v} \bigoplus_{j=1}^{d} f^{(i)}(x^{(j)}),$$

where $x^{(j)} \in \{0,1\}^n$ for each j, $f^{(i)}$ denotes a random target function for each i, and v and d (which stand for "variety" and "duplication," respectively) are determined depending on ϵ and η.

Herein, we present a brief explanation of how to translate an adversary A for the above generator into a strong hypothesis for all but η fraction of target functions. First, we introduce the notion of a "bad" function for A as a function that reduces the advantage of A drastically under the condition that $f^{(i)} \equiv f$ for some $i \in \{1, \dots, v\}$. Then, we show that the fraction η_B of "bad" functions is not so large, which follows from the fact that the probability that all selected functions are not "bad" is quite small (less than $(1-\eta_B)^v$). In fact, by proper choices of v and the notion of "bad," η_B is bounded above by η. In this case, it is enough to learn all target functions that are not "bad".

If a target function f is not "bad," then there exists at least one position $i^* \in \{1, \dots, v\}$ such that the advantage of A does not decrease so much even if $f^{(i^*)} \equiv f$. In such a scenario, a learner can guess i^* at random, select other functions $f^{(i)}$ for each $i \in \{1, \dots, v\} \setminus \{i^*\}$ by itself, simulate an example oracle for $f^{\oplus\oplus}$, and translate A into a weak hypothesis for $f^{\oplus\oplus}$ by using Yao's next bit generator. Because the learner knows all functions $f^{(i)}$ except f, the learner can also construct a weak hypothesis for $f^{\oplus}$ and a strong hypothesis for the original target f by using the XOR lemma.

In fact, there are several flaws including formality in the above sketch. The most critical one is that we cannot fix d and v in advance because they depend on the input, especially, ϵ and η. Because the learner itself relies on the adversary and its advantage, the above heurPAC learner is not well-defined. To resolve this, we must first fix some pseudorandom generator G, independent of v and d, assume an arbitrary adversary A for G, and then construct a learner based on G and A. For such a prior generator G, we take different approaches in the cases of AIOWF and io-OWF.

In the case of AIOWF, an adversary must break a primitive for any auxiliary-input; thus, we can use non-determinism on descriptions of the primitives. We fix the prior generator G to the natural universal generator composed of the standard evaluation algorithms for n^2-sized circuits, where the description of circuits is regarded as auxiliary-input. For any adversary A for G, we can construct a heurPAC learner L as follows: L determines d and m based on its input and A properly, constructs an auxiliary-input analog of pseudorandom generator G' by applying the dualization technique (Nanashima, 2020) in the above construction (i.e., L regards a target function f as input for G' and input for f as auxiliary-input for G'), and reduces the circuit complexity of G' by using a standard padding technique to fit the form of G. Then, L constructs an adversary for G' based on A and a strong hypothesis for the original target function as the above argument.

In the case of io-OWF, we cannot use the above non-determinism, so we need to construct one specific generator. For each input size n, representation size s, v, and d, we map each $(n, s, v, d) \in \mathbb{N}^4$ to a different security parameter $N \in \mathbb{N}$ by using a reasonable pairing function. We define the prior generator G by using the above construction with respect to (n, s, v, d) reconstructed from a security parameter N, where the input for a target function is regarded as a part of seeds for G and outputted directly to avoid reducing the stretch of G as employed by Blum et al. (1994). Then, for any adversary A for G, we can construct a heurPAC learner on the uniform distribution as in the case of AIOWF. Note that for the learner working in all settings of (n, ϵ, η), the adversary A must work for all corresponding security parameters N, which is the main source of the "infinitely often" security in Theorem 8.

Acknowledgments

This work was supported by JST, ACT-X Grant Number JPMJAX190M, Japan. The author thanks Toshiya Itoh and the anonymous reviewers for many helpful comments.

References

H. Aizenstein and L. Pitt. On the Learnability of Disjunctive Normal Form Formulas. *Mach. Learn.*, 19(3):183–208, June 1995.

D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

D. Angluin and D. Chen. Learning a Random DFA from Uniform Strings and State Information. In *Proceedings of the 26th International Conference on Algorithmic Learning Theory*, ALT'15, pages 119–133. Springer International Publishing, 2015.

D. Angluin and M. Kharitonov. When Won't Membership Queries Help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995.

B. Applebaum, B. Barak, and D. Xiao. On Basing Lower-Bounds for Learning on Worst-Case Assumptions. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'08, pages 211–220, 2008.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1):245–271, 1997.

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

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

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis Dimension. *J. ACM*, 36(4):929–965, October 1989.

D. Boneh and R. Lipton. Amplification of Weak Learning under the Uniform Distribution. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pages 347–351, New York, NY, USA, 1993. ACM.

G. Cantor. Ein Beitrag zur Mannigfaltigkeitslehre. *Journal für die reine und angewandte Mathematik*, (84):242–258, 1878.

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

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

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

O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. *J. ACM*, 33(4): 792–807, August 1986. ISSN 0004-5411.

O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR-Lemma. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation, Springer Berlin Heidelberg*, pages 273–301, 2011.

J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from Any One-way Function. *SIAM J. Comput.*, 28(4):1364–1396, March 1999.

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, pages 42–55, 1988.

D. Helmbold, R. Sloan, and M. K. Warmuth. Learning Integer Lattices. *SIAM Journal on Computing*, 21(2):240–266, 1992.

R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of IEEE Tenth Annual Conference on Structure in Complexity Theory*, pages 134–147, 1995.

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

R. Impagliazzo and M. Luby. One-way Functions Are Essential for Complexity Based Cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 230–235, 1989.

J. Jackson. An Efficient Membership-Query Algorithm for Learning DNF with Respect to the Uniform Distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

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

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

M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, January 1994. ISSN 0004-5411.

M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, USA, 1994. ISBN 0-262-11193-4.

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

L. Levin. One Way Functions and Pseudorandom Generators. *Combinatorica*, 7(4):357–363, 1987.

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

E. Mossel, R. O'Donnell, and R. Servedio. Learning Functions of k Relevant Variables. *J. Comput. Syst. Sci.*, 69(3):421–434, 2004.

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

M. Nanashima. On Basing Auxiliary-Input Cryptography on NP-Hardness via Nonadaptive Black-Box Reductions. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *LIPIcs*, pages 29:1–29:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

N. Nisan and A. Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

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

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

R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory and Computing Systems*, ISTCS'93, pages 3–17, June 1993.

L. Pitt and M. K. Warmuth. Prediction-preserving Reducibility. *J. Comput. Syst. Sci.*, 41(3):430–467, 1990.

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

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

R. Schapire. The Strength of Weak Learnability. *Mach. Learn.*, 5(2):197–227, 1990.

M. Schneider. A Note on the Construction and Upper Bounds of Correlation-Immune Functions. In *Proceedings of the 6th IMA International Conference on Cryptography and Coding*, pages 295–306. Springer Berlin Heidelberg, 1997.

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

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

T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, 30(5):776–780, 1984.

S. Vadhan. An Unconditional Study of Computational Zero Knowledge. *SIAM J. Comput.*, 36(4): 1160–1214, December 2006.

L. Valiant. A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142, 1984. ISSN 0001-0782. doi: 10.1145/1968.1972.

A. Wigderson. *Mathematics and Computation: A Theory Revolutionizing Technology and Science*. Princeton University Press, 2019. ISBN 9780691192543.

D. Xiao. On basing ZK $\neq$ BPP on the hardness of PAC learning. In *Proceedings of the 24th Conference on Computational Complexity*, CCC'09, pages 304–315, 2009.

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

Organization of Appendices

In appendices, we present formal descriptions based on the sketch in the main text. In Appendix A, we introduce additional preliminaries for the formal arguments. In Appendix B, we present the proof of the basic facts introduced in Section 1. We present the formal argument of Sections 4.1 and 4.2 in Appendices C and D, respectively. In Appendix D.3, we also mention the corollaries presented in Section 2.

Appendix A. Preliminaries

For each $n \in \mathbb{N}$, we define $[n] = \{1, \ldots, n\}$. For any binary string $x \in \{0,1\}^n$ and $k \in [n]$, let $x_{[k]}$ denote a prefix $x_1 \circ \cdots \circ x_k$. For a randomized algorithm A using $r(n)$ random bits on n-bit input, we use $A(x; s)$ to refer to the execution of $A(x)$ with random tape s for $x \in \{0,1\}^n$ and $s \in \{0,1\}^{r(n)}$.

We fix a proper binary encoding for circuits and identify a circuit with its binary encoding. Additionally, we assume that (1) every binary string represents some circuit (by assigning invalid encodings to the trivial circuit $C(x) \equiv 0$); and (2) zero-padding is available. These assumptions allow us to assure that there exists a function $e_s(\cdot)$ such that any n-input circuit of size $s(n)$ has a binary representation of the length $e_s(n) = O(s(n)\log(s(n)))$ for any function $s : \mathbb{N} \to \mathbb{N}$.

For any distribution D on binary strings, we use the notation $x \leftarrow D$ to denote a random sampling x according to D. For a finite set S, we also use the notation $x \leftarrow_u S$ to denote the uniform sampling from S. For $x \in \{0,1\}^*$, let $D(x) \in [0,1]$ be the probability that x is generated according to D. For each $m \in \mathbb{N}$, we also use the notation D^m to refer to the distribution on m-tuple of strings where each element is selected independently and identically according to D. Herein, we assume basic knowledge of probability theory, including union bound, Markov's inequality, and Hoeffding's inequality.

For a function $f : \{0,1\}^n \to \{0,1\}$, we define its weight $w(f)$ by $\#\{x \in \{0,1\}^n : f(x) = 1\}$. For any $d \in \mathbb{N}$ and function $f : \{0,1\}^n \to \{0,1\}$, we define a function $f^{\oplus d} : \{0,1\}^{dn} \to \{0,1\}$ by

$$f^{\oplus d}(x^{(1)}, \ldots, x^{(d)}) = \bigoplus_{i=1}^{d} f(x^{(i)}),$$

where $x^{(i)} \in \{0,1\}^n$ for each $i \in [d]$.

In this paper, we may use $\{-1,1\}$ instead of $\{0,1\}$ to refer to a binary value, where -1 (resp. 1) corresponds to 1 (resp. 0). In such a case, for a Boolean-valued function $f : \{0,1\}^n \to \{-1,1\}$ and $\alpha \in \{0,1\}^n$, a Fourier coefficient of f on α is rewritten simply as $\widehat{f}(\alpha) = \mathrm{E}_x[f(x)\chi_\alpha(x)]$, where $\chi_\alpha(x) = (-1)^{x_1\alpha_1 + \cdots + x_n\alpha_n}$. We will also introduce the following notion of correlation-immune.

Definition 14 (Correlation-immune, Siegenthaler, 1984) *For $n, m \in \mathbb{N}$ ($m \le n$), we say that a function $f : \{0,1\}^n \to \{0,1\}$ is m-th order correlation-immune if $\widehat{f}(\alpha) = 0$ for any $\alpha \in \{0,1\}^n$ with $1 \le |\alpha| \le m$.*

A.1. Weak Cryptography

In this section, we formally introduce other standard cryptographic primitives.

Definition 15 (Pseudorandom generator) *Let $\Sigma = \{0,1\}$ be an alphabet set and $s, \ell : \mathbb{N} \to \mathbb{N}$ be polynomially-related functions. We say that $G = \{G_z : \{0,1\}^{s(|z|)} \to \{0,1\}^{\ell(|z|)}\}_{z \in \Sigma^*}$ is an auxiliary-input pseudorandom generator (AIPRG) if G is polynomial-time computable, $s(n) < \ell(n)$ for any $n \in \mathbb{N}$, and for any polynomial p and polynomial-time randomized algorithm A, there exists an infinite subset $Z_A \subseteq \Sigma^*$ such that for any $z \in Z_A$,*

$$\left|\Pr\left[A(z, G_z(U_{s(|z|)})) = 1\right] - \Pr\left[A(z, U_{\ell(|z|)}) = 1\right]\right| < \frac{1}{p(|z|)}.$$

Further, we say that $G = \{G_z : \{0,1\}^{s(|z|)} \to \{0,1\}^{\ell(|z|)}\}_{z \in \Sigma^*}$[6] *is an infinitely-often pseudorandom generator (io-PRG) if G satisfies the above conditions except for $\Sigma = \{1\}$.*

Definition 16 (Pseudorandom function) *Let $\Sigma = \{0,1\}$ be an alphabet set. We say that $F = \{F_z : \{0,1\}^{|z|} \times \{0,1\}^{|z|} \to \{0,1\}\}_{z \in \Sigma^*}$ is an auxiliary-input pseudorandom function (AIPRF)*

6. We may write $\{G_z\}_{z \in \Sigma^*}$ as $\{G_n\}_{n \in \mathbb{N}}$ in the case where $\Sigma = \{1\}$.

if F is polynomial-time computable, and for any polynomial p and polynomial-time randomized oracle machine $A^?$, there exists an infinite subset $Z_A \subseteq \Sigma^*$ such that for every $z \in Z_A$,

$$\left| \Pr_{A,u\sim\{0,1\}^{|z|}} \left[A^{F_z(u,\cdot)}(z) = 1 \right] - \Pr_{A,\phi_{|z|}} \left[A^{\phi_{|z|}(\cdot)}(z) = 1 \right] \right| < \frac{1}{p(|z|)},$$

where $\phi_{|z|} : \{0,1\}^{|z|} \to \{0,1\}$ denotes a truly random function.

Further, we say that $F = \{F_z : \{0,1\}^{s(|z|)} \times \{0,1\}^{m(|z|)} \to \{0,1\}\}_{z\in\Sigma^*}$ is an infinitely-often pseudorandom function (io-PRF) if F satisfies the above conditions except for $\Sigma = \{1\}$.

Definition 17 (Distributional one-way function) *Let $s, \ell : \mathbb{N} \to \mathbb{N}$ be polynomials. We say that a function $f = \{f_n : \{0,1\}^{s(n)} \to \{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is an infinitely-often distributional one-way function (io-DistOWF) if there exists a polynomial p such that for all polynomial-time randomized algorithms A, the statistical distance between $\left(A(1^n, f_n(U_{s(n)})), f_n(U_{s(n)})\right)$ and $\left(U_{s(n)}, f_n(U_{s(n)})\right)$ is greater than $1/p(n)$ for infinitely many $n \in \mathbb{N}$.*

Fact 1 (Goldreich et al., 1986; Håstad et al., 1999; Impagliazzo and Luby, 1989) *The followings 1–3 are equivalent:*

1. *AIOWF exists;*
2. *AIPRG exists; and*
3. *AIPRF exists.*

Further, the followings 4–7 are equivalent:

4. *io-OWF exists;*
5. *io-PRG exists;*
6. *io-PRF exists; and*
7. *io-DistOWF exists.*

Appendix B. Basic Facts on HeurPAC Learning

In this section, we formally state and prove basic facts for heurPAC learning in Section 1. In fact, proofs in this section are somewhat fundamental, so the reader may skip them.

B.1. Natural Representation of Target Functions

We show that our formulation in Definition 1 captures several natural distributions on target functions. Generally speaking, Theorem 18 shows that we can find a representation for our formulation whenever such a natural distribution is approximately simulated by an efficient algorithm.

Theorem 18 *Let $s : \mathbb{N} \to \mathbb{N}$ be a function and $\mathscr{C} = \{\mathscr{C}_n\}_{n\in\mathbb{N}}$ be a concept class, where $\mathscr{C}_n \subseteq \{f : \{0,1\}^n \to \{0,1\}\}$ and every $f \in \mathscr{C}_n$ has representation with a length of at most $s(n)$ for each $n \in \mathbb{N}$. Let $R = \{R_n\}_{n\in\mathbb{N}}$ be any (natural) family of distributions, where R_n is a distribution on $\mathscr{C}_n$. Suppose there exists a randomized machine M which approximately simulates R in the following sense:*

- *for any $n \in \mathbb{N}$, $M(n, s(n))$ outputs a target function $f \in \mathscr{C}_n$ or a special symbol $\perp$ (which represents "abortion") in time $p(n, s(n))$ where p represents a polynomial;*
- *there exist polynomials p_u and p_ℓ such that for any $n \in \mathbb{N}$ and $f \in \mathscr{C}_n$,*

$$\frac{1}{p_\ell(n, s(n))} \cdot R_n(f) \le \Pr_M [M(n, s(n)) \text{ outputs } f] \le p_u(n, s(n)) \cdot R_n(f).$$

Let $C' : \{0,1\}^{p(n,s(n))} \times \{0,1\}^n \to \{0,1\}$ be an evaluation defined by

$$C'(r, x) = \begin{cases} f(x) & \text{if } f \leftarrow M(n, s(n); r) \\ 1 & \text{if } \perp \leftarrow M(n, s(n); r). \end{cases}$$

A concept class $\mathscr{C}'$ defined by C' is heurPAC learnable iff $\mathscr{C}$ is heurPAC learnable with respect to R, i.e., there exists a learner L satisfying the same conditions in Definition 1 except that a target function is selected according to R_n for each input size n. The same relation also holds in heurPAC learning on a fixed example distribution.

Further, if $\mathscr{C}$ is evaluated in polynomial-time, then $\mathscr{C}'$ is also evaluated in polynomial-time.

Proof First, we assume that there exists a heurPAC learner L for $\mathscr{C}$ with respect to R. Then, we can construct a heurPAC learner L' for $\mathscr{C}'$ as follows: on input $(n, s(n), \epsilon, \delta, \eta)$, L' first examines whether a constant function $\mathbf{1}$ is ϵ-close to the target function with confidence error δ by looking at $O(\epsilon^{-2} \log \delta^{-1})$ examples. Hence, we can assume that the target function is not a constant function $\mathbf{1}$. Then L' executes $L(n, s(n), \epsilon, \delta, \frac{\eta}{p_u(n,s(n))})$ for the same example oracle and outputs L's hypothesis. Obviously, L' halts in time $\mathsf{poly}(n, s(n), \epsilon^{-1}, \delta^{-1}, \eta^{-1})$.

Let B a set of "bad" functions L fails in learning in the above setting. By the choice of the heuristic parameter, we have that $\sum_{f \in B} R_n(f) \le \frac{\eta}{p_u(n,s(n))}$. Notice that L' fails in learning only if L also fails. Over the choice of a concept in $\mathscr{C}'$ (i.e., randomness for M), such a "bad" function is selected with probability at most

$$\sum_{f \in B} \Pr[M \text{ outputs } f] \le \sum_{f \in B} R_n(f) \cdot p_u(n, s(n)) \le \eta.$$

In the opposite direction, we assume that there exists a heurPAC learner L' for $\mathscr{C}'$. W.l.o.g., we can assume that L' succeeds in learning a constant function $\mathbf{1}$, which is the case where M outputs $\perp$. Then, we can construct a heurPAC learner L for $\mathscr{C}$ as follows: On input $(n, s(n), \epsilon, \delta, \eta)$, L executes $L'(n, s(n), \epsilon, \delta, \frac{\eta}{p_\ell(n,s(n))})$ with the same example oracle and outputs L''s hypothesis. Obviously, L halts in time $\mathsf{poly}(n, s(n), \epsilon^{-1}, \delta^{-1}, \eta^{-1})$.

Let B be a set of "bad" functions L' fails in learning in the above setting. Notice that L fails in learning only if L' also fails, i.e., a target function is contained in B. By the choice of the heuristic parameter, we have that

$$\frac{\eta}{p_\ell(n, s(n))} \ge \sum_{f \in B} \Pr[M \text{ outputs } f] \ge \sum_{f \in B} \frac{R_n(f)}{p_\ell(n, s(n))}.$$

This implies that $\sum_{f \in B} R_n(f) \le \eta$. This proof also holds even in learning on a fixed distribution.

If $\mathscr{C}$ is evaluated in polynomial-time, then $\mathscr{C}'$ is also evaluated in polynomial-time because executions of $f \leftarrow M(n, s(n); r)$ and $f(x)$ halt in polynomial-time. ■

B.2. Equivalence to Average-Case Prediction Model

In this part, we extend the equivalence property between the capabilities of the PAC learning model and the prediction model (Haussler et al., 1988) to average-case learnability. First, we introduce a natural average-case extension of the prediction model (Pitt and Warmuth, 1990) as follows.

Definition 19 (Average-case prediction) *Let $s : \mathbb{N} \to \mathbb{N}$ and $\mathscr{C}$ be an $s(n)$-represented concept class. We say that $\mathscr{C}$ is polynomially predictable on average if there exists a polynomial $m := m(n, s(n), \epsilon^{-1})$ and a randomized algorithm P such that for any $n \in \mathbb{N}$, $\epsilon \in (0, 1]$, and distribution D on $\{0, 1\}^n$, P satisfies the following expression:*

$$\Pr_{\substack{P, f \sim \{0,1\}^{s(n)} \\ x^{(1)}, \ldots, x^{(m)}, x^* \leftarrow D}} \left[P\left(n, s(n), \epsilon, \left(x^{(1)}, f(x^{(1)})\right), \ldots, \left(x^{(m)}, f(x^{(m)})\right), x^*\right) = f(x^*) \right] \geq 1 - \epsilon$$

and halts in polynomial-time in n, $s(n)$, and ϵ^{-1}.

Note that we refer to x^* in the above expression as a challenge. First, we show the following equivalence.

Theorem 20 *A concept class $\mathscr{C}$ is heurPAC learnable iff $\mathscr{C}$ is polynomially predictable on average.*

Proof First, we assume that L is a heurPAC learner for $\mathscr{C}$. We construct an average-case predictor P as follows: on input $n, s(n) \in \mathbb{N}$ and the error parameter ϵ, P executes L in the settings of an accuracy parameter $\epsilon/3$, a confidence parameter $\epsilon/3$, and a heuristic parameter $\epsilon/3$, and then P outputs the value of L's hypothesis on the challenge as P's prediction. Because L halts in time $\mathsf{poly}(n, s(n), \epsilon^{-1})$, it is enough to take $\mathsf{poly}(n, s(n), \epsilon^{-1})$ examples. Thus, P also halts in time $\mathsf{poly}(n, s(n), \epsilon^{-1})$.

Notice that if P fails in the prediction, then at least one of the following three events occurs: (1) a target function is hard to learn for L; (2) a target function is not a hard instance, but L fails in producing a good hypothesis $\epsilon/3$-close to the target function; or (3) L succeeds in producing such a good hypothesis h, but h is inconsistent with the target function on the challenge. By the choice of parameters for L, each event will occur with probability at most $\epsilon/3$. Thus, by the union bound, the failure probability of P is at most $3 \cdot \epsilon/3 = \epsilon$.

In the opposite direction, we assume that an algorithm P polynomially predicts $\mathscr{C}$ on average. We construct a heurPAC learner L as follows: on input $(n, s(n), \epsilon, \delta, \eta)$, L outputs a hypothesis $h_{X^m,r}$ defined by

$$h_{X^m,r}(x) = P(n, s(n), \epsilon\delta\eta, X^m, x; r),$$

where $X^m = ((x^{(1)}, b^{(1)}), \ldots, (x^{(m)}, b^{(m)}))$ denotes L's example set of size $m = m(n, s(n), (\epsilon\delta\eta)^{-1})$, and $r \in \{0, 1\}^{\mathsf{poly}(n, s(n), 1/\epsilon\delta\eta)}$ denotes randomness for P selected by L.

By the correctness of P, we have that

$$\Pr_{f, r, X^m, x}[h_{X^m,r}(x) \neq f(x)] \leq \epsilon\delta\eta,$$

where each example in X^m and x are selected according to an example distribution D.

By Markov's inequality, we obtain:

$$\Pr_f \left[\Pr_{r, X^m, x}[h_{X^m,r}(x) \neq f(x)] \geq \epsilon\delta \right] \leq \eta.$$

By applying Markov's inequality again, we have that for all but η fraction of target functions f,

$$\Pr_{r,X^m}\left[\Pr_{x\leftarrow D}[h_{X^m,r}(x)\neq f(x)]\geq\epsilon\right]\leq\delta.$$

Because P is computable in time $\mathsf{poly}(n,s(n),(\epsilon\delta\eta)^{-1})$, L can output the above hypothesis $h_{X^m,r}$ in time $\mathsf{poly}(n,s(n),\epsilon^{-1},\delta^{-1},\eta^{-1})$ by using the standard encoding. ■

Blum et al. (1994) proposed the following weak version of the average-case prediction model.

Definition 21 (Average-case weak prediction) *Let $s:\mathbb{N}\to\mathbb{N}$ and $\mathscr{C}$ be an $s(n)$-represented concept class. We say that $\mathscr{C}$ is weakly predictable on average if there exist polynomials $m:=m(n,s(n))$ and $p(n,s(n))$ and a randomized algorithm P such that for any $n\in\mathbb{N}$ and distribution D on $\{0,1\}^n$, P satisfies the expression:*

$$\Pr_{\substack{P,f\sim\{0,1\}^{s(n)}\\ x^{(1)},\ldots,x^{(m)},x^*\leftarrow D}}\left[P\left(n,s(n),\left(x^{(1)},f(x^{(1)})\right),\ldots,\left(x^{(m)},f(x^{(m)})\right),x^*\right)=f(x^*)\right]\geq\frac{1}{2}+\frac{1}{p(n,s(n))}$$

and halts in polynomial-time in n and $s(n)$.

Note that we refer to the above function p as an advantage of P. Herein, we present a clear relation between average-case weak predictability and weak heurPAC learnability as follows.

Theorem 22 *For any $s(n)$-represented concept class $\mathscr{C}$, the following statements are equivalent:*

1. *$\mathscr{C}$ is weakly predictable on average and*
2. *$\mathscr{C}$ is heurPAC learnable in time $\mathsf{poly}(n,s(n))$ for fixed parameters ϵ, δ, and η satisfying the expression:*

$$\epsilon\leq\frac{1}{2}-\frac{1}{p_\epsilon(n,s(n))},\ \eta\leq1-\frac{1}{p_\eta(n,s(n))},\ \text{and } \delta\leq1-\frac{1}{p_\delta(n,s(n))},$$

   where p_ϵ, p_η, and p_δ are some polynomials.

The above equivalence also holds even on a fixed example distribution.

Proof (1$\Longrightarrow$2) Let P be a prediction algorithm for $\mathscr{C}$ and $p(n,s(n))$ be its advantage. Then we construct a weak heurPAC learner L as follows: on input n and $s(n)$, L outputs a hypothesis h defined by

$$h_{X^m,r}(x)=P\left(n,s(n),X^m,x;r\right),$$

where X^m denotes L's example set of size m and r denotes randomness for P selected by L. Obviously, L halts in polynomial-time in n and $s(n)$.

Because L executes P in the valid settings for the same target function f and the example distribution D, we have that

$$\Pr_{f,r,X^m,x}[h_{X^m,r}(x)\neq f(x)]\leq\frac{1}{2}-\frac{1}{p(n,s(n))}.$$

By Markov's inequality,

$$\Pr_{f,r,X^m}\left[\Pr_{x\leftarrow D}[h_{X^m,r}(x)\neq f(x)]\geq \frac{1}{2}-\frac{1}{2p(n,s(n))}\right]\leq \frac{\frac{1}{2}-\frac{1}{p(n,s(n))}}{\frac{1}{2}-\frac{1}{2p(n,s(n))}}$$
$$=1-\frac{1}{p(n,s(n))-1}$$
$$\leq 1-\frac{1}{p(n,s(n))}.$$

Let E_{f,r,X^m} denote the above event that $\Pr_x[h_{X^m,r}(x)\neq f(x)]\geq \frac{1}{2}-\frac{1}{2p(n,s(n))}$. By applying Markov's inequality again,

$$\Pr_f\left[\Pr_{r,X^m}[E_{f,r,X^m}]\geq 1-\frac{1}{2p(n,s(n))}\right]\leq \frac{1-\frac{1}{p(n,s(n))}}{1-\frac{1}{2p(n,s(n))}}$$
$$=1-\frac{1}{2p(n,s(n))-1}$$
$$\leq 1-\frac{1}{2p(n,s(n))}.$$

Therefore, the heurPAC learner L achieves the following parameters

$$\epsilon=\frac{1}{2}-\frac{1}{2p(n,s(n))},\ \eta=1-\frac{1}{2p(n,s(n))},\ \text{and } \delta=1-\frac{1}{2p(n,s(n))}.$$

(2$\Longrightarrow$1) Let L be a weak heurPAC learner for $\mathscr{C}$ with parameters ϵ,η, and δ as in Theorem 22. First, we apply the standard repeating and testing technique to reduce the confidence parameter to $2^{-(ns(n)+1)}$ with multiplicative loss of time $\mathsf{poly}(n,s(n))$. For details, refer to (Haussler et al., 1988, Lemma 3.4). Thus, we assume that $\delta=2^{-(ns(n)+1)}$ in this proof.

We construct a weak predictor P for $\mathscr{C}$ as follows: on input n and $s(n)$, P takes enough examples to successfully execute $L(n,s(n))$. If L outputs some hypothesis h, then P estimates the probability p_h that h agrees with the target function under the example distribution within accuracy $\pm 1/2p_\epsilon(n,s(n))$ with probability at least $1-2^{ns(n)+1}$. Note that, by the standard empirical estimation, it is enough to take $M=O(p_\epsilon(n,s(n))^2 ns(n))=\mathsf{poly}(n,s(n))$ examples. If the estimate $\tilde{p}$ is at least $1/2+1/2p_\epsilon(n,s(n))$, then P outputs $h(x^*)$, otherwise outputs a random bit as P's prediction. It can be verified easily that the number of required examples is at most $\mathsf{poly}(n,s(n))$, and P halts in time $\mathsf{poly}(n,s(n))$.

At first we assume that L always succeeds in learning at least $(1-\eta)$ fraction of target functions (i.e., $\delta=0$) and in estimating p_h within error $\pm 1/2p_\epsilon(n,s(n))$ with probability 1. Let f denote a target function. If L outputs a hypothesis h that is ϵ-close to f, then the estimate $\tilde{p}$ satisfies the condition $\tilde{p}\geq(1-\epsilon)-1/2p_\epsilon(n,s(n))\geq 1/2+1/2p_\epsilon(n,s(n))$, so such an h is used for prediction. On the other hand, if a hypothesis h passes the test, then such an h must satisfy the condition $p_h+1/2p_\epsilon(n,s(n))\geq 1/2+1/2p_\epsilon(n,s(n))$, i.e., $p_h\geq 1/2$. Even if h does not pass the test, P makes a prediction at random. Thus, for any target function, P succeeds in predicting with probability at least $1/2$. Therefore, P's success probability is at least

$$\begin{aligned}(1-\eta)(1-\epsilon)+\eta\cdot\frac{1}{2} &\geq (1-\eta)\left(\frac{1}{2}+\frac{1}{p_\epsilon(n,s(n))}\right)+\eta\cdot\frac{1}{2}\\ &\geq \frac{1}{2}+\frac{1-\eta}{p_\epsilon(n,s(n))}\\ &\geq \frac{1}{2}+\frac{1}{p_\eta(n,s(n))p_\epsilon(n,s(n))}.\end{aligned}$$

Even if we take the confidence errors of L and the estimation into account, the error is bounded above by $2^{-ns(n)}$, which is negligible in n and $s(n)$. Therefore, the above P still succeeds in predicting with probability at least $1/2+1/\mathsf{poly}(n,s(n))$.

Note that the above argument does not depend on a specific example distribution, so it holds even in learning on a fixed distribution. ■

Appendix C. Heuristic PAC Learning $\Omega(\log n)$-Junta Functions

In this section, we show the following learnability for junta functions based on the sketch in Section 4.1. For convenience, we apply $\{-1,1\}$ to a binary value instead of $\{0,1\}$ in Appendix C.

Theorem 3 *For any $k(n)=\Omega(\log n)$, $k(n)$-junta functions are heurPAC learnable on the uniform distribution.*

As stated in Section 4.1, the essential task for the above result is to find all relevant coordinates of the target junta function. Thus, we only focus on the essential task in this section. For the sake of simplicity, we also assume that a learner with access to $\mathrm{EX}(f,U_n)$ can simulate $\mathrm{EX}(f|_\rho,U_{n-|I|})$ for any subset $I\subseteq[n]$ of size at most k and restriction ρ on I in time $\mathsf{poly}(n,2^k)$.

Before presenting the main algorithm, we will first introduce a subroutine **Find** as Algorithm 1. When **Find** is given oracle access to $\mathrm{EX}(f,U_n)$ for a k-junta function $f:\{0,1\}^n\to\{-1,1\}$ and a threshold m $(\leq k)$, it determines relevant coordinates by identifying all non-zero Fourier coefficients of order of up to m. Formally, **Find** satisfies Lemma 23.

Algorithm 1 Find(n,k,m,δ)

```
Input  : $n,k,m\in\mathbb{N}$, $\delta\in(0,1]$, and oracle access to $\mathrm{EX}(f,U_n)$
Output: a subset of relevant coordinates $I\subseteq[n]$
1 let $I=\emptyset$ and $M=\lceil 2^{2k+1}(m\ln n+\ln\delta^{-1}+\ln 2)\rceil$
2 forall $\alpha\in\{0,1\}^n$ with $1\leq|\alpha|\leq m$ do
3     take samples $(x^{(1)},b^{(1)}),\ldots,(x^{(M)},b^{(M)})\leftarrow\mathrm{EX}(f,U_n)$
4     estimate $\widehat{f}(\alpha)$ by $\tilde{f}(\alpha)=\frac{1}{M}\sum_{i=1}^{M}b^{(i)}\cdot\chi_\alpha(x^{(i)})$
5     if $\left|\tilde{f}(\alpha)\right|\geq 3\cdot 2^{-(k+1)}$ then
6         add all coordinates $j\in[n]$ satisfying that $\alpha_j\neq 0$ to $I$
```

Lemma 23 *For any input $n, k, m \in \mathbb{N}$, $\delta \in (0,1]$ and k-junta function $f : \{0,1\}^n \to \{-1,1\}$, Algorithm 1 (**Find**) outputs a subset $I \subseteq [n]$ of all coordinates i satisfying the following condition:*

there exists $\alpha \in \{0,1\}^n$ such that (i) $\alpha_i \neq 0$, (ii) $|\alpha| \leq m$, and (iii) $\widehat{f}(\alpha) \neq 0$,

with probability at least $1-\delta$ over the choice of examples by EX(f, U_n).

Further, the running time is bounded above by

$$n^m \cdot O\left(n \cdot 2^{2k}(m \log n + \log \delta^{-1})\right) \leq n^m \cdot \mathsf{poly}(n, 2^k, \log \delta^{-1}).$$

Proof This lemma mainly follows from the following basic fact: for any k-junta function, if $\widehat{f}(\alpha) \neq 0$ for $\alpha \in \{0,1\}^n$, then

$$\left|\widehat{f}(\alpha)\right| = |\mathrm{E}[f(x)\chi_\alpha(x)]| = |2\Pr[f(x) = \chi_\alpha(x)] - 1| \geq 2^{-(k-1)},$$

where the last inequality holds because $f(x)$ and $\chi_\alpha(x)$ depend only on at most k coordinates. By Hoeffding's inequality, **Find** estimates each Fourier coefficient within an estimation error $\pm 2^{-(k+1)}$ with failure probability at most $2\exp\left(-2M2^{-2(k+1)}\right) \leq \delta/n^m$ as indicated in line 4. Note that **Find** estimates all Fourier coefficients on α with $1 \leq |\alpha| \leq m$, and the number of such α is at most n^m. Thus, with probability at least $1-\delta$, $\alpha \in \{0,1\}^n$ passes the test in line 5 if and only if α satisfies the conditions (ii) and (iii). Because **Find** puts i satisfying the condition $\alpha_i \neq 0$ into I for only α that passed the test, I consists of i satisfying the condition in the lemma.

For each $\alpha \in \{0,1\}^n$, **Find** takes at most $O(nM) = O\left(n \cdot 2^{2k}(m \log n + \log \delta^{-1})\right)$ time. Since the number of α with $1 \leq |\alpha| \leq m$ is at most n^m, the running time of **Find** is bounded above by

$$n^m \cdot O\left(n \cdot 2^{2k}(m \log n + \log \delta^{-1})\right).$$

■

We use the above subroutine with a proper threshold m depending on η repeatedly to find all relevant coordinates of f as follows: Let $I = \emptyset$. We find some relevant coordinates by using **Find** with the threshold m and putting them into I. Then, we find other relevant coordinates and put them into I by applying **Find** again with respect to the same threshold m and a restricted function $f|_\rho$ for all restrictions ρ on I. We repeat this procedure until I gets unchanged.

Notice that the above strategy does not depend on the position of relevant and irrelevant coordinates. Thus, unless **Find** fails at some stage, it corresponds to the procedure in Definition 13 for a base function of a target function. Therefore, it is enough to establish the upper bound on the fraction of m-elusive functions in k-input functions to guarantee the success probability of the above strategy over the choice of target junta functions. For this reason, we will prove the following technical lemma in the next section.

Lemma 24 *For any $n, m \in \mathbb{N}$ with $2 \leq m \leq n$, the fraction of n-input m-elusive functions is bounded above by*

$$\left(\frac{m+1}{2^{n+1}}\right)^{\frac{m-1}{2}}.$$

Algorithm 2 A Heuristic Learner for k-Junta Functions (**HeurJunta**)

```
Input  : input size $n \in \mathbb{N}$, parameters $\delta, \eta \in (0, 1]$, and oracle access to EX$(f, U_n)$
Output: the set $I$ of relevant coordinates for $f$
7   let $I = \emptyset$ find the minimum $m \in \mathbb{N}$ satisfying that $2 \leq m \leq k-1$ and
        $$\left(\frac{m+1}{2^{k+1}}\right)^{\frac{m-1}{2}} \leq \eta, \qquad (1)$$
    if not, then set $m := k$
8   while $|I| < k$ do
9       let $I_{prev} := I$
10      forall restrictions $\rho$ on $I$ do
11          $I_{new} \leftarrow$ Find$(n - |I_{prev}|, k, m, \frac{\delta}{k2^{k-1}})$ with oracle access to EX$(f|_\rho, U_{n-|I_{prev}|})$
12          (if any) add all elements in $I_{new}$ to $I$
13      if $I = I_{prev}$ then break;
14  return $I$
```

Assume that Lemma 24 is correct at first. Then the heuristic algorithm to find relevant coordinates can be obtained in accordance with the above strategy as shown in Algorithm 2.

We show the following theorem on **HeurJunta**, which immediately implies Theorem 3.

Theorem 25 *Assume that $k := k(n) \geq c \log n$ for sufficiently large n, where $c > 0$ is a constant. On any input (n, δ, η), Algorithm 2 (**HeurJunta**) outputs a set of all relevant coordinates for at least $(1-\eta)$ fraction of k-junta functions with probability at least $1-\delta$ in time*

$$\eta^{-\frac{4}{c}} \cdot \mathsf{poly}(n, 2^k, \ln \delta^{-1}) \leq \mathsf{poly}(n, 2^k, \ln \delta^{-1}, \eta^{-1}).$$

Proof We only consider the case where n is sufficiently large to satisfy the condition $k \geq c \log n$.

For each loop in lines 8–13, the size of I increases by at least one. When the size of I reaches k, then **HeurJunta** jumps out of the loop. Thus, the loop is repeated at most k times. For each loop, **Find** is executed at most $2^{|I|} \leq 2^{k-1}$ times. Therefore, by the choice of confidence for **Find**, all (at most $k2^{k-1}$) executions of **Find** will succeed with probability at least $1-\delta$. In the subsequent argument, we assume that all executions of **Find** will succeed.

First, we show the correctness of **HeurJunta**. If $m = k$, then **Find** estimates all Fourier coefficients of order up to k and finds all relevant coordinates for any target k-junta functions. Thus, we only need to consider the case where $m \leq k-1$.

Note that **HeurJunta** is not affected by the position of relevant coordinates. By assuming that **Find** does not fail, **HeurJunta** fails in finding all relevant coordinates for a target k-junta function f if and only if the base function of f is m-elusive. By Lemma 24, the fraction of such m-elusive functions in k-input base functions is at most

$$\left(\frac{m+1}{2^{k+1}}\right)^{\frac{m-1}{2}} \leq \eta.$$

The remaining part is to give an upper bound on the running time of **HeurJunta**. We can assume that m satisfies the condition $m^2 \leq 2^m$ (i.e., $m \geq 4$), otherwise **HeurJunta** halts in time at most $k2^{k-1} \cdot \mathsf{poly}(n, 2^k, \log \delta^{-1}) = \mathsf{poly}(n, 2^k, \log \delta^{-1})$ by Lemma 23.

In both cases where $m \leq k-1$ and $m = k$, the inequality (1) is not satisfied by $m-1$. Thus, we have that

$$\left(\frac{2^k}{m}\right)^m < \eta^{-2} \cdot 2^{2-m} \cdot \left(\frac{2^k}{m}\right)^2 < 2^{2k}\eta^{-2},$$

and

$$m^m = \left(\frac{m^2}{m}\right)^m \leq \left(\frac{2^m}{m}\right)^m \leq \left(\frac{2^k}{m}\right)^m < 2^{2k}\eta^{-2}.$$

From the above two inequalities, we obtain

$$2^{km} < 2^{2k}\eta^{-2} \cdot m^m < 2^{4k}\eta^{-4}.$$

Remember that $2^k \geq n^c$ holds. By Lemma 23, the running time of **HeurJunta** is bounded above by

$$\begin{aligned} k \cdot 2^{k-1} \cdot n^m \cdot \mathsf{poly}(n, 2^k, \log \delta^{-1}) &\leq 2^{\frac{km}{c}} \cdot \mathsf{poly}(n, 2^k, \log \delta^{-1}) \\ &\leq 2^{\frac{4k}{c}} \eta^{-\frac{4}{c}} \cdot \mathsf{poly}(n, 2^k, \log \delta^{-1}) \\ &\leq \eta^{-\frac{4}{c}} \cdot \mathsf{poly}(n, 2^k, \log \delta^{-1}). \end{aligned}$$

■

C.1. A Proof of Lemma 24

As mentioned in Section 4.1, our basic strategy is to give a lower bound on the minimum distance of m-elusive functions, where the minimum distance of a class of functions is defined as follows.

Definition 26 (Minimum distance) *The Hamming distance $d(f,g)$ between two functions $f, g : \{0,1\}^n \to \{-1,1\}$ is given by $d(f,g) = \#\{x \in \{0,1\}^n : f(x) \neq g(x)\}$.*

For a class $\mathscr{C}_n$ of n-input functions, the minimum distance of $\mathscr{C}_n$ is defined by

$$\min_{f,g \in \mathscr{C}_n : f \neq g} d(f,g).$$

We will use the following useful facts on m-th correlation-immune functions.

Fact 2 (O'Donnell, 2014, Corollary 6.14) *A function $f : \{0,1\}^n \to \{-1,1\}$ is m-th order correlation-immune iff f's mean is unchanged by any restriction on a set of size by at most m.*

Lemma 27 *If $f : \{0,1\}^n \to \{-1,1\}$ is m-th order correlation-immune, then for any restriction ρ on the set of size i ($\leq m$),*

$$w(f|_\rho) = \frac{w(f)}{2^i}.$$

Further, $w(f) = 2^m \cdot M$ for some $M \in \mathbb{N} \cup \{0\}$ whenever f is m-th order correlation-immune.

Proof Let ρ be an arbitrary restriction on the set of size i. From Fact 2 and simple calculation, we have that

$$1 - 2^{-n+1} \cdot w(f) = \mathrm{E}[f] \overset{\text{(Fact 2)}}{=} \mathrm{E}[f|_\rho] = 1 - 2^{-n+i+1} \cdot w(f|_\rho).$$

Thus, $w(f|_\rho) = 2^{-i} w(f)$ holds. By applying this for $i = m$, we have that $w(f) = 2^m \cdot w(f|_\rho)$. ■

As a first step, we present a lower bound on the minimum distance of correlation-immune functions.

Lemma 28 *For $m \geq 2$, the minimum distance of m-th order correlation-immune functions is greater than $m+1$.*

Proof For contradiction, we assume that there exist two distinct functions $f, f' : \{0,1\}^n \to \{-1,1\}$ which are m-th order correlation-immune and their Hamming distance is at most $m+1$. In other words, there exist ℓ ($\leq m+1$) distinct elements $x^{(1)}, \dots, x^{(\ell)} \in \{0,1\}^n$ such that f takes a different value from f' on only $x^{(1)}, \dots, x^{(\ell)}$.

From Lemma 27, there exists $M \in \mathbb{N} \cup \{0\}$ such that $w(f') = 2^m M$. Because the distance between f and f' is at most $m+1$, we have that

$$2^m(M-1) < 2^m M - (m+1) \leq w(f) \leq 2^m M + (m+1) < 2^m(M+1),$$

where the first and last inequalities hold because $m \geq 2$. Thus, by Lemma 27, $w(f) = 2^m M = w(f')$.

For each $j \in [\ell - 1]$, we can take $i_j \in [n]$ such that $x^{(\ell)}_{i_j} \neq x^{(j)}_{i_j}$ because $x^{(\ell)} \neq x^{(j)}$. Let $I = \{i_1, \dots, i_{\ell-1}\}$. Now we define a restriction ρ on I by $\rho(i) = x^{(\ell)}_i$ for each $i \in I$. Because $|I| \leq m$, we have that $w(f'|_\rho) = w(f') \cdot 2^{-|I|}$ and $w(f|_\rho) = w(f) \cdot 2^{-|I|}$ by applying Lemma 27.

From the definition, ρ is inconsistent with all of $x^{(1)}$,..., $x^{(\ell-1)}$. Therefore, $f|_\rho$ takes a different value from $f'|_\rho$ on only $x^{(\ell)}$. This implies that either $w(f|_\rho) = w(f') \cdot 2^{-|I|} + 1$ or $w(f|_\rho) = w(f') \cdot 2^{-|I|} - 1$. In any case, we have that

$$w(f|_\rho) = w(f') \cdot 2^{-|I|} \pm 1 = w(f) \cdot 2^{-|I|} \pm 1 \neq w(f) \cdot 2^{-|I|} = w(f|_\rho),$$

which is a contradiction. ■

Herein, we show the lower bound on the minimum distance of elusive functions.

Theorem 29 *For $m \geq 2$, the minimum distance of m-elusive functions is greater than $m+1$.*

Proof Fix two distinct n-input m-elusive functions f and g arbitrarily. Let I and J be the sets yielded by the procedure in Definition 13 for f and g, respectively. Note that I (resp. J) is strictly contained in the set of relevant coordinates for f (resp. g). For any restriction ρ on I (resp. J), every α with $1 \leq |\alpha| \leq m$ satisfies the condition $\widehat{f|_\rho}(\alpha) = 0$ (resp. $\widehat{g|_\rho}(\alpha) = 0$), otherwise, another element i satisfying $\alpha_i \neq 0$ must be added to I (resp. J). This means that f (resp. g) becomes m-th order correlation-immune by applying any restriction on I (resp. J).

We demonstrate the theorem by using case studies on I and J.

(i) $I = J$. In this case, for any restriction ρ on I, $f|_\rho$ and $g|_\rho$ are both m-th order correlation-immune. However, because $f \neq g$, one of such restrictions ρ must satisfy $f|_\rho \neq g|_\rho$. From Lemma 28, the Hamming distance between $f|_\rho$ and $g|_\rho$ must be greater than $m+1$, so are f and g.

(ii) $I \neq J$. Remark that the procedure in Definition 13 does not depends on the order of the choice of α and ρ, so we can fix the order. Then at some stage, either of (a) $\widehat{f|_\rho}(\alpha) \neq 0$ and $\widehat{g|_\rho}(\alpha) = 0$ or (b) $\widehat{f|_\rho}(\alpha) = 0$ and $\widehat{g|_\rho}(\alpha) \neq 0$ must occur, otherwise, $I = J$. W.l.o.g., we can assume that the event (a) occurs. In other words, there exist a set $S \subseteq I \cap J$, a restriction ρ on S, and $\alpha \in \{0,1\}^{n-|S|}$ such that $\widehat{f|_\rho}(\alpha) \neq 0$ and $\widehat{g|_\rho}(\alpha) = 0$. For convenience, we will use the following notations: $n' = n - |S|$, $f' \equiv f|_\rho$, and $g' \equiv g|_\rho$.

Let $T = \{i \in [n'] : \alpha_i \neq 0\}$. Because all $i \in T$ is added to I by performing the procedure, $T \subseteq I \setminus S$ holds. Note that the value of χ_α is determined only by a partial assignment to T. Now we define integers $M^+_{f'}, M^-_{f'}, M^+_{g'}$, and $M^-_{g'}$ as follows.

$$\begin{aligned}
M^+_{f'} &= \#\{x \in \{0,1\}^{n'} : \chi_\alpha(x) = 1 \text{ and } f'(x) = -1\} \\
M^-_{f'} &= \#\{x \in \{0,1\}^{n'} : \chi_\alpha(x) = -1 \text{ and } f'(x) = -1\} \\
M^+_{g'} &= \#\{x \in \{0,1\}^{n'} : \chi_\alpha(x) = 1 \text{ and } g'(x) = -1\} \\
M^-_{g'} &= \#\{x \in \{0,1\}^{n'} : \chi_\alpha(x) = -1 \text{ and } g'(x) = -1\}.
\end{aligned}$$

By a simple calculation,

$$\begin{aligned}
\widehat{f'}(\alpha) &= \mathrm{E}_{x\sim\{0,1\}^{n'}}[f'(x)\chi_\alpha(x)] = 2^{-(n'-1)}\left(M^-_{f'} - M^+_{f'}\right) \\
\widehat{g'}(\alpha) &= \mathrm{E}_{x\sim\{0,1\}^{n'}}[g'(x)\chi_\alpha(x)] = 2^{-(n'-1)}\left(M^-_{g'} - M^+_{g'}\right).
\end{aligned}$$

Therefore, we have that $M^-_{f'} \neq M^+_{f'}$ and $M^-_{g'} = M^+_{g'}$.

For any restriction ρ^+ on $I \setminus S$ satisfying $\chi_\alpha|_{\rho^+} = 1$; notice that $f'|_{\rho^+}$ $(= (f|_\rho)|_{\rho^+})$ is m-th order correlation-immune, thus, $w(f'|_{\rho^+}) = 2^m \cdot M_{\rho^+}$ for some $M_{\rho^+} \in \mathbb{N} \cup \{0\}$ by Lemma 27. This implies that

$$M^+_{f'} = \sum_{\substack{\rho^+ \text{on } I\setminus S: \\ \chi_\alpha|_{\rho^+}=1}} w(f'|_{\rho^+}) = \sum_{\substack{\rho^+ \text{on } I\setminus S: \\ \chi_\alpha|_{\rho^+}=1}} 2^m M_{\rho^+}.$$

Therefore, $M^+_{f'}$ is divisible by 2^m. By the same argument, $M^-_{f'}$ is also divisible by 2^m. Because $M^+_{f'} \neq M^-_{f'}$, we have that $\left|M^+_{f'} - M^-_{f'}\right| \geq 2^m$. Thus, the Hamming distance between f' and g' is bounded as follows.

$$d(f', g') \geq \left|M^+_{f'} - M^+_{g'}\right| + \left|M^-_{f'} - M^-_{g'}\right| \geq \left|M^+_{f'} - M^+_{g'} - \left(M^-_{f'} - M^-_{g'}\right)\right| = \left|M^+_{f'} - M^-_{f'}\right| \geq 2^m.$$

Therefore, the Hamming distance between f and g is also at least $2^m > m+1$ for $m \geq 2$. ■

Theorem 29 implies Lemma 24 as follows.

Proof (Lemma 24) For each n-input function f, we define a Hamming ball B_f of radius $\lfloor \frac{m+1}{2} \rfloor$ as

$$B_f = \left\{ g : \{0,1\}^n \to \{-1,1\} \mid d(f,g) \leq \left\lfloor \frac{m+1}{2} \right\rfloor \right\}.$$

Then for any n-input function f, the size (volume) V of B_f is bounded below as follows.

$$V = \sum_{i=0}^{\lfloor \frac{m+1}{2} \rfloor} \binom{2^n}{i} \geq \binom{2^n}{\lfloor \frac{m+1}{2} \rfloor} \geq \left(\frac{2^n}{\lfloor \frac{m+1}{2} \rfloor} \right)^{\lfloor \frac{m+1}{2} \rfloor} \geq \left(\frac{2^{n+1}}{m+1} \right)^{\frac{m-1}{2}}.$$

Let $N_{n,m}$ be the number of n-input m-elusive functions. From Theorem 29, for any two distinct n-input m-elusive functions f and g, V_f and V_g are disjoint, otherwise, $d(f,g) \leq \lfloor \frac{m+1}{2} \rfloor + \lfloor \frac{m+1}{2} \rfloor \leq m+1$. Thus, we have that

$$2^{2^n} \geq N_{n,m} \cdot V \geq N_{n,m} \cdot \left(\frac{2^{n+1}}{m+1} \right)^{\frac{m-1}{2}}.$$

Hence, we can conclude that

$$\frac{\#(n\text{-input } m\text{-elusive functions})}{\#(n\text{-input functions})} = \frac{N_{n,m}}{2^{2^n}} \leq \left(\frac{m+1}{2^{n+1}} \right)^{\frac{m-1}{2}}.$$

■

Appendix D. Weak Cryptography and HeurPAC Learnability

In this section, we show the following main theorem and its corollaries. Note that Theorems 6 and 8 immediately follow from Theorem 30 and Fact 1.

Theorem 30 *For any class $\mathscr{C}$ evaluated in polynomial-time, the followings hold:*

1. *if $\mathscr{C}$ is not heurPAC learnable, then AIPRG exists and*
2. *if $\mathscr{C}$ is not heurPAC learnable on the uniform distribution, then io-PRG exists.*

Now, we introduce the following key lemma, which was originally introduced by Levin (1987).

Fact 3 (Levin's version of XOR lemma) *There exists a randomized algorithm* **Boost** *such that for any function $f : \{0,1\}^n \to \{0,1\}$ and distribution D on $\{0,1\}^n$,* **Boost** *is given $d \in \mathbb{N}$, $\gamma \in (0,1]$, a circuit $C^{\oplus} : \{0,1\}^{dn} \to \{0,1\}$ of size s, and oracle access to $EX(f,D)$ as input, then it outputs a (possibly randomized) circuit $C : \{0,1\}^n \to \{0,1\}$ of size at most $\mathsf{poly}(n,d,s,\gamma^{-1})$ in time $\mathsf{poly}(n,d,s,\gamma^{-1})$. Further, if the given $C^{\oplus}$ satisfies the expression:*

$$\Pr_{x \leftarrow D^m}[C^{\oplus}(x) = f^{\oplus d}(x)] \geq \frac{1}{2} + \gamma,$$

then, the resulting circuit C satisfies the expression:

$$\Pr_{C, x \leftarrow D}[C(x) = f(x)] \geq \frac{1}{2} + \frac{\gamma^{\frac{1}{d}}}{2},$$

with high probability over the choice of examples.

Proof Refer to (Goldreich et al., 2011, Sections 2 and 3). Note that their construction algorithm in the XOR lemma (Goldreich et al., 2011, Lemma 2) works even in the case where ϵ is given as input (instead of a fixed function). Fact 3 is shown by setting the function ϵ in the XOR lemma to γ in the above statement. ■

Note that the failure probability of **Boost** can be reduced to arbitrary δ within multiplicative loss of time $O(\log \delta^{-1})$ by applying the standard repeating and testing technique. Thus, for the sake of simplicity, we assume that **Boost** succeeds in finding a circuit C as in Fact 3 with probability 1.

D.1. The Case of AIPRG

First, we introduce a natural universal AIPRG as follows.

Definition 31 *An auxiliary-input function $UC = \{UC_z : \{0,1\}^{n(|z|)} \to \{0,1\}^{n(|z|)+1}\}_{z\in\{0,1\}^*}$, where $n(|z|) = \max n \in \mathbb{N} : e_{n^2}(n) \cdot (n+1) \le |z|$, is given by*

$$UC_z(x) = C_z^{(1)}(x) \circ \cdots \circ C_z^{(n+1)}(x),$$

where $z = C_z^{(1)} \circ \cdots \circ C_z^{(n+1)} \circ z_{left}$, each $C_z^{(i)} \in \{0,1\}^{e_{n^2}(n)}$ (regarded as an n-input circuit of size n^2), and $z_{left} \in \{0,1\}^{|z|-e_{n^2}(n)\cdot(n+1)}$.

Then we can restate Theorem 30 (1) and present a formal proof as follows.

Theorem 32 *Let $\mathscr{C}$ be a concept class with a polynomial-time evaluation $C : \{0,1\}^{s(n)} \times \{0,1\}^n \to \{0,1\}$. If UC (in Definition 31) is not AIPRG, then $\mathscr{C}$ is heurPAC learnable.*

Proof By the assumption that UC is not AIPRG, there exist a polynomial-time adversary[7] A and a function $\gamma(n) = \mathsf{poly}^{-1}(n)$ such that for all $z \in \{0,1\}^*$,

$$\Pr[A(z, UC_z(U_n)) = 1] - \Pr[A(z, U_{n+1}) = 1] \ge \gamma(n).$$

We consider heurPAC learning on the setting of input size n, accuracy parameter ϵ, heuristic parameter η, and example distribution D on $\{0,1\}^n$. Remark that we will construct the heurPAC learner with non-negligible confidence error in $n, s(n), \epsilon^{-1}$, and η^{-1}, which can be improved to arbitrary δ by applying the standard repeating and testing method with multiplicative loss of time $\ln \delta^{-1} \cdot \mathsf{poly}(n, s(n), \epsilon^{-1}, \eta^{-1})$. For a more detailed explanation, refer to (Haussler et al., 1988, Lemma 3.4).

The outline of the heurPAC learner is given as follows: first, we define an intermediate auxiliary-input generator G based on C and the given ϵ and η, then we construct an adversary A' for G based on A, and finally for any but η fraction of target function f, we construct a hypothesis h that is ϵ-close to f based on A' with access to an example oracle. Note that the above G, A' and h are uniformly and efficiently constructible.

7. Strictly speaking, to remove the vertical bars for an absolute value in Definition 15, we test the behavior of the adversary for the given z first, then take a negation according to the result.

Now we define a generator $G : \{0,1\}^{d\cdot(vs(n)+1)\cdot n} \times \{0,1\}^{vs(n)} \to \{0,1\}^{vs(n)+1}$ as follows:

$$G\left(x^{(1,1)} \circ \cdots \circ x^{(1,d)} \circ x^{(2,1)} \cdots \circ x^{(2,d)} \circ x^{(3,1)} \circ \cdots \circ x^{(vs(n)+1,d)}, f^{(1)} \circ \cdots \circ f^{(v)}\right)$$
$$= \bigoplus_{j=1}^{d} \bigoplus_{\ell=1}^{v} f^{(\ell)}\left(x^{(1,j)}\right) \circ \cdots \circ \bigoplus_{j=1}^{d} \bigoplus_{\ell=1}^{v} f^{(\ell)}\left(x^{(vs(n)+1,j)}\right),$$

where $x^{(i,j)} \in \{0,1\}^n$ for each $(i,j) \in [vs(n)+1] \times [d]$, $v = \lceil \frac{n}{\eta} \rceil$, and d denotes an integer specified later to satisfy that $d \geq \epsilon^{-1} \cdot g(d, n, s(n), \epsilon^{-1}, \eta^{-1})$ for a certain function $g(d, n, s(n), \epsilon^{-1}, \eta^{-1}) = O(\log(\mathsf{poly}(d, n, s(n), \epsilon^{-1}, \eta^{-1})))$. Note that, as we will see below, g depends only on $\gamma(\cdot)$ and the running time of C, and it is enough to take as $d = \epsilon^{-2} \cdot O(\log n + \log s(n) + \log \eta^{-1})$ at the end.

It can be easily verified that each bit of G is computable in time $d \cdot v \cdot p(n, s(n))$ for some polynomial p depending on the computational complexity of C. Define $N := N(d, n, s(n), \epsilon, \eta) \in \mathbb{N}$ as

$$N = \max\left\{vs(n), \left\lceil \sqrt{d \cdot v \cdot p(n, s(n))} \right\rceil\right\},$$

and another generator $G' : \{0,1\}^{d(vs(n)+1)n} \times \{0,1\}^N \to \{0,1\}^{N+1}$ by

$$G'(X, u) = G(X, u_{[vs(n)]}) \circ u_{vs(n)+1} \circ \cdots \circ u_N.$$

Then each bit of G' is computable in time $d \cdot v \cdot p(n, s(n)) \leq N^2$. Therefore, by embedding a given $X \in \{0,1\}^{d(vs(n)+1)n}$ into G', we can construct $N+1$ circuits $C_X^{(1)}, \ldots, C_X^{(N+1)} \in \{0,1\}^{e_{N^2}(N)}$ of input length N and size N^2 such that

$$G'(X, u) = C_X^{(1)}(u) \circ \cdots \circ C_X^{(N+1)}(u).$$

Thus, by plugging $C_X^{(1)}, \ldots, C_X^{(N+1)}$ into A, we have that for any $X \in \{0,1\}^{d(vs(n)+1)n}$,

$$\Pr_{A,U_N}\left[A(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, G'(X, U_N)) = 1\right] - \Pr_{A,U_{N+1}}\left[A(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, U_{N+1}) = 1\right] \geq \gamma(N). \tag{2}$$

Define an adversary A' for G as

$$A'(X, r) = A\left(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, r \circ r'\right),$$

where $X \in \{0,1\}^{d(vs(n)+1)n}$, $r \in \{0,1\}^{vs(n)+1}$, $r' \in \{0,1\}^{N-vs(n)}$, and r' is selected at random by A. Then A' is executed in time $\mathsf{poly}(N) = \mathsf{poly}(d, n, s(n), \epsilon^{-1}, \eta^{-1})$ and satisfies the condition that for any $X \in \{0,1\}^{d(vs(n)+1)n}$,

$$\begin{aligned}
&\Pr_{A', f^{(1)}, \ldots, f^{(v)}}\left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(v)})) = 1\right] - \Pr_{A', U_{vs(n)+1}}\left[A'(X, U_{vs(n)+1}) = 1\right] \\
&\quad = \Pr_{A, U_{vs(n)}, U_{N-vs(n)}}\left[A(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, G(X, U_{vs(n)}) \circ U_{N-vs(n)}) = 1\right] \\
&\qquad\qquad - \Pr_{A, U_{vs(n)+1}, U_{N-vs(n)}}\left[A(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, U_{vs(n)+1} \circ U_{N-vs(n)}) = 1\right] \\
&\quad = \Pr_{A,U_N}\left[A(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, G'(X, U_N)) = 1\right] - \Pr_{A,U_{N+1}}\left[A(C_X^{(1)} \circ \cdots \circ C_X^{(N+1)}, U_{N+1}) = 1\right] \\
&\quad \geq \gamma(N),
\end{aligned} \tag{3}$$

where the last inequality follows from inequality (2).

Now, we assume that $\gamma(N) > 2e^{-n}$. Note that we will deal with the other case where $\gamma(N) \leq 2e^{-n}$ later. For convenience, we will use the following notations:

$$p_X^{TR} := \Pr_{A', U_{vs(n)+1}} \left[A'(X, U_{vs(n)+1}) = 1\right]$$
$$p_D^{TR} := \mathrm{E}_{X \leftarrow D^{d(vs(n)+1)}} \left[p_X^{TR}\right] = \Pr_{\substack{X \leftarrow D^{d(vs(n)+1)} \\ A', U_{vs(n)+1}}} \left[A'(X, U_{vs(n)+1}) = 1\right],$$

where we regard $(x^{(1)}, \ldots, x^{(d(vs(n)+1))}) \leftarrow D^{d(vs(n)+1)}$ as a concatenated string $x^{(1)} \circ \cdots \circ x^{(d(vs(n)+1))}$.

For the example distribution D, we define "bad" target functions as follows: $f \in \{0,1\}^{s(n)}$ is "bad" iff for all $i \in [v]$,

$$\Pr_{\substack{X \leftarrow D^{d(vs(n)+1)}, \\ A', f^{(1)}, \ldots, f^{(i-1)}, \\ f^{(i+1)}, \ldots, f^{(v)}}} \left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(i-1)} \circ f \circ f^{(i+1)} \circ \cdots \circ f^{(v)})) = 1\right] - p_D^{TR} \leq \frac{\gamma(N)}{2}.$$

Let $B_D \subseteq \{0,1\}^{s(n)}$ be the set of all "bad" target functions. Then we will show the following two claims: (1) $|B_D| \leq \eta \cdot |\{0,1\}^{s(n)}|$ and (2) for any $f \notin B_D$, we can construct a hypothesis h that is ϵ-close to f based on A'.

We show the first claim by contradiction. Assume that $\Pr_{f \sim \{0,1\}^{s(n)}}[f \in B_D] > \eta$. Then we have that

$$\begin{aligned}
\gamma(N) &\leq \min_{X \in \mathrm{Supp} D^{d(vs(n)+1)}} \left(\Pr_{A', f^{(1)}, \ldots, f^{(v)}} \left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(v)})) = 1\right] - p_X^{TR} \right) \qquad (\because (3)) \\
&\leq \Pr_{\substack{X \leftarrow D^{d(vs(n)+1)} \\ A', f^{(1)}, \ldots, f^{(v)}}} \left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(v)})) = 1\right] - p_D^{TR} \\
&= \Pr\left[\exists i \in [v] \text{ s.t. } f^{(i)} \in B_D\right] \cdot \left(\Pr\left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(v)})) = 1 \middle| \exists i \in [v] \text{ s.t. } f^{(i)} \in B_D\right] - p_D^{TR}\right) \\
&\quad + \Pr\left[\forall i \in [v]\ f^{(i)} \notin B_D\right] \cdot \left(\Pr\left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(v)})) = 1 \middle| \forall i \in [v]\ f^{(i)} \notin B_D\right] - p_D^{TR}\right) \\
&\leq 1 \cdot \frac{\gamma(N)}{2} + \Pr\left[\forall i \in [v]\ f^{(i)} \notin B_D\right] \cdot 1 \\
&\leq \frac{\gamma(N)}{2} + (1-\eta)^v \leq \frac{\gamma(N)}{2} + (1-\eta)^{\frac{n}{\eta}} \leq \frac{\gamma(N)}{2} + e^{-n} < \gamma(N),
\end{aligned}$$

which is a contradiction. Therefore, $\Pr_{f \sim \{0,1\}^{s(n)}}[f \in B_D] \leq \eta$ holds.

For the second claim, assume that the target function f is not "bad," i.e., there exists $i^* \in [v]$ such that

$$\Pr_{\substack{X \leftarrow D^{d(vs(n)+1)}, \\ A', f^{(1)}, \ldots, f^{(i^*-1)}, \\ f^{(i^*+1)}, \ldots, f^{(v)}}} \left[A'(X, G(X, f^{(1)} \circ \cdots \circ f^{(i^*-1)} \circ f \circ f^{(i^*+1)} \circ \cdots \circ f^{(v)})) = 1\right] - p_D^{TR} > \frac{\gamma(N)}{2}. \tag{4}$$

For heurPAC learning, the learner randomly guesses the above i^*. In the following, we assume that the learner succeeds in guessing such an i^*, which occurs with probability at least $1/v$.

We apply Yao's next-bit generator to translate the distinguisher A' into a prediction algorithm for $f^{\oplus\oplus} : \{0,1\}^{dn} \to \{0,1\}$ defined by

$$f^{\oplus\oplus}(x^{(1)}, \dots, x^{(d)}) = \bigoplus_{j=1}^{d} \bigoplus_{\ell=1}^{v} f^{(\ell)}\left(x^{(j)}\right), \text{ where } f^{(i^*)} \equiv f.$$

Notice that G can be rewritten by using $f^{\oplus\oplus}$ as

$$\begin{aligned} G(x^{(1,1)}\circ\cdots\circ x^{(1,d)}\circ\cdots\circ x^{(vs(n)+1,1)}\circ\cdots\circ x^{(vs(n)+1,d)}, f^{(1)}\circ\cdots\circ f^{(i^*-1)}\circ f\circ f^{(i^*+1)}\circ\cdots\circ f^{(v)})) \\ = f^{\oplus\oplus}(x^{(1,1)}, \dots, x^{(1,d)}) \circ \cdots \circ f^{\oplus\oplus}(x^{(vs(n)+1,1)}, \dots, x^{(vs(n)+1,d)}) \end{aligned}$$

Now, we construct a learning algorithm $L^{\oplus\oplus}$ as Algorithm 3. For the sake of simplicity, we use the notation F to refer to $(f^{(1)}, \dots, f^{(i^*-1)}, f^{(i^*+1)}, \dots, f^{(v)})$.

Algorithm 3 $L^{\oplus\oplus}$

```
Input  : examples $S = \{(x^{(1)}, b^{(1)}), \dots, (x^{(vs(n))}, b^{(vs(n))})\}$, and a challenge $x \leftarrow D^d$,
         where $f^{\oplus\oplus} : \{0,1\}^{dn} \to \{0,1\}$ (defined as above w.r.t. $F \leftarrow_u \{0,1\}^{s(n)(v-1)}$),
         and $(x^{(i)}, b^{(i)}) \leftarrow$EX$(f^{\oplus\oplus}, D^d)$.
Output: $f^{\oplus}(x)$
15 select $i \leftarrow_u [vs(n)+1]$ and $c^*, c_i, \dots, c_{vs(n)} \leftarrow_u \{0,1\}$
16 let $X = x^{(1)} \circ \cdots \circ x^{(i-1)} \circ x \circ x^i \circ \cdots \circ x^{(vs(n))}$
17 execute $b \leftarrow A'\left(X, b^{(1)} \circ \cdots \circ b^{(i-1)} \circ c^* \circ c_i, \circ \cdots \circ c_{vs(n)}\right)$
18 if $b = 1$ then return $c^*$, otherwise return $1 - c^*$
```

Then the running time of $L^{\oplus\oplus}$ is bounded above by

$$O(dnvs(n)) + (\text{the running time of } A') \leq O(dnvs(n)) + \mathsf{poly}(N),$$

and by the standard hybrid argument (for the details, refer to Goldreich, 2006),

$$\Pr_{L^{\oplus\oplus},S,x,F}\left[L^{\oplus\oplus}(S,x) = f^{\oplus\oplus}(x)\right] \geq \frac{1}{2} + \frac{\gamma(N)}{2(vs(n)+1)}.$$

Hence, we have that

$$\Pr_{L^{\oplus\oplus},S,F}\left[\Pr_{x\leftarrow D}\left[L^{\oplus\oplus}(S,x) = f^{\oplus\oplus}(x)\right] \geq \frac{1}{2} + \frac{\gamma(N)}{4(vs(n)+1)}\right] \geq \frac{\gamma(N)}{4(vs(n)+1)}, \tag{5}$$

otherwise,

$$\Pr_{L^{\oplus\oplus},S,x,F}\left[L^{\oplus\oplus}(S,x) = f^{\oplus\oplus}(x)\right] < 1 \cdot \frac{\gamma(N)}{4(vs(n)+1)} + \left(\frac{1}{2} + \frac{\gamma(N)}{4(vs(n)+1)}\right) \cdot 1 = \frac{1}{2} + \frac{\gamma(N)}{2(vs(n)+1)}.$$

Now, consider a hypothesis constructed by the following procedure:

1. guess i^* satisfying inequality (4) by $\tilde{i} \leftarrow_u [\ell]$;
2. select $F \leftarrow_u \{0,1\}^{s(n)(v-1)}$;
3. get examples $S = \left\{(x^{(1)}, f(x^{(i)})), \dots, (x^{(dvs(n))}, f(x^{(dvs(n))}))\right\} \leftarrow \mathrm{EX}(f, D)$;
4. calculate each value $f^{(\ell)}(x^{(i)})$ for $i \in [dvs(n)]$ and $\ell \in [v] \setminus \{\tilde{i}\}$ and construct an example set S' of size $vs(n)$ for a new target function $f^{\oplus\oplus}$ under an example distribution D^d (note that we need $dvs(n)$ examples for the original f);
5. select a random seed $r \in \{0,1\}^{O(dnvs(n))+\mathsf{poly}(N)}$ for $L^{\oplus\oplus}$;
6. define $h^{\oplus\oplus}_{\tilde{i},r,S,F} : \{0,1\}^{dn} \to \{0,1\}$ by $h^{\oplus\oplus}_{\tilde{i},r,S,F}(x) = L^{\oplus\oplus}(S', x; r)$.

Then under the condition that $f \notin B_D$,

$$\Pr_{\tilde{i}}[\tilde{i} \text{ satisfies inequality (4)}] \geq \frac{1}{v}.$$

By applying inequality (5), with probability at least $\frac{\gamma(N)}{4(vs(n)+1)}$,

$$\Pr_{x \leftarrow D^d}[h^{\oplus\oplus}_{i^*,r,S,F}(x) = f^{\oplus\oplus}(x)] \geq \frac{1}{2} + \frac{\gamma(N)}{4(vs(n)+1)}$$

for any i^* satisfying inequality (4). Now we move on to the following step 7,

7. define a hypothesis $h^{\oplus}_{\tilde{i},r,S,F} : \{0,1\}^{dn} \to \{0,1\}$ by

$$h^{\oplus}_{\tilde{i},r,S,F}(x) = h^{\oplus\oplus}_{\tilde{i},r,S,F}(x) \oplus f^{(1)^{\oplus d}}(x) \oplus \cdots \oplus f^{(\tilde{i}-1)^{\oplus d}}(x) \oplus f^{(\tilde{i}+1)^{\oplus d}}(x) \oplus \cdots \oplus f^{(v)^{\oplus d}}(x),$$

Then we can show that for any $f \notin B_D$,

$$\Pr_{x \leftarrow D^d}[h^{\oplus}_{\tilde{i},r,S,F}(x) = f^{\oplus d}(x)] = \Pr_{x \leftarrow D^d}[h^{\oplus\oplus}_{\tilde{i},r,S,F}(x) = f^{\oplus\oplus}(x)] \geq \frac{1}{2} + \frac{\gamma(N)}{4(vs(n)+1)}, \tag{6}$$

with probability at least $\frac{1}{v} \cdot \frac{\gamma(N)}{4(vs(n)+1)}$ over the choices of $\tilde{i}, r, S$, and F.

Now assume that the learner can find such a good hypothesis $h^{\oplus} \equiv h^{\oplus}_{i^*,r,S,F}$ satisfying inequality (6). Then by applying the algorithm **Boost** in the XOR lemma (Fact 3) on input d, $\gamma(N)/4(vs(n)+1)$, and $h^{\oplus}$ with access to $\mathrm{EX}(f, D)$, the learner can also construct (possibly randomized) $h : \{0,1\}^n \to \{0,1\}$ such that

$$\Pr_{h, x \leftarrow D}[h(x) \neq f(x)] \leq \frac{1}{2} - \frac{1}{2} \cdot \left(\frac{\gamma(N)}{4(vs(n)+1)}\right)^{\frac{1}{d}}.$$

To bound the above error probability, we will use the following simple inequality.

Claim 1 *For any $x > 0$ and $d > 0$, if $d \geq \ln x^{-1}$, then*

$$1 - x^{\frac{1}{d}} \leq \frac{\ln x^{-1}}{d}.$$

Proof (Claim 1) The claim simply holds by rearranging the following inequality:

$$\left(1-\frac{\ln x^{-1}}{d}\right)^{d} \leq e^{-\ln x^{-1}} = x.$$

■

Now, assume that $d \in \mathbb{N}$ was selected to satisfy the expression that

$$d \geq \frac{1}{\epsilon} \cdot \left(\ln \gamma(N)^{-1} + \ln(vs(n)+1) + \ln 4\right). \tag{7}$$

Then, we can show that h is $\epsilon/2$-close to f as follows:

$$\begin{aligned}\Pr_{h,x\leftarrow D}[h(x) \neq f(x)] &\leq \frac{1}{2} - \frac{1}{2} \cdot \left(\frac{\gamma(N)}{4(vs(n)+1)}\right)^{\frac{1}{d}} \\ &\leq \frac{1}{2d} \cdot \left(\ln \gamma(N)^{-1} + \ln(vs(n)+1) + \ln 4\right) \qquad (\because \text{Claim 1}) \\ &\leq \frac{\epsilon}{2}. \quad (\because (7))\end{aligned}$$

To derandomize the hypothesis h, we simply select a binary string at random and embed it into h as h's randomness. By Markov's inequality, such an h is ϵ-close to f with probability at least $1/2$.

Remember that $N \leq \sqrt{d} \cdot \mathsf{poly}(n, s(n), \eta^{-1})$ and $\gamma^{-1}(N) \leq \mathsf{poly}(N)$. Therefore, there exist a constant $c > 0$ and a function $l(n, s(n), \eta) = O(\log n + \log s(n) + \log \eta^{-1})$ such that

$$\frac{c}{\epsilon}\ln d + \frac{l(n, s(n), \eta)}{\epsilon} \geq \frac{1}{\epsilon} \cdot \left(\ln \gamma(N)^{-1} + \ln(vs(n)+1) + \ln 4\right).$$

It is not hard to see that, for instance, the following choice of d is enough for inequality (7),

$$d = \max\left\{\left\lceil\frac{c}{\epsilon}\right\rceil^2, \left\lceil\frac{l(n, s(n), \eta)}{\epsilon}\right\rceil^2, 49\right\} = O\left(\epsilon^{-2}(\log n + \log s(n) + \log \eta^{-1})\right).$$

Therefore, the running time of the above learning procedure is at most

$$dnvs(n) \cdot \mathsf{poly}(N) + \mathsf{poly}\left(n, d, dnvs(n)\mathsf{poly}(N), \frac{4(vs(n)+1)}{\gamma(N)}\right) \leq \mathsf{poly}(n, s(n), \epsilon^{-1}, \eta^{-1}),$$

and the confidence probability is at least

$$\frac{1}{v} \cdot \frac{\gamma(N)}{4(vs(n)+1)} \cdot \frac{1}{2}.$$

By applying the standard repeating and testing technique, the confidence error can be reduced to an arbitrary δ in time

$$O\left(v \cdot 4(vs(n)+1) \cdot \gamma(N)^{-1} \cdot \ln \delta^{-1}\right) \cdot \mathsf{poly}(n, s(n), \epsilon^{-1}, \eta^{-1}) \leq \mathsf{poly}(n, s(n), \epsilon^{-1}, \eta^{-1}) \cdot \ln \delta^{-1}.$$

The remaining part is the case where $\gamma(N) \leq 2e^{-n}$. In this case, we have that

$$2^n < e^n \leq 2\gamma(N)^{-1} \leq q(n, s(n), \epsilon^{-1}, \eta^{-1}),$$

for some polynomial q. Thus, the learner can approximate a truth-table of arbitrary target function f directly from examples (i.e., $\eta = 0$). Specifically, for each $x \in \{0,1\}^n$, the learner tries to find the pair $(x, f(x))$ in $M := \lceil 2^n \epsilon^{-1}(n \ln 2 + \ln \delta^{-1}) \rceil$ examples from $\mathrm{EX}(f, D)$, and then the learner outputs a hypothesis h consistent with the identified (partial) truth-table.

We show the correctness of the method as follows. If $x \in \{0,1\}^n$ satisfies $D(x) > \epsilon/2^n$, then from the standard probabilistic argument, $(x, f(x))$ is contained in M examples with probability at least $1 - \delta/2^n$. By the union bound, the learner can find all such x with probability at least $1 - \delta$. In this case, the hypothesis h corresponds to f on every x satisfying $D(x) > \epsilon/2^n$. Thus, the error probability is bounded above by $2^n \cdot \epsilon/2^n = \epsilon$ on the distribution D. Obviously, the running time is bounded above by the expression:

$$\begin{aligned} O(2^n \cdot nM) &= 2^{2n} \cdot \mathsf{poly}(n, \epsilon^{-1}) \cdot \ln \delta^{-1} \\ &\le q(n, s(n), \epsilon^{-1}, \eta^{-1})^2 \cdot \mathsf{poly}(n, \epsilon^{-1}) \cdot \ln \delta^{-1} \le \mathsf{poly}(n, s(n), \epsilon^{-1}, \eta^{-1}) \cdot \ln \delta^{-1}. \end{aligned}$$

■

D.2. The Case of io-PRG

In the case of io-PRG, we will use the famous pairing function at the core of the construction.

Fact 4 (Cantor's pairing function, Cantor, 1878) *Define a function $N_C : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by*

$$N_C(n, m) = \frac{1}{2}(n + m - 1)(n + m - 2) + m.$$

Then N_C is a bijection, and its inverse function $N_C^{-1}(N) = (n, m)$ can be expressed as follows:

$$a = \left\lceil \frac{\sqrt{8N+1} - 3}{2} \right\rceil, \quad m = N - \frac{1}{2}(a^2 + a), \quad n = a + 2 - m.$$

We define a function $N_{C,4} : \mathbb{N}^4 \to \mathbb{N}$ by $N_{C,4}(n_1, n_2, n_3, n_4) = N_C(N_C(N_C(n_1, n_2), n_3), n_4)$. From Fact 4, it is not hard to see the following:

- $N_{C,4}$ is a bijection, and its inverse function $N_{C,4}^{-1}$ is efficiently computable and
- $\max_{i \in [4]} n_i \le N_{C,4}(n_1, n_2, n_3, n_4) \le \mathsf{poly}(n_1, n_2, n_3, n_4)$.

Now, we restate Theorem 30 (2) and present its formal proof as follows.

Theorem 33 *Let $\mathscr{C}$ be a concept class with a polynomial-time evaluation $C : \{0,1\}^{s(n)} \times \{0,1\}^n \to \{0,1\}$. If $\mathscr{C}$ is not heurPAC learnable on the uniform distribution, then io-PRG exists.*

Proof For each $N \in \mathbb{N}$, let $(n, s, v, d) = N_{C,4}^{-1}(N)$. We define a generator $G = \{G_N : \{0,1\}^{N^2(N^2+1)+N^2} \to \{0,1\}^{N^2(N^2+1)+N^2+1}\}_{N \in \mathbb{N}}$ as follows:

$$G_N(x) = \begin{cases} x^{(1,1)} \circ \cdots \circ x^{(vs+1,d)} \circ x_{left} \circ \bigoplus_{j=1}^{d} \bigoplus_{\ell=1}^{v} f^{(\ell)}\left(x^{(1,j)}\right) \circ \cdots \circ \bigoplus_{j=1}^{d} \bigoplus_{\ell=1}^{v} f^{(\ell)}\left(x^{(vs+1,j)}\right) & \text{if } s = s(n) \\ 0^{N^2(N^2+1)+N^2+1} & \text{otherwise} \end{cases}$$

where $x^{(i,j)} \in \{0,1\}^n$ for each $(i,j) \in [vs(n)+1] \times [d]$, $f^{(\ell)} \in \{0,1\}^{s(n)}$ for each $\ell \in [v]$, $x_{left} \in \{0,1\}^{N^2(N^2+1)+N^2-(dn(vs(n)+1)+vs(n))}$, and

$$x = f^{(1)} \circ \cdots \circ f^{(v)} \circ x^{(1,1)} \circ \cdots \circ x^{(1,d)} \circ x^{(2,1)} \cdots \circ x^{(2,d)} \circ x^{(3,1)} \circ \cdots \circ x^{(vs+1,d)} \circ x_{left}.$$

Note that G is computable in time $\mathsf{poly}(N) + dv(vs+1) \cdot \mathsf{poly}(n, s(n)) = \mathsf{poly}(N)$. In the remaining part, we show by contraposition that G is io-PRG if $\mathscr{C}$ is not heurPAC learnable on the uniform distribution.

Let A be a polynomial-time adversary for G. Because $N_{C,4}(n,s,v,d) \le \mathsf{poly}(n,s,v,d)$, there exists a function $\gamma(n,s,\eta^{-1},d) = \mathsf{poly}^{-1}(n,s,\eta^{-1},d)$ such that for each $n, d \in \mathbb{N}$ and $\eta \in (0,1]$,

$$\Pr\left[A\left(1^N, G_N\left(U_{N^4+2N^2}\right)\right) = 1\right] - \Pr\left[A\left(1^N, U_{N^4+2N^2+1}\right) = 1\right] \ge \gamma(n, s(n), \eta^{-1}, d),$$

where $N = N_{C,4}(n, s(n), \lceil \frac{n}{\eta} \rceil, d)$.

The subsequent argument mainly follows the proof of Theorem 32. We consider heurPAC learning on a setting of input size n, accuracy parameter ϵ, and heuristic parameter η. Let $v = \lceil \frac{n}{\eta} \rceil$ and $N = N_{C,4}(n, s(n), v, d)$ as above.

For the sake of simplicity, let p^{TR} denote the following probability:

$$p^{TR} := \Pr_{A, U_{N^4+2N^2+1}}\left[A\left(1^N, U_{N^4+2N^2+1}\right) = 1\right].$$

We define a "bad" target function $f \in \{0,1\}^{s(n)}$ as a function satisfying the condition that for any $i \in [v]$,

$$\Pr_{\substack{A, U_M, \\ f^{(1)}, \ldots, f^{(i-1)}, \\ f^{(i+1)}, \ldots, f^{(v)}}}\left[A(1^N, G_N(f^{(1)} \circ \cdots \circ f^{(i-1)} \circ f \circ f^{(i+1)} \circ \cdots \circ f^{(v)} \circ U_M)) = 1\right] - p^{TR} \le \frac{\gamma(n, s(n), \eta^{-1}, d)}{2},$$

where $M := N^4 + 2N^2 - v \cdot s(n)$. Let B be the set of such "bad" functions.

First, assume that $\gamma(n, s(n), \eta^{-1}, d) > 2e^{-n}$. Then by the same argument as Theorem 32, we have that $\Pr_{f \sim \{0,1\}^{s(n)}}[f \in B] \le \eta$.

Thus, we consider a case where $f \notin B$, i.e., there exists at least one $i^* \in [v]$ such that

$$\Pr_{\substack{A, U_M, \\ f^{(1)}, \ldots, f^{(i^*-1)}, \\ f^{(i^*+1)}, \ldots, f^{(v)}}}\left[A(1^N, G_N(f^{(1)} \circ \cdots \circ f^{(i^*-1)} \circ f \circ f^{(i^*+1)} \circ \cdots \circ f^{(v)} \circ U_M)) = 1\right] - p^{TR} > \frac{\gamma(n, s(n), \eta^{-1}, d)}{2}.$$

For heurPAC learning, the learner randomly guesses the above i^*. In the following, we assume that the learner succeeds in guessing such an i^*, which occurs with probability at least $\lceil n/\eta \rceil^{-1}$.

Now, we apply Yao's next bit generator to translate the distinguisher A into a weak learner $L^{\oplus\oplus}$ for $f^{\oplus\oplus} : \{0,1\}^{dn} \to \{0,1\}$ defined by

$$f^{\oplus\oplus}(x^{(1)}, \ldots, x^{(d)}) = \bigoplus_{j=1}^{d} \bigoplus_{\ell=1}^{v} f^{(\ell)}\left(x^{(j)}\right), \text{ where } f^{(i^*)} \equiv f.$$

Then, G_N is rewritten by using $f^{\oplus\oplus}$ as

$$\begin{aligned}&G_N(f^{(1)}\circ\cdots\circ f^{(i^*-1)}\circ f\circ f^{(i^*+1)}\circ\cdots\circ f^{(v)}\circ x^{(1,1)}\circ\cdots\circ x^{(1,d)}\circ\cdots\circ x^{(vs(n)+1,1)}\circ\cdots\circ x^{(vs(n)+1,d)}\circ x_{left})\\&\quad= x^{(1,1)}\circ\cdots\circ x_{left}\circ f^{\oplus\oplus}(x^{(1,1)},\dots,x^{(1,d)})\circ\cdots\circ f^{\oplus\oplus}(x^{(vs(n)+1,1)},\dots,x^{(vs(n)+1,d)}).\end{aligned}$$

Now, we construct a learning algorithm $L^{\oplus\oplus}$ as Algorithm 4. Note that, for the sake of simplicity, we use the notation F to refer to $(f^{(1)},\dots,f^{(i^*-1)},f^{(i^*+1)},\dots,f^{(v)})$ below.

Algorithm 4 $L^{\oplus\oplus}$

Input : parameters $n\in\mathbb{N}$, $s(n)\in\mathbb{N}$, $v=\lceil n/\eta\rceil\in\mathbb{N}$, and $d\in\mathbb{N}$, examples $S=\{(x^{(1)},b^{(1)}),\dots,(x^{(vs(n))},b^{(vs(n))})\}$, and a challenge $x\leftarrow U_n^d$, where $f^{\oplus\oplus}:\{0,1\}^{dn}\to\{0,1\}$ (defined as above w.r.t. $F\leftarrow_u\{0,1\}^{s(n)(v-1)}$), and $(x^{(i)},b^{(i)})\leftarrow$EX$(f^{\oplus\oplus},U_n^d)$.
Output: $f^{\oplus\oplus}(x)$

```
19 let $N = N_{C,4}(n, s(n), v, d)$
20 select $x_{left} \leftarrow_u \{0,1\}^{N^4+2N^2-(dn(vs(n)+1)+vs(n))}$
21 select $i \leftarrow_u [vs(n)+1]$ and $c^*, c_i, \dots, c_{vs(n)} \leftarrow_u \{0,1\}$
22 let $X = x^{(1)} \circ \cdots \circ x^{(i-1)} \circ x \circ x^{(i)} \circ \cdots \circ x^{(vs(n))} \circ x_{left}$
23 execute $b \leftarrow A\left(1^N, X \circ b_1 \circ \cdots \circ b_{i-1} \circ c^* \circ c_i, \circ \cdots \circ c_{vs(n)}\right)$
24 **if** $b = 1$ **then return** $c^*$, **otherwise return** $1 - c^*$
```

Note that the following distribution H_0 is statistically identical to $U_{N^4+2N^2+1}$:

$$H_0 = G_N(U_{N^4+2N^4})_{[N^4+2N^4-vs(n)]}\circ U_{vs(n)+1}.$$

Thus, by applying the standard hybrid argument, we can show that (for a more detailed discussion, refer to Goldreich, 2006),

$$\Pr_{L^{\oplus\oplus},S,x,F}\left[L^{\oplus\oplus}(n,s(n),\lceil n/\eta\rceil,d,S,x)=f^{\oplus\oplus}(x)\right]\geq\frac{1}{2}+\frac{\gamma(n,s(n),\eta^{-1},d)}{2(\lceil n/\eta\rceil s(n)+1)}.$$

We can also easily check that $L^{\oplus\oplus}$ is executed in time $\mathsf{poly}(N_{C,4}(n,s(n),v,d))\leq\mathsf{poly}(n,s,\eta^{-1},d)$.

By applying the same algorithm in the proof of Theorem 32 with access to EX(f,U_n), we can translate the above $L^{\oplus\oplus}$ into a randomized hypothesis h satisfying the condition that

$$\Pr_{h,x\leftarrow U_n}[h(x)\neq f(x)]\leq\frac{1}{2}-\frac{1}{2}\left(\frac{\gamma(n,s(n),\eta^{-1},d)}{4(\lceil n/\eta\rceil s(n)+1)}\right)^{\frac{1}{d}},$$

with confidence probability at least

$$\frac{1}{v}\cdot\frac{\gamma(n,s(n),\eta^{-1},d)}{4(\lceil n/\eta\rceil s(n)+1)}\geq\mathsf{poly}^{-1}(n,s(n),\eta^{-1},d).$$

At this point, we assume that $d\in\mathbb{N}$ was selected to satisfy the condition that

$$d\geq\frac{1}{\epsilon}\cdot\left(\ln\gamma(n,s(n),\eta^{-1},d)^{-1}+\ln(\lceil n/\eta\rceil s(n)+1)+\ln 4\right). \tag{8}$$

Then, we can show that h is $\epsilon/2$-close to f as follows:

$$\begin{aligned}\Pr_{h,x\leftarrow D}[h(x)\neq f(x)] &\leq \frac{1}{2}-\frac{1}{2}\cdot\left(\frac{\gamma(n,s(n),\eta^{-1},d)}{4(\lceil n/\eta\rceil s(n)+1)}\right)^{\frac{1}{d}}\\ &\leq \frac{1}{2d}\cdot\left(\ln\gamma(n,s(n),\eta^{-1},d)^{-1}+\ln(\lceil n/\eta\rceil s(n)+1)+\ln 4\right) \qquad (\because \text{Claim 1})\\ &\leq \frac{\epsilon}{2}. \qquad (\because (8))\end{aligned}$$

To derandomize the hypothesis h, we simply select a binary string at random and embed it into h as h's randomness. By Markov's inequality, such an h is ϵ-close to f with probability at least $1/2$.

Because $\gamma^{-1}(n,s(n),\eta^{-1},d)\leq \mathsf{poly}(n,s(n),\eta^{-1},d)$, there exist a constant $c>0$ and a function $l(n,s(n),\eta)=O(\log n+\log s(n)+\log\eta^{-1})$ such that

$$\frac{c}{\epsilon}\ln d+\frac{l(n,s(n),\eta)}{\epsilon}\geq\frac{1}{\epsilon}\cdot\left(\ln\gamma(n,s(n),\eta^{-1},d)^{-1}+\ln(\lceil n/\eta\rceil s(n)+1)+\ln 4\right).$$

It is not hard to see that, for instance, the following choice of d is enough for inequality (8) to hold,

$$d=\max\left\{\left\lceil\frac{c}{\epsilon}\right\rceil^2,\left\lceil\frac{l(n,s(n),\eta)}{\epsilon}\right\rceil^2,49\right\}=O\left(\epsilon^{-2}(\log n+\log s(n)+\log\eta^{-1})\right).$$

Therefore, the above procedure is executed in time at most

$$\mathsf{poly}(n,s(n),\eta^{-1},d)\leq\mathsf{poly}(n,s(n),\epsilon^{-1},\eta^{-1}).$$

By applying the standard repeating and testing technique, arbitrary confidence δ can also be achieved in time

$$O\left(\mathsf{poly}(n,s(n),\epsilon^{-1},\eta^{-1})\cdot\ln\delta^{-1}\right)\cdot\mathsf{poly}(n,s(n),\epsilon^{-1},\eta^{-1})\leq\mathsf{poly}(n,s(n),\epsilon^{-1},\eta^{-1})\cdot\ln\delta^{-1}.$$

In the remaining case where $\gamma(n,s(n),\eta^{-1},d)\leq 2e^{-n}$, we have that for the above choice of d,

$$2^n<e^n\leq 2\cdot\gamma(n,s,\eta^{-1},d)^{-1}\leq\mathsf{poly}(n,s(n),\epsilon^{-1},\eta^{-1}).$$

Therefore, as in the proof of Theorem 32, the learner can directly approximate a truth table of an arbitrary target function with probability at least $1-\delta$ in time $\mathsf{poly}(n,s(n),\epsilon^{-1},\eta^{-1})\cdot\ln\delta^{-1}$. ■

D.3. Corollaries of Theorems 6 and 8

In this section, we present Corollaries 9–12 to Theorems 6 and 8. To show Corollary 9, we will apply Corollary 11. Thus, we will present the formal proofs for Corollaries 10 and 11 first, then Corollaries 9 and 12.

D.3.1. PROOF OF COROLLARY 10

(3$\Longrightarrow$2) holds obviously, and (2$\Longrightarrow$1) follows from Theorem 6.

For (1$\Longrightarrow$3), assume that AIOWF exists. Then, by Fact 1, AIPRF $F = \{F_z : \{0,1\}^{|z|} \times \{0,1\}^{|z|} \to \{0,1\}\}_{z \in \{0,1\}^*}$ also exists.

Now, we determine a concept class $\mathscr{C}$ by the following evaluation $C = \{C_n : \{0,1\}^{\lceil n/2 \rceil} \times \{0,1\}^n \to \{0,1\}\}_{n \in \mathbb{N}}$: for each $n \in \mathbb{N}$,

$$C_{2n-1}(u, y) = 1$$
$$C_{2n}(u, z \circ x) = F_z(u, x),$$

where $u \in \{0,1\}^n$, $z, x \in \{0,1\}^n$, and $y \in \{0,1\}^{2n-1}$.

Then $\mathscr{C}$ is obviously polynomial-time evaluated. Now, we show that $\mathscr{C}$ is not weakly heurPAC learnable by contradiction. By Theorem 22, we assume that there exists a weak predictor L for $\mathscr{C}$ with an error parameter $\epsilon \leq 1/2 - \mathsf{poly}^{-1}(n)$.

We construct an adversary A for F based on P as follows: on input $z \in \{0,1\}^n$ and query access to a function $f : \{0,1\}^n \to \{0,1\}$ (which is either of a pseudorandom function or a truly random function), A executes P, where A passes $(z \circ x, f(x))$ as each example and $z \circ x^*$ as a challenge for independently selected $x, x^* \leftarrow_u \{0,1\}^n$. If P outputs some prediction b, then A checks whether $f(x^*) = b$ by its own oracle access. If P succeeds in predicting, A outputs 1, otherwise, it outputs 0. Thus, the probability that A outputs 1 is exactly the same as the probability that P succeeds in predicting over the choice of f and randomness for A. Obviously, A halts in polynomial-time in n.

On one hand, consider the case where f is selected from pseudorandom functions. In this case, it is not hard to check that A executes P in the valid setting where the target function is f and the example distribution is $z \circ U_n$. Thus, A outputs 1 with probability at least $1/2 + 1/\mathsf{poly}(n)$.

On the other hand, consider the case where f is selected from all n-input functions, i.e., a truly random function. Because P looks at only $\mathsf{poly}(n)$ values of f, randomly selected x^* is contained in previous examples with only negligibly small probability. Additionally, if x^* is not contained in the previous examples, then P cannot guess the value of $f(x^*)$ better than at random because f is a truly random function. Thus, A outputs 1 with probability at most $1/2 + \mathsf{negl}(n)$, where negl denotes some negligible function.

Therefore, A distinguishes the above two cases with non-negligible probability, which contradicts the assumption that F is AIPRF.

D.3.2. PROOF OF COROLLARY 11

(5$\Longrightarrow$4), (5$\Longrightarrow$3), and (4$\Longrightarrow$2) hold obviously, and (2$\Longrightarrow$1) follows from Theorem 8.

The remaining part (1$\Longrightarrow$5) mainly follows from the observation by Valiant (1984). Assume that io-OWF exists, then io-PRF $F = \{F_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}\}_{n \in \mathbb{N}}$ also exists from Fact 1.

Let $\mathscr{C}$ be a concept class with an evaluation $C = \{C_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}\}_{n \in \mathbb{N}}$ defined by $C_n(u, x) = F_n(u, x)$. Note that $\mathscr{C}$ can obviously be evaluated in polynomial-time. Thus, we only need to show that $\mathscr{C}$ is not weakly heurPAC learnable with membership queries on the uniform distribution by contradiction. From Theorem 22 (for learning with membership queries[8]), we can assume that there exists a weak predictor P for $\mathscr{C}$ with an error parameter $\epsilon \leq 1/2 - \mathsf{poly}^{-1}(n)$.

We can construct an adversary A for F based on P as follows: on input 1^n and query access to a function $f : \{0,1\}^n \to \{0,1\}$ (which is either of a pseudorandom function or a truly random function), A executes P where the target function is f and the example distribution is U_n. Note that A can correctly answer all P's queries by its own query access to f. If P outputs a prediction b for a challenge x^*, then A checks whether $f(x^*) = b$. If P succeeds in predicting, A outputs 1, otherwise, A outputs 0. Obviously, A halts in polynomial-time in n.

The following argument is almost the same as the one in Appendix D.3.1. If f is selected from pseudorandom functions, then A can execute P properly and output 1 with probability at least $1/2 + 1/\mathsf{poly}(n)$. On the other hand, if f is selected from truly random functions, then a polynomial-time predictor P cannot succeed in predicting non-negligibly better than at random. Thus, A outputs 1 with probability at most $1/2 + \mathsf{negl}(n)$, where negl denotes some negligible function. Therefore, A distinguishes the above two cases with non-negligible probability, which contradicts the assumption that F is io-PRF.

D.3.3. PROOF OF COROLLARY 9

Let $\mathscr{C}$ be a concept class with a polynomial-time evaluation $C : \{0,1\}^{s(n)} \times \{0,1\}^n \to \{0,1\}$ and assume that $\mathscr{C}$ is not heurPAC learnable on a polynomial-time sampleable distribution D. W.l.o.g., we can also assume that $s(\cdot)$ is an increasing function. By Theorem 8, we only need to construct another concept class $\mathscr{C}'$ which is not heurPAC learnable on the uniform distribution.

Let M be the sampling algorithm for D, i.e., $M(1^n)$ is statistically identical to D_n for each $n \in \mathbb{N}$. W.l.o.g., we can assume that $M(1^n)$ uses $p(n)$ random bits for some increasing polynomial p such that $p(n) \geq n$. Then, we determine $\mathscr{C}'$ by the following evaluation $C' : \{0,1\}^{s(n)} \times \{0,1\}^n \to \{0,1\}$:

$$C'(u,x) = \begin{cases} C(u_{[s(n')]}, M(1^{n'}; x)) & \text{if } n = p(n') \text{ for some } n' \in \mathbb{N} \\ 1 & \text{otherwise.} \end{cases}$$

We also define a function $f = \{f_N : \{0,1\}^{m \cdot p(n)} \to \{0,1\}^{nm}\}_{N \in \mathbb{N}}$ where $(n,m) = N_C^{-1}(N)$ as follows: for each $N \in \mathbb{N}$,

$$f_N(x) = M(1^n; x^{(1)}) \circ \cdots \circ M(1^n; x^{(m)})$$

where $x = x^{(1)} \circ \cdots \circ x^{(m)}$ and $|x^{(i)}| = p(n)$ for each $i \in [m]$.

Notice that $\mathscr{C}'$ is evaluated in time $\mathsf{poly}(p^{-1}(n), s(p^{-1}(n))) \leq \mathsf{poly}(n, s(n))$, and f is polynomial-time computable.

We only consider the case where f is not an io-DistOWF, otherwise there exists another concept class which is not heurPAC learnable on the uniform distribution by Fact 1 and Corollary 11. Therefore, we additionally assume that there exists a polynomial-time randomized algorithm I such that the statistical distance between following two distributions is at most $1/4$: for any $n, m \in \mathbb{N}$ and target function $u \in \{0,1\}^{s(n)}$,

8. In the prediction with membership query (pwm) model formally introduced by Angluin and Kharitonov (1995), a predictor is given examples and a challenge one by one, and the membership query oracle is also available only before the learner gets the challenge. It is easily checked that the proof of Theorem 22 holds even if the membership query oracle is available.

1. $\left(\left(U^{(1)}_{p(n)},\ldots,U^{(m)}_{p(n)}\right),\left(x^{(1)},\ldots,x^{(m)}\right),\left(C(u,x^{(1)}),\ldots,C(u,x^{(m)})\right)\right)$; and
2. $\left(I\left(1^{N_C(n,m)},x^{(1)}\circ\cdots\circ x^{(m)}\right),\left(x^{(1)},\ldots,x^{(m)}\right),\left(C(u,x^{(1)}),\ldots,C(u,x^{(m)})\right)\right)$,

where $x^{(i)}=M\left(1^n;U^{(i)}_{p(n)}\right)$ for each $i\in[m]$.

Now, we show that $\mathscr{C}'$ is not heurPAC learnable on the uniform distribution by contradiction. Assume that L' is a heurPAC learner for $\mathscr{C}'$ which works on the uniform example distribution. Then we construct a heurPAC learner L for $\mathscr{C}$ which works on D.

On given parameters n,ϵ,δ,η, we let L execute L' with parameters $n'=p^{-1}(n),\epsilon'=\epsilon,\delta'=1/4$, and $\eta'=\eta$. Let m $(\leq \mathsf{poly}(n,\epsilon^{-1},\eta^{-1}))$ be the number of examples sufficient to execute L'. By the construction of C', a distribution on target functions for C' corresponds to a distribution on target functions for C, and a random choice of input for C' (according to the uniform distribution) corresponds to a random choice of input for C (according to D). Thus, if random seeds for M are available in addition to outputs of M (i.e., examples selected according to D), then L can validly simulate examples for learning $\mathscr{C}'$. However, such random seeds for M are unavailable for L.

Therefore, L first applies the inverter I with the security parameter $N_C(n,m)$ to obtain the random seeds, then executes L' by regarding the inverse elements as inputs in examples for learning $\mathscr{C}'$ on the uniform distribution. Since the statistical distance between true examples and examples generated by I as above is at most $1/4$ over the choice of examples for $\mathscr{C}'$ and random seeds for I, the loss of confidence probability is at most $1/4$. Thus, L can learn $\mathscr{C}$ with confidence error at most $1/4+1/4=1/2$, which can be improved to arbitrary δ by applying the standard repeating and testing method. Therefore, $\mathscr{C}$ is heurPAC learnable on D, which contradicts the assumption.

D.3.4. PROOF OF COROLLARY 12

We consider the nonuniform computational model in this section.

In this paper, we do not present a formal description of the Universality Conjecture to avoid introduction of several additional notions. For the details, refer to the original paper (Santhanam, 2020). Thus, we introduce the following as a fact.

Fact 5 (Santhanam, 2020, Theorem 22) *Under the Universality Conjecture, the followings are equivalent:*

1. *AIOWF exists;*
2. *n^2-sized circuits are not PAC learnable with membership queries on the uniform distribution; and*
3. *io-OWF exists.*[9]

Note that Theorems 6 and 8 hold even in the nonuniform model by the same proof. This is because a learner can get polynomial-length advice to execute an adversary and evaluate a concept class as its own advice. By Theorems 6 and 8, items 1 and 3 in Corollary 12 correspond to items 1 and 3 in Fact 5, respectively.

Thus, the remaining part is to show the correspondence between item 2 in Corollary 12 and item 2 in Fact 5, which can be stated as follows:

9. In his work, the equivalence also holds even for the standard one-way function with "sufficiently large" security.

Claim 2 *The followings are equivalent:*

1. *n^2-sized circuits are not PAC learnable with membership queries on the uniform distribution and*

2. *there exists a concept class $\mathscr{C}$ with a polynomial-time evaluation $C : \{0,1\}^{s(n)} \times \{0,1\}^n \to \{0,1\}$ such that $\mathscr{C}$ is not PAC learnable with membership queries on the uniform distribution.*

Proof For the sake of simplicity, we will omit the argument on advice for nonuniform computation.

(1)$\Longrightarrow$(2) obviously holds.

(2)$\Longrightarrow$(1) is shown by the simple padding argument. Assume that C is computable by a circuit of size $p(n, s(n))$ where p is a polynomial. For each target function $f \in \{0,1\}^{s(n)}$ in $\mathscr{C}$, we define a function $f' : \{0,1\}^{n+\sqrt{p(n,s(n))}} \to \{0,1\}$ by $f'(x) = f(x_{[n]})$. Then f' must be computable by a circuit of size $p(n, s(n)) \leq \left(n + \sqrt{p(n, s(n))}\right)^2$. Thus, any learner for n^2-sized circuits successfully PAC learns f' is time $\mathsf{poly}(n + \sqrt{p(n, s(n))}, \epsilon^{-1}, \delta^{-1}) \leq \mathsf{poly}(n, s(n), \epsilon^{-1}, \delta^{-1})$. Note that an example oracle $\mathrm{EX}(f', U_{n+\sqrt{p(n,s(n))}})$ and a membership oracle for f' are trivially simulated by $\mathrm{EX}(f, U_n)$ and a membership oracle for f, respectively. Thus, we can construct a learner that PAC learns $\mathscr{C}$ with membership queries based on a learner that PAC learns n^2-sized circuits with membership queries. ■