Learning algorithms from circuit lower bounds

Ján Pich

University of Oxford

June 2024

Abstract

We revisit known constructions of efficient learning algorithms from various notions of constructive circuit lower bounds such as distinguishers breaking pseudorandom generators or efficient witnessing algorithms which find errors of small circuits attempting to compute hard functions. As our main result we prove that if it is possible to find efficiently, in a particular interactive way, errors of many p-size circuits attempting to solve hard problems, then p-size circuits can be PAC learned over the uniform distribution with membership queries by circuits of subexponential size. The opposite implication holds as well. This provides a new characterisation of learning algorithms and the natural proofs barrier of Razborov and Rudich. The proof is based on a method of reconstructing Nisan-Wigderson generators introduced by Krajíček (2010) and used to analyze complexity of circuit lower bounds in bounded arithmetic.

An interesting consequence of known constructions of learning algorithms from circuit lower bounds is a learning speedup of Oliveira and Santhanam (2016). We present an alternative proof of this phenomenon and discuss its potential to advance the program of hardness magnification.

1 Introduction

While the central conjectures in complexity theory such as $\mathsf{P} \neq \mathsf{NP}$ have the form of impossibility results, we hope that a better understanding of the impossibility phenomena will also shed light on the question of constructing new useful algorithms. A successful formalization of such hopes can be found in cryptography, where the impossibility results in the form of average-case lower bounds are turned into cryptographic primitives. In the present paper we are interested in turning complexity lower bounds into efficient learning algorithms.

Results of this form can be traced back to cryptography as well. The '*pseudorandomness from unpredictability*' paradigm was used by Blum, Furst, Kearns and Lipton [5] to show that efficient distinguishers breaking pseudorandom generators imply an efficient learning of p-size circuits on average. The distinguishers from [5] can be interpreted as constructive circuit lower bounds distinguishing partial truth-tables of easy Boolean functions from partial truth-tables of hard functions, cf. Section 4. The existing methods for proving circuit lower bounds have been also applied in constructions of new learning algorithms for restricted circuit classes, e.g. Linial, Mansour and Nisan [31] used AC^0 lower bounds to get learning algorithms for AC^0. More recently, in a landmark work, Carmosino, Impagliazzo, Kabanets and Kolokolova [7] gave a generic construction of learning algorithms from natural proofs of circuit lower bounds. Oliveira and Santhanam [41] extended their result to a dichotomy between the non-existence of non-uniform pseudorandom function families and the existence of efficient learning of small circuits. These results led Oliveira and Santhanam [41] also to a discovery of a surprising learning speedup. For example, learning p-size circuits over the uniform distribution with membership queries by circuits of weakly subexponential size $2^n/n^{\omega(1)}$ implies that for each constant k and $\epsilon > 0$, circuits of size n^k can be learned over the uniform distribution with membership queries by circuits of strongly subexponential size 2^{n^ϵ}.

1.1 Our contribution

In the present paper we revisit these connections. We start by considering a simple *instance-specific* model of learning in which proving a single circuit lower bound implies a reliable prediction of the value of a target function on a single input. The model can be intuitively seen as a very basic version of the construction of learning algorithms from [5, 7] and differs from the standard PAC learning model mainly in that it does not ask learners to construct a circuit which computes the target function on a big fraction of inputs, cf. Section 3.1.

Learning from witnessing lower bounds. Our main result is a construction of efficient PAC learning of p-size circuits from a constructive circuit lower bound for an arbitrary Boolean function H. More precisely, we obtain subexponential-size circuits learning p-size circuits over the uniform distribution with membership queries. The assumption of a constructive circuit lower bound we need is defined as the existence of $2^{O(n)}$-size 'witnessing' circuits W which given an oracle access to a p-size circuit D with n inputs find a not-yet-queried input on which D fails to compute H. The circuits W are allowed to fail on $1/poly(n)$ fraction of circuits D. Moreover, even if circuits W succeed on a circuit D they are allowed to output incorrect answer $\log n$ times (receiving a correction from a helpful counterexample oracle in each round) before generating the right answer, cf. Theorem 5. The implication can be also interpreted as a construction of PAC learning algorithms from a frequent interactive instance-specific[1] learning: If we are given an al-

[1] We use the adjective 'instance-specific' only informally in this paper. The instance-specific model discussed earlier actually differs slightly from the concept in Theorem 5.

gorithm which is able to predict a value of a big fraction of p-size circuits (after a small number of queries and $\leq \log n$ mistakes) even on a single input, this already implies learnability of p-size circuits over the uniform distribution on almost all inputs. The opposite implication producing efficient witnessing of lower bounds from learning algorithms holds as well, which yields a new characterisation of PAC learning of small circuits over the uniform distribution, cf. Lemma 7.

Relation to proof complexity, natural proofs and witnessing theorems. The notion of interactive witnessing of circuit lower bounds from Theorem 5 is inspired by witnessing theorems in bounded arithmetic. One of the most prominent theories of bounded arithmetic is Cook's theory PV_1 formalizing p-time reasoning. Theories of bounded arithmetic satisfy many kinds of witnessing theorems which show, for example, that if we can prove a p-size circuit lower bound for a function $H \in \mathsf{NP}$ in PV_1, then there exists a witnessing analogous to the one from Theorem 5 except that the witnessing circuits W have white-box access to D (i.e. they have access to a full description of D), see Section 3.2 for a more detailed comparison.[2] The witnessing from Theorem 5 is also closely related to algorithms finding hard instances of NP problems by Gutfreund, Shaltiel, Ta-Shma [15] and Atserias [3]. The main difference is that the algorithms from [15] have white-box access to the algorithm whose error they search for. While Atserias [3] made [15] work with the black-box (oracle) access, his algorithm achieves much smaller probability of success than the one required in Theorem 5, cf. Section 3.2.

The proof of Theorem 5 is an adaptation of a method of exploiting Nisan-Wigderson generators introduced by Krajíček [23] in order to give a model-theoretic evidence for Razborov's conjecture in proof complexity. Razborov's conjecture [49] states a conditional hardness of deriving tautologies expressing the existence of an element outside of the range of a suitable NW-generator in strong proof systems. Krajíček's result significantly strengthens a similar but much simpler proof of the validity of Razborov's conjecture for proof systems with feasible interpolation [43]. The method has been also used to show a conditional hardness of generating hard tautologies [26], a conditional unprovability of p-size circuit lower bounds for SAT in theories of bounded arithmetic below PV_1 [44] and an unconditional unprovability of strong nondeterministic lower bounds in PV_1 [46]. We take advantage of its unique way of exploiting the NW generator: it gives us a reconstruction algorithm which after breaking the NW-generator in a particular interactive fashion allows us to approximately compute the function on which the generator is based. There are, however, technical issues with adapting this method in our context. For example, unlike in previous applications of Krajíček's reconstruction of the NW-generator, our witnessing circuits can fail with a significant probability. Our main contribution is in finding the right notions which allow the arguments to go through (in both directions).

A competing notion of constructive circuit lower bounds has been developed in the

[2] It is not clear how to define a theory of bounded arithmetic so that the witnessing algorithm resulting from the provability of circuit lower bounds in the theory would match the witnessing in Theorem 5.

influential theory of natural proofs of Razborov and Rudich [50], which explains why many of the existing lower bound methods cannot yield separations such as $\mathsf{P} \neq \mathsf{NP}$ without disproving cryptographic conjectures. Natural proofs are known to be equivalent to the existence of efficient learning algorithms, cf. [7]. For example, $\mathsf{P/poly}$-natural proofs useful against $\mathsf{P/poly}$[3] are equivalent to subexponential-size circuits learning p-size circuits over the uniform distribution with membership queries. Furthermore, natural proofs have been used to derive unprovability results in proof complexity as well. Specifically, to derive unprovability of circuit lower bounds in proof systems with the feasible interpolation property, cf. [48, 22]. Despite similar applications and motivations for defining these concepts, the relation between natural proofs and the witnessing method has not been clear. In fact, a priori the 'static' definition of natural proofs (which inspect the whole truth-table) appears to be quite orthogonal to the witnessing from Theorem 5 (which prints a single error of a small candidate circuit). Theorem 5 thus presents a bridge between two seemingly different notions of constructivizing circuit lower bounds. It thus not only extends the scope of the natural proofs barrier by providing another equivalent characterisation which incorporates interactivity but also helps to clarify its relation to the witnessing method in bounded arithmetic.

Learning speedup. Our second contribution is a simple proof of a generalized learning speedup of Oliveira and Santhanam [41]. Specifically, we show that for each superpolynomial function s, if for each constant k, circuits of size n^k are learnable by circuits of size s over the uniform distribution with random examples, then for each constant k and $\epsilon > 0$, circuits of size n^k are learnable over the uniform distribution with membership queries by circuits of size $O(s^\epsilon)$, cf. Theorem 19. We obtain the speedup by a more direct exploitation of a slightly modified NW-generator. In comparison to the proof from [41], this sidesteps the need to construct natural proofs and invoke the construction of Carmosino et al. [7]. A disadvantage of the method is that we need to assume learning with random examples instead of membership queries. Nevertheless, we present one more alternative proof of the learning speedup based on (a simple case of) Theorem 5, which allows to start with membership queries, cf. Theorem 20. We emphasize, however, that behind all proofs of the learning speedup is essentially the same general idea of reconstructing, in this or that way, the base function of some form of the NW-generator.

Relation to hardness magnification and locality. The generalized learning speedup can be interpreted as a 'nonlocalizable' hardness magnification theorem reducing a complexity lower bound into a seemingly weaker one. In general, hardness magnification refers to an approach to strong complexity lower bounds developed in a series of recent papers, cf. Section 5. Unfortunately, while the approach avoids (in certain cases provably [8])

[3] $\mathsf{P/poly}$-natural proofs useful against $\mathsf{P/poly}$ are defined as $2^{O(n)}$-size circuits with 2^n inputs accepting a $1/2^{O(n)}$-fraction of inputs and rejecting all inputs which represent truth-tables of Boolean functions on n inputs computable by p-size circuits, cf. Definition 1.

the natural proofs barrier, it suffers from a '*locality barrier*': magnification theorems typically yield unconditional upper bounds for specific problems if the computational model in question is allowed to use oracles with small fan-in (local oracles), but the existing lower bounds actually work even against the presence of local oracles. In fact, a better understanding of nonlocalizable lower bounds (i.e. lower bounds which do not remain valid in the presence of local oracles) is essential for further progress on strong complexity lower bounds in general, see Section 5 for more details. A promising aspect of the learning speedup (Theorem 19) is that it avoids the locality barrier, cf. Section 5.

Learning from breaking cryptographic pseudorandom generators. In Section 4 we survey known constructions of learning algorithms from various ways of breaking pseudorandom generators (PRGs). While several such constructions are known, the question of extracting efficient learning of p-size circuits from the mere non-existence of cryptographic PRGs remains open. A positive answer to this question would establish an interesting win-win situation: either safe cryptography or efficient learning is possible. In the already mentioned approach, Oliveira and Santhanam [41] showed that efficient learning of p-size circuits with membership queries follows from the non-existence of nonuniform pseudorandom function families. By a straightforward adaptation of the proof method behind their result we show that efficient learning of p-size circuits *with random examples* follows from the non-existence of succinct nonuniform pseudorandom function families, cf. Theorem 15. Finally, we point out that the desired construction of learning algorithms from the non-existence of cryptographic PRGs is closely related to a question of Rudich about turning demibits to superbits, cf. Section 4.4.

2 Preliminaries

$[n]$ denotes $\{1, \dots, n\}$. $\mathsf{Circuit}[s]$ denotes fan-in two Boolean circuits of size at most s. The size of a circuit is the number of gates. A function $f : \{0,1\}^n \mapsto \{0,1\}$ is γ-approximated by a circuit C, if $\Pr_x[C(x) = f(x)] \geq \gamma$.

Definition 1 (Natural property [50])**.** *Let $m = 2^n$ and $s, d : \mathbb{N} \mapsto \mathbb{N}$. A sequence of circuits $\{C_m\}_{m=1}^{\infty}$ is a* $\mathsf{Circuit}[s(m)]$-natural property useful against $\mathsf{Circuit}[d(n)]$ *if*

1. Constructivity. *C_m has m inputs and size $s(m)$,*
2. Largeness. $\Pr_x[C_m(x) = 1] \geq 1/m^{O(1)}$,
3. Usefulness. *For each sufficiently big m, $C_m(x) = 1$ implies that x is a truth-table of a function on n inputs which is not computable by circuits of size $d(n)$.*

Definition 2 (Pseudorandom generator)**.** *A function $g : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ computable by p-size circuits is a* pseudorandom generator safe against circuits of size $s(n)$*, if for each circuit D of size $s(n)$,*

$$\left| \Pr_{y \in \{0,1\}^{n+1}}[D(y) = 1] - \Pr_{x \in \{0,1\}^n}[D(g(x)) = 1] \right| < \frac{1}{s(n)}.$$

Definition 3 (PAC learning)**.** *A circuit class $\mathcal{C}$ is* learnable over the uniform disribution by a circuit class $\mathcal{D}$ up to error ϵ with confidence δ*, if there are randomized oracle circuits L^f from $\mathcal{D}$ such that for every Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$ computable by a circuit from $\mathcal{C}$, when given oracle access to f, input 1^n and the internal randomness $w \in \{0,1\}^*$, L^f outputs the description of a circuit satisfying*

$$\Pr_w[L^f(1^n, w) \ (1-\epsilon)\text{-}approximates\ f] \geq \delta.$$

L^f uses non-adaptive membership queries *if the set of queries which L^f makes to the oracle does not depend on the answers to previous queries. L^f uses* random examples *if the set of queries which L^f makes to the oracle is chosen uniformly at random.*

In this paper, PAC learning always refers to learning over the uniform distribution. While, a priori, learning over the uniform distribution might not reflect real-world scenarios very well (and on the opposite end, learning over all distributions is perhaps overly restrictive), as far as we can tell it is possible that PAC learning of p-size circuits over the uniform distribution implies PAC learning of p-size circuits over all p-samplable distributions. Binnendyk, Carmosino, Kolokolova, Ramyaa and Sabin [4] proved the implication, if the learning algorithm in the conclusion is allowed to depend on the p-samplable distribution.

Boosting confidence and reducing error. The confidence of the learner can be efficiently boosted in a standard way. Suppose an s-size circuit L^f learns f up to error ϵ with confidence δ. We can then run L^f k times, test the output of L^f from every run with m new random queries and output the most accurate one. By Hoeffding's inequality, m random queries fail to estimate the error ϵ of an output of L^f up to γ with probability at most $2/e^{2\gamma^2 m}$. Therefore the resulting circuit of size $poly(s, m, k)$ learns f up to error $\epsilon+\gamma$ with confidence at least $1 - 2k/e^{2\gamma^2 m} - (1-\delta)^k \geq 1 - 2k/e^{2\gamma^2 m} - e^{-k\delta}$. If we are trying to learn small circuits we can get even confidence 1 by fixing the internal randomness of the learner nonuniformly without losing much on the running time or the error of the output.

It is also possible to reduce the error up to which L^f learns f without a significant blowup in the running time and confidence. If we want to learn f with a better error, we first learn an amplified version of f, $Amp(f)$. Employing direct product theorems and Goldreich-Levin reconstruction algorithm, Carmosino et. al. [7, Lemma 3.5] showed that for each $0 < \epsilon, \gamma < 1$ it is possible to map a Boolean function f with n inputs to a Boolean function $Amp(f)$ with $poly(n, 1/\epsilon, \log(1/\gamma))$ inputs so that $Amp(f) \in \mathsf{P/poly}^f$ and there is a probabilistic $poly(|C|, n, 1/\epsilon, 1/\gamma)$-time machine which given a circuit C $(1/2+\gamma)$-approximating $Amp(f)$ and an oracle access to f outputs with high probability a circuit $(1-\epsilon)$-approximating f. We can thus often ignore the optimisation of the confidence and error parameter. Note, however, that the error reduction of Carmosino et al. requires membership queries.

3 Instance-specific learning

In Section 3, we consider two types of 'instance-specific' learning. While in Theorem 5 we show that the second type coincides with the standard PAC learning, the first type presented in Section 3.1 is more rudimentary and its connection to PAC learning is less clear.

3.1 Learning from individual circuit lower bounds

The most direct way of turning circuit lower bounds into a certain type of learning can be described as follows.

A. Prediction from lower bound. Suppose we are given bits $f(y_1), \dots, f(y_k)$ for n-bit strings $y_1, \dots, y_k$ defining a partial Boolean function f. We want to predict the value of f on a new input $y_{k+1} \in \{0,1\}^n$. A priori $f(y_{k+1})$ is not defined but we will interpret the minimal-size circuit C^f coinciding with f on $y_1, \dots, y_k$ as 'the right' prediction of $f(y_{k+1})$. That is, we want to find $C^f(y_{k+1})$. Here, we assume that the minimal circuit C^f determines the value $f(y_{k+1})$. Otherwise, there are two circuits C^1, C^2 of minimal size such that $C^1(y_{k+1}) \neq C^2(y_{k+1})$, and therefore any prediction is equally good. Say that the size of the minimal circuit C^f is s (here we assume that we know s). Then the task to predict the value $C^f(y_{k+1})$ can be formulated as the task to prove an s-size circuit lower bound of the form

$$\forall \text{ circuit } C \text{ of size } s, \bigvee_{i=1,\dots,k} C(y_i) \neq f(y_i) \vee C(y_{k+1}) \neq \epsilon$$

for $\epsilon = 0$ or $\epsilon = 1$.

The simple observation from box A appeared in [36, Section 4.5] and [45]. We are not aware of a more systematic treatment of this concept. There are related models of learning such as 'knows what it knows' model by Li-Littman-Walsh [30] and 'reliable learning' by

Rivest-Sloan [51] which prohibit incorrect predictions in various ways.[4] These models, however, follow the formalization of PAC learning in that the goal of the learner is to learn the target concept by accessing it. In box A we do not assume that the target concept f is determined on all inputs or prior to the given samples.

An interesting aspect of the prediction method described in box A is that by proving even a single circuit lower bound we can learn something about the function f (if we know the value s). More precisely, we predict C^f on a single input but do not necessarily gain knowledge of the values of C^f on other inputs. This instance-specific learning should be contrasted with PAC learning, Definition 3, where one is required to generate a circuit predicting the target function f on most inputs. This, however, does not mean that it is easier to learn in the sense of box A: in Definition 3 we do not need to recognize when the prediction errs while the prediction from box A is zero-error in the sense that it guarantees to output the right value of $C^f(y_{k+1})$.

Provability vs truth. The definition of 'the right' prediction in terms of minimal circuits used in box A can be interpreted as an implicit (alternative) definition of truth. Consider, for example, that strings y_j encode statements in set theory ZFC and the value $f(y_j)$ is 1 if and only if the statement encoded by y_j is provable in ZFC. It would be interesting to find out whether the minimal circuit coinciding with a sufficiently rich list of such samples $(y_j, f(y_j))$ determines a truth value of the Continuum Hypothesis or of the consistency of ZFC, statements which are independent of ZFC. Unfortunately, in general, such questions seem to be out of reach of the contemporary mathematics.

Determining minimal circuit size. A drawback of the observation in box A is that it requires knowledge of the size s of the minimal circuit C^f. The size s could be determined by deciding s t-size circuit lower bounds for all $t \in [s]$, but this can be a hard task. Perhaps a more practical way of addressing the issue is to take instead of s a sufficiently big s' (intended to approximate s), choose a random $t \in [s']$ and prove t-size lower bounds (as in box A with t instead of s). If $s' \leq n^{O(1)}$, the probability that we have the right $t = s$ is $1/n^{O(1)}$. Informally speaking, if we want to get an approximation of the accuracy of the predictions obtained in this way, it suffices to solve polynomially many t-size lower bounds (in order to predict $C^f(y)$ on polynomially many y's). If the accuracy is not high, we can repeat the process with a new random $t \in [s']$. The advantage of the resulting prediction method is that it does not rely on deciding correctly whether some particular t-size circuit lower bounds hold - we are actually allowed to err on some fraction of lower bounds. However, its predictions are no longer zero-error. A closely related argument is formalized in Theorem 11.

Proof complexity. The prediction method from box A relies on proof complexity of

[4] In a recent paper, Amit et al. [2] address the reliability issue using interactive protocols.

circuit lower bounds, cf. [27].[5] It would be interesting to find out if proving circuit lower bounds in standard proof systems suffices to construct learning circuits.

Question 4 (Learning interpolation). *Is there a p-time function which given an Extended Frege proof of a formula $\bigvee_{y\in A} C(y) \neq f(y) \vee C(x) \neq \epsilon$, for $\epsilon = 0$ or $\epsilon = 1$, with free variables representing s-size circuits C with n inputs, a fixed set A of n-bit inputs of a sufficiently big size $|A| = poly(s,n)$, a fixed n-bit string $x \notin A$ and values of $f \in \mathsf{Circuit}[s]$ on A, outputs a circuit $(1/2+1/n)$-approximating f?*

3.2 Learning from witnessing lower bounds

We now give a construction of PAC learning algorithms from an interactive witnessing of circuit lower bounds. As discussed in the introduction, the implication can be also interpreted as a construction of PAC learning algorithms from a frequent interactive instance-specific learning.

Theorem 5 (Learning from interactive witnessing of lower bounds). *Let $d \geq 2; k, K \geq 1$ and H be a Boolean function with n inputs. Assume there are 2^{Kn}-size circuits*

$$\{W^j_t\}_{j\in[2^{Kn}],t\in[\lfloor \log n\rfloor]}$$

such that for each distribution $\mathcal{R}$ on n^{10dk}-size circuits with n inputs there exists $j \in [2^{Kn}]$ such that circuits $W^j_1,\dots,W^j_{\lfloor \log n\rfloor}$ witness errors of n^{10dk}-size circuits attempting to compute H in the following way.

> *Given black-box oracle access to a random n^{10dk}-size circuit $D(x)$ with n inputs, with probability at least $1-3/n^3$ over $\mathcal{R}$, the following interactive protocol succeeds: After querying values of circuit D, W^j_1 outputs a not-yet-queried $x_1 \in \{0,1\}^n$ s.t. $D(x_1) \neq H(x_1)$ or W^j_2 receives a correction in the form of bits $D(x_1), H(x_1)$ s.t. $D(x_1) = H(x_1)$. Having $D(x_1), H(x_1)$ and the examples queried by W^j_1, W^j_2 makes further queries to D and generates the second not-yet-queried candidate $x_2 \in \{0,1\}^n$ for the claim $D(x_2) \neq H(x_2)$. If $D(x_2) = H(x_2)$, W^j_3 receives a correction and the protocol continues in this way until some W^j_t, for some $t \leq \log n$, with access to all previous corrections and examples, finds the right x_t which has not been queried by $W^j_1,\dots,W^j_t$ and witnesses $D(x_t) \neq H(x_t)$.*

[5] Notably, Razborov [49] established that weak proof systems such as Resolution operating with k-DNFs for small k do not have polynomial-size proofs of any superpolynomial circuit lower bound whatsoever and he conjectured this holds under a hardness assumption even for stronger systems such as Frege. The issue is, however, delicate because proof systems like Extended Frege are already capable of formalizing a lot of complexity theory, see e.g. [36], and it is perfectly plausible that if a circuit lower bound is provable at all, then it is efficiently provable in Extended Frege.

Then, circuits of size n^{dk} with n^d inputs can be learned by circuits of size $2^{K'n}$ over the uniform distribution with non-adaptive membership queries, with confidence $1/2^{K'n^2}$ up to error $1/2 - 1/2^{K'n^2}$, where K' is a constant depending only on K.

Note that the witnessing circuits from Theorem 5 can work for arbitrary function H and, for the circuits D on which the witnessing succeeds, the number of queries in each round is implicitly bounded by $< 2^n$ (since after querying D on all inputs it would be impossible to output a not-yet-queried input).

It is not necessary to determine the precise form of inputs of circuits W_t^j. The inputs of W_t^j include examples queried by circuits $W_{t'}^j$, for $t' < t$, and the tuples of corrections from rounds $t' < t$. The examples received by W_t^j include the results of all queries made by $W_{t'}^j$, for $t' < t$. If these results were not included, W_t^j could get them by simulating $W_{t'}^j$, for $t' < t$. W_2^j does not receive the candidate error $x_1 \in \{0,1\}^n$ generated by W_1^j, but again, as W_2^j is allowed to simulate W_1^j, it can obtain x_1 itself.

Proof sketch. The proof of Theorem 5 follows the main construction from [44, 23] in the context of learning. The main technical complication is caused by the fact that the witnessing circuits are allowed to fail on a significant fraction of inputs.[6]

Intuitively, the proof can be described as follows. Given an n^{dk}-size circuit C with n^d inputs, which we want to learn, we define a set of circuits $D^w(x) := NW_C(w)_x$, for $w \in \{0,1\}^{n^{2d}}$, using a suitable Nisan-Wigderson generator NW_C based on the circuit C. The size of C and the parameters of the NW-generator will ensure that D has n inputs and size n^{10dk}. By the assumption of the theorem, the witnessing circuits W_t^j can find errors of such circuits D^w, for many w's.

We will use the witnessing circuits to reconstruct the circuit C. To this end, we show that there is a frequent 'trace' Tr of the candidate errors generated by circuits W_t^j in the witnessing protocol applied on D^w. In more detail, for a $1/2^{O(n)}$ fraction of w's, the candidate errors generated by $W_1^j, \dots, W_t^j$, with $t \leq \log n$, proceed along the fixed trace $Tr = \langle x_1, \dots, x_t \rangle$ and in roughly 2/3 of these cases the protocol stops at x_t and succeeds in finding an error of D^w satisfying $D^w(x_t) \neq H(x_t)$.

Using Tr and the truth-table of H as a non-uniform advice, we are able to construct a single $2^{O(n)}$-size circuit D' simulating the interactive protocol given by circuits W_t^j, for a $1/2^{O(n)}$ fraction of w's. The simulation of the protocol by circuit D' is obtained after making a small number of queries to C. The queries (together with H) serve as the corrections provided in the interactive witnessing. To guarantee that the set of queries is small (and independent of the choice of w), we exploit the design of the NW-generator.

[6] A similar form of witnessing with some failure probability appears in [46], but in [46] the errors can be removed after giving the witnessing algorithm a non-uniform advice. Krajíček [25] generalizes the construction from [23] by allowing a small fraction of errors, but the fraction is significantly smaller than what is needed in our application. Roughly, [25] allows $\sim p2^{m^\epsilon}$ errors, for some $p, \epsilon < 1$ and $m >> 1$, while we allow $> p2^m$ errors.

Once D' can simulate the witnessing protocol for a $1/2^{O(n)}$ fraction of w's and in roughly 2/3 of these cases find an error x such that $D^w(x) \neq H(x)$, it can determine the value of $NW_C(w)_x$ on a $1/2^{O(n)}$ fraction of w's with advantage roughly 2/3 and approximate C with advantage $1/2^{O(n)}$.

The final learning algorithm is described as the meta-algorithm generating circuits D' but it uses a uniformly random trace Tr' instead of Tr (Tr depends on C and it is unclear how to find it efficiently). Since there are $2^{O(n \log n)}$ possible traces, $Tr' = Tr$ with probability $\geq 1/2^{O(n \log n)}$.

Proof of Theorem 5. In order to derive the conclusion of the theorem it suffices to assume that the witnessing circuits work for distributions $\mathcal{R}$ induced by specific Nisan-Wigderson generators.

Consider a Nisan-Wigderson generator based on a circuit C which we aim to learn. Specifically, for $d \geq 2$ and $n^{2d} \leq m \leq 2n^{2d}$, let $A = \{a_{i,j}\}^{i\in[2^n]}_{j\in[m]}$ be a $2^n \times m$ 0-1 matrix with n^d ones per row and $J_i(A) := \{j \in [m]; a_{i,j} = 1\}$. Then define an NW-generator $NW_C : \{0,1\}^m \mapsto \{0,1\}^{2^n}$ as

$$(NW_C(w))_i = C(w|J_i(A))$$

where $w|J_i(A)$ consists of the w_j's such that $j \in J_i(A)$.

For any $d \geq 2$, Nisan and Wigderson [38] constructed a $2^n \times m$ 0-1 matrix A with n^d ones per row and $n^{2d} \leq m \leq 2n^{2d}$ which is also an (n, n^d)-design meaning that for each $i \neq j$, $|J_i(A) \cap J_j(A)| \leq n$ and $|J_i(A)| = n^d$. Moreover, there are n^{9d}-size circuits which given $i \in \{0,1\}^n$ and $w \in \{0,1\}^m$ output $w|J_i(A)$, cf. [7]. Therefore, if C has n^d inputs and size n^{dk}, then for each $w \in \{0,1\}^m$, $(NW_C(w))_x$ is a function on n inputs x computable by circuits of size n^{10dk}. We want to learn C by a circuit of size $2^{O(n)}$.

Let $\mathcal{R}$ be the distribution on n^{10dk}-size circuits defined so that a random circuit over $\mathcal{R}$ is $(NW_C(w))_x$ for $w \in \{0,1\}^m$ chosen uniformly at random.

By the assumption of the theorem, we have 2^{Kn}-size circuits $\{W^j_t\}_{j\in[2^{Kn}],t\in[\lfloor \log n \rfloor]}$ such that for some $j \in [2^{Kn}]$, for $1 - 3/n^3$ fraction of all $w \in \{0,1\}^m$, circuits $W^j_1, \ldots, W^j_{\lfloor \log n \rfloor}$ find an error of the n^{10dk}-size circuit $(NW_C(w))_x$ attempting to compute H. We will use them in order to break, in a certain sense, the generator NW_C and reconstruct the circuit C.

For each w, define a trace $tr(C, w) = x_1, \ldots, x_t$ as the sequence of $t \leq \log n$ strings generated by $W^j_1, \ldots, W^j_t$ on $(NW_C(w))_x$ such that W^j_t is the first circuit which succeeds in witnessing the error, i.e. $H(x_t) \neq (NW_C(w))_{x_t}$. If circuits $W^j_1, \ldots, W^j_{\lfloor \log n \rfloor}$ do not find an error, $x_t = x_{\lfloor \log n \rfloor}$. The trace is defined w.r.t. a fixed 'helpful' oracle Y providing corrections in the form of bits $(NW_C(w))_x, H(x)$.

For $u \in \{0,1\}^{n^d}$ and $v \in \{0,1\}^{m-n^d}$ define $r_x(u, v) \in \{0,1\}^m$ by putting bits of u into positions $J_x(A)$ and filling the remaining bits by v (in the natural order). We say that

$w \in \{0,1\}^m$ is *good* if the trace $tr(C,w)$ ends with a string witnessing an error of the circuit $(NW_C(w))_x$ where $(NW_C(w))_x \neq H(x)$ and *bad* otherwise. Similarly, we say that $u \in \{0,1\}^{n^d}$ is good w.r.t. $v \in \{0,1\}^{m-n^d}$ and $x' \in \{0,1\}^n$, if $r_{x'}(u,v)$ is good.

The core claim of the proof is the existence of a frequent trace on which the circuits $W_1^j, \dots, W_{\lfloor \log n \rfloor}^j$ succeed in witnessing the error with a significant advantage.

Claim 6. *There is a trace $Tr = X_1, \dots, X_t, t \leq \log n$ such that for $s \geq 1/(6^{2n(t-1)}2^{2n}n)$ fraction of all $a \in \{0,1\}^{m-n^d}$, there is $s' \geq s$ fraction of all $u \in \{0,1\}^{n^d}$, such that $tr(C, r_{X_t}(u,a))$ starts with Tr and at least $(2/3 - 6^t/n^3 - 2/n)s'2^{n^d}$ of these u's are good w.r.t. a, X_t and satisfy $tr(C, r_{X_t}(u,a)) = Tr$.*

The trace Tr is constructed inductively: in step i we want to find $X_1, \dots, X_{i-1}$ such that for $\geq 1/6^{2n(i-1)}$ fraction of all w's $tr(C,w)$ strictly extends $X_1, \dots, X_{i-1}$ and of the w's such that $tr(C,w)$ strictly extends $X_1, \dots, X_{i-1}$ at least $1 - 6^i/2n^3$ fraction are good. For $i = 1$ this holds by the assumption. Assume we have such $X_1, \dots, X_{i-1}$. We want to extend them to $X_1, \dots, X_i$.

Let S be the set of w's such that $tr(C,w)$ strictly extends $X_1, \dots, X_{i-1}$. Since there are at most 2^n strings X_i, there is X_i such that for $s'' \geq 1/(2^{2n}6^{2n(i-1)})$ fraction of w's $tr(C,w)$ starts with $X_1, \dots, X_i$ and $\leq 6^i/n^3$ fraction of these w's are bad. Otherwise, the fraction of good w's in S would be $\leq 1/2^n + 1 - 6^i/n^3 < 1 - 6^i/2n^3$ if $2n^3 \leq 2^n$. Here, the term $1 - 6^i/n^3$ is contributed by the fraction of good w's in S for which $tr(C,w)$ extends $X_1, \dots, X_{i-1}$ by 'frequent' X_i's, where X_i is frequent if $tr(C,w)$ extends $X_1, \dots, X_{i-1}$ to X_i, for $\geq 1/2^{2n}$ fraction of w's in S. The term $1/2^n$ is contributed by the fraction of good w's in S for which $tr(C,w)$ extends $X_1, \dots, X_{i-1}$ by the remaining X_i's.

Now, either for $\geq (2/3)s''$ fraction of w's $tr(C,w)$ stops at X_i (hence, for $\leq (1/3)s''$ fraction of w's the trace continues and $\leq 6^i s''/n^3$ fraction of w's such that $tr(C,w)$ starts with $X_1, \dots, X_i$ are bad) or for $\geq (1/3)s'' \geq 1/6^{2ni}$ fraction of w's the trace strictly extends $X_1, \dots, X_i$. In the latter case, $\leq 6^i s''/n^3$ fraction of w's such that $tr(C,w)$ starts with $X_1, \dots, X_i$ are bad, which means that the fraction of bad w's of w's such that $tr(C,w)$ strictly extends $X_1, \dots, X_i$ is $\leq 3 \cdot 6^i/n^3$.

Since for all w, the length of $tr(C,w)$ is bounded by $\log n$, the process of extending $X_1, \dots, X_{i-1}$ has to stop at some step $1 \leq i \leq \log n$. That is, there is $Tr = X_1, \dots, X_t, t \leq \log n$ such that for $\geq (2/3)s$ fraction of w's $tr(C,w) = Tr$, for $\leq (1/3)s$ fraction of w's $tr(C,w)$ strictly extends Tr and $\leq 6^t s/n^3$ fraction of w's such that $tr(C,w)$ is consistent with Tr are bad, where $s \geq 1/(6^{2n(t-1)}2^{2n})$. The number of good w's such that $tr(C,w) = Tr$ is at least $(2/3 - 6^t/n^3)s2^m$. Therefore, $\geq s/n$ fraction of a's can be completed by $s' \geq s/n$ fraction of u's to a string $w = r_{X_t}(u,a)$ such that $tr(C,w)$ starts with Tr and at least $(2/3 - 6^t/n^3 - 2/n)s'2^{n^d}$ of these u's are good w.r.t. a, X_t and satisfy $tr(C, r_{X_t}(u,a)) = Tr$. This proves the claim.

In order to design a circuit approximating C we will make a small number of queries to C. The queries will form a part of the sets of corrections described as follows. For

$X \in \{0,1\}^n$ and $a' \in \{0,1\}^{m-n^d}$, define $r_X(\cdot, a') \in \{0,1,*\}^m$ by placing the bits of a' in the positions $[m]\backslash J_X(A)$ (in the natural order) and placing $*$'s in the remaining positions. Since A is an (n, n^d)-design, for any row $x \neq X$, there are at most n $*$'s in $r_X(\cdot, a')|J_x(A)$. For $x \neq X$, let $Y^{X,a'}_{x,C}$ be the set of all corrections provided by Y on x, C and $r_X(u, a')|J_x(A)$, for all $u \in \{0,1\}^{n^d}$. Formally, the elements of $Y^{X,a'}_{x,C}$ are pairs $\langle C(r_X(u,a')|J_x(A)), H(x)\rangle$, for $u \in \{0,1\}^{n^d}$. As there are at most n $*$'s in $r_X(\cdot, a')|J_x(A)$, the size of each set $Y^{X,a'}_{x,C}$ is $2^{O(n)}$.

We are ready to describe a circuit D' that approximates C. First, choose uniformly at random $a' \in \{0,1\}^{m-n^d}$, a trace $X^1, \ldots, X^t$ with $t \leq \log n$, a bit $maj \in \{0,1\}$ and $j' \in [2^{Kn}]$. Query C so that all values of C from sets $Y^{X^t,a'}_{x,C}$, for $x \neq X^t$, are obtained. In order to get access to all corrections from $Y^{X^t,a'}_{X^1,C}, \ldots, Y^{X^t,a'}_{X^{t-1},C}$ we provide also the full truth-table of H as a nonuniform advice of D'. The truth table of H is a single nonuniform advice of the learner which works for every C. Then D' computes as follows. For each $u \in \{0,1\}^{n^d}$ produce $r_{X^t}(u, a')$. Next, use $W^{j'}_1$ to produce x^1. If a query of $W^{j'}_1$ cannot be answered by $Y^{X^t,a'}_{x,C}$ with $x \neq X^t$ or if $x^1 \neq X^1$, output maj. (That is, if $W^{j'}_1$ does not proceed along the trace $X^1, \ldots, X^t$, D' stops and outputs maj.) Otherwise, use the advice from $Y^{X^t,a'}_{X^1,C}$ to find out if $H(X^1) = NW_C(r_{X^t}(u,a'))_{X^1}$. If the equality does not hold, output maj. Otherwise, use $W^{j'}_2$ to generate x^2 and continue in the same manner until $W^{j'}_t$ produces x^t. If a query of $W^{j'}_t$ cannot be answered by $Y^{X^t,a'}_{x,C}$ with $x \neq X^t$ or if $x^t \neq X^t$, output maj. Otherwise, output 0 iff $H(X^t) = 1$. The resulting circuit D' has n^d inputs and size $2^{O(n)}$, if $m \leq 2^n$ (which holds w.l.o.g.).

By Claim 6, with probability at least $1/(6^{2n \log n} 2^{O(n \log n)})$ the learner guessed $j' = j$, trace Tr and assignment a such that for at least $(2/3 - 6^t/n^3 - 2/n)s'$ fraction of all $u \in \{0,1\}^{n^d}$, D' will successfully predict $C(u)$. Moreover, for at most $(1/3 + 6^t/n^3 + 2/n)s'$ fraction of all u's, the trace extends Tr or starts with Tr but does not end with a string witnessing an error. Since with probability $1/2$ the correct value on at least half of all remaining u's is maj, $\Pr_u[D'(u) = C(u)] \geq 1/2 + (1/6 - 6^t/n^3 - 2/n)s$. □

The assumption from Theorem 5 is justified by the following lemma which establishes the converse (modulo a small change of parameters).

Lemma 7 (Witnessing from learning). *Let $k \geq 1$; $\epsilon < 1$; $2^n/2n \geq 2^{\epsilon n} \geq n^k$ and H be a Boolean function with n inputs hard to $(1-1/n)$-approximate by circuits of size $2^{\epsilon n}$. Assume* Circuit$[n^k]$ *can be learned by* Circuit$[2^{\epsilon n}]$ *over the uniform distribution with confidence* 1 *up to error ϵ'.*

Then, there are $2^{O(n)}$-size circuits $\{W^j\}_{j \in [2^n/2n]}$, such that for each distribution $\mathcal{R}$ on n^k-size circuits with n inputs there exists $j \in [2^n/2n]$ such that given an oracle access to a random n^k-size circuit $D(x)$ with n inputs, with probability at least $1 - 2\epsilon' n$ over $\mathcal{R}$, after $\leq 2^{\epsilon n}$ queries to circuit D, W^j outputs a not-yet-queried $x \in \{0,1\}^n$ s.t. $D(x) \neq H(x)$.

Proof. By the assumption, there exists an $2^{\epsilon n}$-size circuit W which for each n^k-size circuit D, given an oracle access to D, outputs a circuit C $(1-\epsilon')$-approximating D. Since H is hard to $(1-1/n)$-approximate by circuits of size $2^{\epsilon n} \leq 2^n/2n$, there are at least $2^n/2n$ inputs which have not been queried by W and on which C fails to compute H. Therefore, a random input which has not been queried by W and on which C fails to compute H witnesses $D(x) \neq H(x)$ with probability $\geq 1 - 2\epsilon' n$. Let W^j, for $j \in [2^n/2n]$, be circuits such that W^j simulates W and outputs the j-th input on which C fails to compute H ignoring inputs which have been queried by W. The size of each W^j is $2^{O(n)}$ because it uses the whole truth table of H as a nonuniform advice. Let $\mathcal{R}$ be arbitrary distribution on circuits of size n^k. Since for each D, at least $1 - 2\epsilon' n$ of W^j's succeed, there is W^j which succeeds on random D with probability $\geq 1 - 2\epsilon' n$ over $\mathcal{R}$. □

Note that Theorem 5 together with Lemma 7 imply that for suitable H it is possible to collapse the number of rounds in the interactive witnessing from Theorem 5 at the expense of witnessing errors of slightly smaller circuits (and a small increase in the running time of the witnessing).

Learning from witnessing lower bounds with white-box access. Theorem 5 holds also under the stronger assumption that circuits W_t^j witness errors of n^{10dk}-size nondeterministic circuits D with n inputs (and $\leq n^{10dk}$ nondeterministic bits), where D computes a function in $\mathsf{Circuit}[n^{10dk}]$, i.e. D is a nondeterministic circuit computing a function in $\mathsf{P/poly}$. Then it makes sense to allow W_t^j to access a full description of a given nondeterministic circuit D. The conclusion of the resulting theorem remains valid with the only difference that the learning algorithm is given full description of an n^{dk}-size nondeterministic circuit with n^d inputs representing the target function (which is computable by an n^{dk}-size deterministic circuit with n^d inputs).

Comparison to witnessing in bounded arithmetic. The existence of witnessing analogous to the one from Theorem 5 follows from the provability of circuit lower bounds in bounded arithmetic.

If $H : \{0,1\}^n \rightarrow \{0,1\}$ is an NP function and n_0, k are constants, we can write down a $\forall\Sigma_2^b$ formula $\mathsf{LB}(H, n^k)$ stating that H is hard for circuits of size n^k:

$$\forall n,\ n > n_0\ \forall \text{ circuit } D \text{ of size } \leq n^k\ \exists y,\ |y| = n,\ D(y) \neq H(y),$$

where $D(y) \neq H(y)$ is a Σ_2^b formula stating that a circuit D on input y outputs the opposite value of $H(y)$. Here, Σ_2^b is a class of formulas in the language of Cook's theory PV_1 which define precisely the predicates from Σ_2^p level of the polynomial hierarchy, cf. [27].

By a theorem of Krajíček, Pudlák and Takeuti (KPT theorem) [28], if PV_1 proves $\mathsf{LB}(H, n^k)$ then there are finitely many $poly(n)$-time functions $W_1, \ldots, W_l$ which witness the existential quantifiers of $\mathsf{LB}(H, n^k)$ (including the existential quantifier from the subformula $D(y) \neq H(y)$) in the same interactive way as in Theorem 5 except that the corrections include strings standing for the innermost universal quantifier of $\mathsf{LB}(H, n^k)$ (which allow to verify in p-time that $D(y) \neq H(y)$ has not been witnessed by the most recent candidates). Moreover, $W_1, \dots, W_l$ have access to the full description of a given circuit D and do not make queries to D but directly generate potential errors, cf. [44].

It is possible to change the formula $\mathsf{LB}(H, n^k)$ by introducing a parameter m satisfying $2^n = |m|$ (this is often denoted by writing that $n \in \mathit{LogLog}$) so that the witnessing from the PV_1-provability of the new formula is given by circuits $W_1, \dots, W_l$ of size $2^{O(n)}$. In such case, H is allowed to be in NE. We could allow H to be even an arbitrary Boolean function if we formulated the lower bound in QBF proof systems instead of bounded arithmetic.

A crucial difference between the black-box witnessing from Theorem 5 and white-box witnessing in bounded arithmetic is that, under standard hardness assumptions, the white-box witnessing of p-size circuit lower bounds for functions H such as SAT exists, cf. [36].

Comparison to other witnessing theorems. Lipton and Young [33] showed that for each Boolean function H hard for circuits of size $O(n^{k+1})$ there is a multiset of inputs A of size $O(n^k)$, the so called anticheckers, such that each n^k-size circuit fails to compute H on $\geq 1/3$ fraction of inputs from A. Therefore, for each distribution $\mathcal{R}$ on n^k-size circuits, some input from the set of anticheckers will witness an error of a random n^k-size circuits D (without a single query to D) with probability $\geq 1/3$ over $\mathcal{R}$. Using t rounds the probability of witnessing an error can be increased to $1 - (2/3)^t$. This can be done with $\leq n^{O(kt)}$ witnessing circuits W^i_j. More precisely, we can let $W^i_1, \dots, W^i_t$ to be the i-th possible t-tuple of inputs from the set of anticheckers, for $i < n^{O(kt)}$. Theorem 5 shows that it is not possible to increase this probability further to $1 - 3/n^3$ using $\log n$ rounds unless p-size circuits can be learned efficiently.

Gutfreund, Shaltiel and Ta-Shma [15] showed that if $\mathsf{P} \neq \mathsf{NP}$ there is a p-time algorithm which, given a description of an n^k-time machine D, generates a set of ≤ 3 formulas such that D fails to solve SAT on one of them. Atserias [3] extended this by showing that if $\mathsf{NP} \not\subseteq \mathsf{BPP}$ there is a probabilistic p-time algorithm which, given an oracle access to an n^k-time machine D, outputs with probability $\geq 1/8$ a set of formulas such that D fails to solve SAT on one of them. These algorithms differ from the witnessing in Theorem 5 in several ways: they find errors of uniform algorithms, are allowed to generate errors of different lengths, generate errors with a significantly smaller probability than the probability required in Theorem 5 and the set of formulas generated by the algorithm of Atserias includes formulas on which the algorithm queried D.

4 Learning from breaking pseudorandom generators

Circuit lower bounds can be used to construct PAC learning algorithms also if we assume that they break pseudorandom generators. The construction goes back to a relation between predictability and pseudorandomness which can be interpreted in terms of learning algorithms, as shown by Blum, Furst, Kearn and Lipton [5] and later extended by several other works. In this section we survey some of these connections, derive a construction of learning algorithms from the non-existence of succinct nonuniform pseudorandom function families and show how these connections relate to a question of Rudich about turning demibits to superbits.

Our goal in Section 4 is to approach a construction of learning algorithms for p-size circuits from (constructive circuit lower bounds, where the notion of constructivity of lower bounds is relaxed from the natural properties useful against p-size circuits to) the mere non-existence of cryptographic pseudorandom generators. Since efficient learning algorithms for p-size circuits yield natural properties useful against p-size circuits, which by [50] break pseudorandom generators, this would establish an important dichotomy: cryptographic pseudorandom generators do not exist if and only if there are efficient learning algorithms for small circuits (with suitable parameters). This possibility has been explored by Oliveira-Santhanam [41] and Santhanam [53], cf. Section 4.3.

Question 8 (Dichotomy). *Assume that for each $\epsilon < 1$ there is no pseudorandom generator $g : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ computable in $\mathsf{P/poly}$ and safe against circuits of size 2^{n^ϵ} for infinitely many n. Does it follow that p-size circuits are learnable by circuits of size $2^{O(n^\delta)}$, for some $\delta < 1$, with confidence $1/n$, up to error $1/2 - 1/2^{O(n^\delta)}$?* [7]

4.1 Learning from breaking a dependent generator

We start by recalling the construction from [5], which underlies all results in Section 4.

For an n^c-size circuit C with n inputs define a generator

$$G_C : \{0,1\}^{mn} \mapsto \{0,1\}^{mn+m}$$

which maps m n-bit strings $x_1, \dots, x_m$ to $x_1, C(x_1), \dots, x_m, C(x_m)$.

In order to learn the circuit C it suffices to break the generator G_C depending on C.

Lemma 9 (from [5]). *There is a randomized p-time function L such that for every n^c-size circuit C, if an s-size circuit D satisfies*

$$\Pr[D(x) = 1] - \Pr[D(G_C(x)) = 1] \geq 1/s,$$

[7] In a subsequent paper [47], the question has been connected to the problem of reducing circuit lower bounds to proof complexity lower bounds for concrete propositional proof systems.

then the circuit C is learnable by $L(D)$ over the uniform distribution with random examples, confidence $1/2m^2s$, up to error $1/2 - 1/2ms$.

Proof. Given D, $L(D)$ chooses a random $i \in [m]$, random bits $r_i, \dots, r_m$, random n-bit strings $x_1, \dots, x_m$ except x_i and queries the bits $C(x_1), \dots, C(x_{i-1})$. For $x_i \in \{0,1\}^n$, let $p_i := D(x_1, C(x_1), \dots, x_{i-1}, C(x_{i-1}), x_i, r_i, \dots, x_m, r_m)$. Then $L(D)$ on x_i predicts the value $C(x_i)$ by outputting $\neg r_i$ if $p_i = 1$ and r_i otherwise. By the triangle inequality, a random i satisfies

$$\Pr[p_i = 1] - \Pr[p_{i+1} = 1] \geq 1/ms \tag{4.1}$$

with probability $1/m$. Since the probability over $r_i \dots, r_m, x_1, \dots, x_m$ that $L(D)$ predicts $C(x_i)$ correctly is

$$\frac{1}{2}\Pr[p_i = 1 \mid r_i \neq C(x_i)] + \frac{1}{2}(1 - \Pr[p_i = 1 \mid r_i = C(x_i)]),$$

and $\Pr[p_i = 1] = \frac{1}{2}\Pr[p_i = 1 \mid r_i = C(x_i)] + \frac{1}{2}\Pr[p_i = 1 \mid r_i \neq C(x_i)]$, it follows that the probability over $r_i, \dots, r_m, x_1, \dots, x_m$ that $L(D) = C(x_i)$ is $\geq 1/2 + 1/ms$, for i satisfying (4.1). By averaging, for at least $1/2ms$ fraction of tuples $r_i, \dots, r_m, x_1, \dots, x_{i-1}, x_{i+1}, \dots x_m$, $\Pr_{x_i}[L(D)(x_i) = C(x_i)] \geq 1/2 + 1/2ms$, if i satisfies (4.1). Therefore,

$$\Pr_{x_i}[L(D)(x_i) = C(x_i)] \geq 1/2 + 1/2ms$$

with probability $1/2m^2s$ over the internal randomness of $L(D)$. □

The proof of Lemma 9 implies that learning on average follows from breaking (independent) pseudorandom generators. Specifically, let R be a p-size circuit which given r bits outputs an n^c-size circuit C and consider a generator $G : \{0,1\}^{mn+r} \mapsto \{0,1\}^{mn+m}$ which applies R on its first r input bits in order to output a circuit C and then computes as a generator G_C on the remaining mn inputs. Breaking G implies that we can break G_C with significant probability over C drawn from the distribution induced by R. Consequently, breaking G means that we can learn a big fraction of n^c-size circuits w.r.t. R. Can we improve this average-case learning into a worst-case learning which works for all n^c-size circuits?[8]

[8] In a subsequent work, Hirahara and Nanashima [18] generalize this observation by constructing learning algorithms from the non-existence of one-way functions, where the learning algorithms learn samplers (algorithms generating examples of concepts) having logarithmically small 'computational depth'. It seems reasonable to expect that many (if not all) samplers occurring in practice have logarithmically small computational depth. Ideally, we would like to avoid this assumption. However, more importantly, we would like to have a generator g which does not depend on the sampler/concept we want to learn and such that a distinguisher for g on a (practically feasible) input length n can be efficiently turned into a learning algorithm for practical samplers/concepts on an input length $n^{\Omega(1)} \leq m \leq poly(n)$. This is not achieved by [18] (despite the fact that the generator g obtained in [18] does not depend on the sampler/concept).

4.2 Learning from natural proofs

The proof of Lemma 9 shows also that we can construct a worst-case learning algorithm assuming that given an oracle access to a pseudorandom generator we can efficiently produce its distinguisher. In particular, a single method breaking all pseudorandom generators would suffice.

Definition 10. *The* general circuit size problem $\mathsf{GCSP}[s,k]$ *is the problem to decide whether for a given list of k samples (y_i,b_i), $y_i\in\{0,1\}^n, b_i\in\{0,1\}$, there exists a circuit C of size s computing the partial function defined by samples (y_i,b_i), i.e. $C(y_i)=b_i$ for the given k samples (y_i,b_i). The* parameterized minimum circuit size problem $\mathsf{MCSP}[s]$ *stands for* $\mathsf{GCSP}[s,2^n]$ *where the list of 2^n samples defines the whole truth-table of a Boolean function.*

If we were extraordinary in proving circuit lower bounds, we could solve GCSP efficiently. Note that $\mathsf{MCSP}[n^{O(1)}]\in\mathsf{P/poly}$ is a stronger assumption than the existence of a $\mathsf{P/poly}$-natural property useful against $\mathsf{P/poly}$, which breaks pseudorandom generators.

The following theorem appeared (in a different terminology) in Vadhan [55], see also [19].

Theorem 11 (Learning from succinct natural proofs)**.** *Assume* $\mathsf{GCSP}[n^c,n^d]\in\mathsf{P/poly}$ *for constants $d>c+1$. Then,* $\mathsf{Circuit}[n^c]$ *is learnable by* $\mathsf{P/poly}$ *over the uniform distribution with random examples, confidence $1/poly(n)$, up to error $1/2-1/poly(n)$.*

Proof. As the number of partial Boolean functions on a given set of m inputs is 2^m and the number of n^c-size circuits is bounded by $2^{n^{c+1}}$, $\mathsf{GCSP}[n^c,n^d]\in\mathsf{P/poly}$ implies that for $m=n^d$ there are p-size circuits D such that for each n^c-size circuit C,

$$\Pr[D(x)=1]-\Pr[D(G_C(x))=1]\geq 1/2.$$

Now, it suffices to apply Lemma 9. □

In Theorem 11, we can learn $f\in\mathsf{Circuit}[n^c]$ even if the algorithm for GCSP works just for a significant fraction of partial truth-tables $(y_1,b_1),\dots,(y_{n^d},b_{n^d})$ with zero-error on easy partial truth-tables. Carmosino, Impagliazzo, Kabanets and Kolokolova [7] proved that the assumption of Theorem 11 can be weakened to the existence of a standard natural property. The price for this is that the resulting learning uses membership queries instead of random examples. The crucial idea is similar to the proof of Theorem 5: apply the natural property (as an algorithm for suitable GCSP) on a Nisan-Wigderson generator NW_f based on the function f, which we want to learn.

Theorem 12 (Learning from natural proofs [7])**.** *Let R be a* $\mathsf{P/poly}$*-natural property useful against* $\mathsf{Circuit}[n^d]$ *for some $d\geq 1$. Then, for each $\gamma\in(0,1)$,* $\mathsf{Circuit}[n^k]$ *is learnable by* $\mathsf{Circuit}[2^{O(n^\gamma)}]$ *over the uniform distribution with non-adaptive membership queries, confidence 1, up to error $\frac{1}{n^k}$, where $k=\frac{d\gamma}{a}$ and a is an absolute constant.*

4.3 Learning from breaking pseudorandom function families

Oliveira and Santhanam [41] showed that the assumption of the existence of natural proofs from Theorem 12 can be further weakened to the existence of a distinguisher breaking non-uniform pseudorandom function families. Their result follows from a combination of Theorem 12 and the Min-Max Theorem. Using their strategy but combining the Min-Max Theorem with Theorem 11, we will show now that learning *with random examples* can be obtained from distinguishers breaking succinct non-uniform pseudorandom function families.[9]

A *two-player zero-sum game* is specified by an $r \times c$ matrix M and is played as follows. MIN, the row player, chooses a probability distribution p over the rows. MAX, the column player, chooses a probability distribution q over the columns. A row i and a column j are drawn randomly from p and q, and MIN pays $M_{i,j}$ to MAX. MIN plays to minimize the expected payment, MAX plays to maximize it. The rows and columns are called the *pure strategies* available to MIN and MAX, respectively, while the possible choices of p and q are called *mixed strategies*. The Min-Max theorem states that playing first and revealing one's mixed strategy is not a disadvantage:

$$min_p max_j \sum_i p(i)M_{i,j} = max_q min_i \sum_j q(j)M_{i,j}.$$

Note that the second player need not play a mixed strategy - once the first player's strategy is fixed, the expected payoff is optimized for the second player by playing some pure strategy. The expected payoff when both players play optimally is called the *value* of the game. We denote it $v(M)$.

A mixed strategy is *k-uniform* if it chooses uniformly from a multiset of k pure strategies. Let $M_{min} = min_{i,j} M_{i,j}$ and $M_{max} = max_{i,j} M_{i,j}$. Newman [37], Althöfer [1] and Lipton-Young [33] showed that each player has a near-optimal k-uniform strategy for k proportional to the logarithm of the number of pure strategies available to the opponent.

Theorem 13 ([37, 1, 33]). *For each $\epsilon > 0$ and $k \geq \ln(c)/2\epsilon^2$,*

$$min_{p \in P_k} max_j \sum_i p(i)M_{i,j} \leq v(M) + \epsilon(M_{max} - M_{min}),$$

where P_k denotes the k-uniform strategies for MIN. The symmetric result holds for MAX.

[9] There have been some subsequent developments after the composition of the present papers. Goldberg and Kabanets [14] showed that learning of p-size circuits with random examples follows from efficient algorithms solving various versions of MCSP adapted to time-bounded Kolmogorov complexity. Karchmer [20] proved that a stronger form of an average-case learning with random examples follows from natural proofs obtained in a suitable way.

Definition 14 (Succinct non-uniform PRF)**.** *An (m, m')-succinct non-uniform pseudo-random function family from circuit class $\mathcal{C}$ safe against circuits of size s is a set S of partial truth-tables $\langle(x_1, b_1), \dots, (x_m, b_m)\rangle$ where each x_i is an n-bit string and $b_i \in \{0,1\}$ such that each partial truth-table from S is computable by one of m' circuits from $\mathcal{C}$ and for every circuit D of size s,*

$$\Pr_x[D(x) = 1] - \Pr_{x \in S}[D(x) = 1] < 1/s$$

where the first probability is taken over $x \in \{0,1\}^{m(n+1)}$ chosen uniformly at random and the second probability over partial truth-tables chosen uniformly at random from S.

Theorem 15 (Learning or succinct non-uniform PRF)**.** *Let $c \geq 1$ and $s > n, m \geq 1$. There is an $(m, 8s^4)$-succinct non-uniform PRF in $\mathsf{Circuit}[n^c]$ safe against $\mathsf{Circuit}[s]$ or there are circuits of size $poly(s)$ learning $\mathsf{Circuit}[n^c]$ over the uniform distribution with random examples, confidence $1/poly(s)$, up to error $1/2 - 1/poly(s)$.*

Proof. Consider a two-player zero-sum game specified by a matrix M with rows indexed by n^c-size circuits with n inputs and columns indexed by s-size circuits with $m(n+1)$ inputs. Define the entry $M_{C,D}$ of M corresponding to a row circuit C and a column circuit D as

$$M_{C,D} := |\Pr_x[D(x) = 1] - \Pr_x[D(G_C(x)) = 1]|$$

for the generator G_C from the proof of Lemma 9. Hence $M_{max} - M_{min} \leq 1$.

If $v(M) \geq 1/4s$, then by Theorem 13 (with $\epsilon = 1/8s$), there exist a multiset of $k \leq 32n^{c+1}s^2$ s-size circuits $D^1, \dots, D^k$ such that for every n^c-size circuit C, a random D from $D^1, \dots, D^k$ satisfies

$$\mathrm{E}[|\Pr[D(x) = 1] - \Pr[D(G_C(x)) = 1]|] \geq 1/8s.$$

By Lemma 9, for every n^c-size circuit C, one of the circuits $D^1, \dots, D^k$ (or their negations) can be used to learn C with confidence $1/poly(s)$, up to error $1/2 - 1/poly(s)$. A $poly(s)$-size circuit using a random D^i from $D^1, \dots, D^k$ or its negation thus learns $\mathsf{Circuit}[n^c]$ with random examples, confidence $1/poly(s)$, up to error $1/2 - 1/poly(s)$.

If $v(M) < 1/4s$, then by Theorem 13 (with $\epsilon = 1/4s$), there exists a multiset of $k \leq 8s^4$ n^c-size circuits $C^1, \dots, C^k$ such that for every s-size circuit D, a random C from $C^1, \dots, C^k$ satisfies

$$\mathrm{E}[|\Pr[D(x) = 1] - \Pr[D(G_C(x)) = 1]|] \leq 1/2s.$$

Since $\mathrm{E}[|\Pr[D(x) = 1] - \Pr[D(G_C(x)) = 1]|] \geq |\Pr[D(x) = 1] - \mathrm{E}[\Pr[D(G_C(x)) = 1]]|$ a generator

$$G : \{0,1\}^{mn + \lceil \log k \rceil} \mapsto \{0,1\}^{mn+m}$$

which takes as input a string of length $mn + \lceil \log k \rceil$ encoding (an index of) a circuit C from $C^1, \ldots, C^k$ together with m n-bit strings $x_1, \ldots, x_m$ and outputs $x_1, C(x_1), \ldots, x_m, C(x_m)$ is safe against circuits of size s. The range of G defines an $(m, 8s^4)$-succinct non-uniform PRF in $\mathsf{Circuit}[n^c]$ safe against $\mathsf{Circuit}[s]$. □

Note that the existence of a generator G from the proof of Theorem 15 follows directy from a counting argument if we do not require that G defines a PRF of small complexity: a random set of $poly(s, n)$ strings (yielding a non-uniform pseudorandom generator mapping $\{0,1\}^{O(\log s)}$ to $\{0,1\}^n$) fools circuits of size s.

4.4 Superbits vs demibits

Rudich [52] proposed a conjecture about the existence of superbits, a version of pseudorandom generators safe against nondeterministic circuits, and showed that it rules out the existence of NP-natural properties against $\mathsf{P/poly}$. He then asked whether the existence of superbits follows from a seemingly weaker assumption of the existence of so called demibits. We note that an affirmative answer to his question would resolve Question 8 in nondeterministic setting.

Definition 16 (Superbit)**.** *A function $g : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ computable by p-size circuits is a* superbit *if there is $\epsilon < 1$ such that for infinitely many input lengths n, for all nondeterministic circuits C of size $|C| \leq 2^{n^\epsilon}$,*

$$\Pr_{x \in \{0,1\}^{n+1}}[C(x) = 1] - \Pr_{x \in \{0,1\}^n}[C(g(x)) = 1] < 1/|C|.$$

Definition 17 (Demibit)**.** *A function $g : \{0,1\}^n \mapsto \{0,1\}^{n+1}$ computable by p-size circuits is a* demibit *if there is $\epsilon < 1$ such that for infinitely many input lengths n, no nondeterministic circuit C of size $|C| \leq 2^{n^\epsilon}$ satisfies*

$$\Pr_{x \in \{0,1\}^{n+1}}[C(x) = 1] \geq 1/|C| \quad \textit{and} \quad \Pr_{x \in \{0,1\}^n}[C(g(x)) = 1] = 0.$$

Proposition 18 (Question 8 vs Rudich's problem)**.** *Assume the existence of demibits implies the existence of superbits. Then, either superbits exist or for each $c \geq 1$, for each $\epsilon < 1$, $\mathsf{Circuit}[n^c]$ is learnable by $\mathsf{Circuit}[2^{O(n^\epsilon)}]$ over the uniform distribution with random examples, confidence $1/2^{O(n^\epsilon)}$ up to error $1/2 - 1/2^{O(n^\epsilon)}$, where the learner is allowed to generate a nondeterministic or co-nondeterministic circuit approximating the target function.*

Proof. Assume that superbits do not exist and that their non-existence implies the non-existence of demibits. Consider a generator $G : \{0,1\}^{mn+n^{c+1}} \mapsto \{0,1\}^{mn+m}$, with $m = n^{c+1} + 1$, which interprets the first n^{c+1} bits of its input as a description of an n^c-size circuit C and then computes on the remaining mn inputs as generator G_C from Lemma 9. Since G is not a demibit (otherwise, it would contradict our assumptions), for each $\epsilon < 1$ there are nondeterministic circuits D of size $2^{(mn+m-1)^\epsilon}$, such that for each n^c-size circuit C,

$$\Pr[D(x) = 1] - \Pr[D(G_C(x)) = 1] \geq 1/|D|.$$

By the proof of Lemma 9, this means that n^c-size circuits are learnable by circuits of size $poly(|D|)$ with confidence $1/poly(|D|)$ up to error $1/2 - 1/poly(|D|)$, except that the learner might generate nondeterministic (if $r_i = 0$) or co-nondeterminitic (if $r_i = 1$) circuit approximating the target function. □

5 Learning speedup

A striking consequence of the relation between natural proofs and learning algorithms is a learning speedup of Oliveira and Santhanam [41].

Suppose $\mathsf{P/poly}$ is learnable by circuits of weakly subexpoential size $2^n/n^{\omega(1)}$. The learning circuits can be used to accept truth-tables of all functions in $\mathsf{P/poly}$ while their size guarantees that many hard functions are going to be rejected. This implies the existence of a $\mathsf{P/poly}$-natural property useful against $\mathsf{P/poly}$, which by Theorem 12, gives us circuits of strongly subexponential size 2^{n^γ}, $\gamma < 1$, learning $\mathsf{P/poly}$.

The argument of Oliveira and Santhanam can be generalized to a speedup of learners of arbitrary size s. Here, we show how to derive such a generalized version more directly without constructing natural proofs and invoking Theorem 12. This is possible thanks to a more direct exploitation of a slightly modified NW-generator. A drawback of the approach is that we need to assume learning with random examples instead of membership queries.

Theorem 19 (Generalized speedup). *Let $d, k \geq 1$ and $n \leq s(n) \leq 2^n/n$. Assume $\mathsf{Circuit}[n^{10dk}]$ is learnable by $\mathsf{Circuit}[s(n)]$ over the uniform distribution with random examples, confidence 1, up to error $1/2 - 5/n$. Then circuits of size m^k with $m = n^d$ inputs are learnable by circuits of size $n^{dK}(s(n))^3$ over the uniform distribution with non-adaptive membership queries, confidence $1/n^3$, up to error $1/2 - 1/n$. Here, K is an absolute constant.*

Theorem 19 implies, for example, that if for each $d, k \geq 1$ and each sufficiently big n there is an $n^{10\log n}$-size circuit learning circuits of size n^{10dk} (with n inputs), then for each $d, k \geq 1$ and for infinitely many n there is an $n^K n^{(30\log n)/d^2}$-size circuit learning circuits of size n^k (with n inputs). That is, if p-size circuits are learnable with random examples by circuits of quasipolynomial size $n^{O(\log n)}$, then p-size circuits are learnable with membership queries by circuits of size $O(n^{\epsilon \log n})$, for each $\epsilon > 0$. The speedup is achieved w.r.t. the input length of target functions at the expense of their circuit complexity.

Proof. Let A be a $2^b \times u$ 0-1 matrix forming a (b, n^d)-design with $|J_i(A)| = n^d$ for $n^{2d} \leq u \leq 2n^{2d}$, a constant d and parameter b such that $ns \leq 2^b \leq 2ns$. The design is constructed in the usual way by evaluating polynomials of degree $\leq b$ on n^d points of a field with $n^d \leq p \leq 2n^d$ elements. In particular, there are n^{9d}-size circuits which given $i \in \{0,1\}^b$ and $w \in \{0,1\}^u$ output $w|J_i(A)$. Define NW_f-generator mapping strings w of length u to strings of length 2^n as

$$(NW_f(w))_{x_1,\dots,x_n} = f(w|J_{x_1,\dots,x_b}(A)).$$

Then for each m-input function $f \in \mathsf{Circuit}[m^k]$ and $w \in \{0,1\}^u$, $(NW_f(w))_x$ is computable as a function of $x = x_1, \dots, x_n \in \{0,1\}^n$ by a circuit of size n^{10dk}.

By the assumption of the theorem every such circuit $(NW_f(w))_x$ is learnable by a circuit L of size s with confidence $\delta = 1$, up to error $1/2 - \epsilon$. Consequently, there is a circuit D^f of size $O(s^3)$ such that

$$\Pr_{w,x,y^1,\dots,y^t}[D^f(x_1,\dots,x_n,w,y^1,\dots,y^t) = f(w|J_{x_1,\dots,x_b}(A))] \geq (1/2+\epsilon)\delta \tag{5.1}$$

where D^f queries values $f(w|J_{y^j}(A))$ for $t \leq s$ random strings $y^j \in \{0,1\}^b$, $j = 1, \dots, t$. The size of D^f takes into account the need to simulate the circuit described by L. Now, random $y^1, \dots, y^t$ satisfy

$$\Pr_{w,x}[D^f(x_1,\dots,x_n,w,y^1,\dots,y^t) = f(w|J_{x_1,\dots,x_b}(A))] \geq 1/2+\epsilon-1/n \tag{5.2}$$

with probability at least $1/n$. Otherwise, the probability in (5.1) would be $< 1/n + (1/2 + \epsilon - 1/n)$. Similarly, given $y^1, \dots, y^t$ such that (5.2) holds, a random $x \in \{0,1\}^n$ satisfies

$$\Pr_{w}[D^f(x_1,\dots,x_n,w,y^1,\dots,y^t) = f(w|J_{x_1,\dots,x_b}(A))] \geq 1/2+\epsilon-3/n \tag{5.3}$$

with probability at least $2/n$. Moreover, since every y^j specifies 2^{n-b} values of $(NW_f(w))_x$, given $y^1, \dots, y^t$, a random $x \in \{0,1\}^n$ equals some y^j on the first b bits with probability $\leq t/2^b \leq 1/n$. Applying the same averaging one more time, for $y^1, \dots, y^t$ and x which differs on the first b bits from each y^j and satisfies (5.3), randomly fixed $u - n^d$ bits of w on the positions of $[u] \backslash J_x(A)$ preserve the probability (5.3) up to an additional error $1/n$ with probability at least $1/n$.

For each $y^1, \dots, y^t$, each x which differs on the first b bits from every y^j and for each fixation of $u - n^d$ bits of w on the positions of $[u] \backslash J_x(A)$, (b, n^d)-design guarantees that the number of all queries $f(w|J_{y^j}(A))$, $j = 1, \dots, t$, of D^f for all possible w with the $u - n^d$ fixed bits is $\leq t2^b$. We can thus learn a circuit D' approximating $f \in \mathsf{Circuit}[m^k]$ with $m = n^d$ inputs with advantage $1/2 + \epsilon - 4/n$ in the following way. Choose random $y^1, \dots, y^t$, x, random $u - n^d$ bits of w corresponding to $[u] \backslash J_x(A)$ and query $\leq t2^b$ values $f(w|J_{y^j}(A))$ for all possible w with the $u - n^d$ fixed bits. Then the circuit D', given n^d bits of w corresponding to $J_x(A)$, generates w and computes as D^f with the provided queries $f(w|J_{y^j}(A))$. Since w can be constructed from given n^d bits, x and the $u-n^d$ fixed bits of w by a circuit of size $n^{O(d)}$, each $w|J_{y^j}(A)$ can be constructed from w and y^j by a circuit of size n^{9d} and for each query to f the right value can be selected by a circuit of size $O(n^d t 2^b)$, the size of D' is $O(s^3 + tn^{9d} + n^d t^2 2^b + n^{O(d)}) \leq n^{O(d)} s^3$. D' can be described by $n^{dK} s^3$ bits, for an absolute constant K, and constructed by a circuit of the same size which just substitutes y^j, x and $u - n^d$ bits of w in the otherwise fixed description of D'.

Since random $y^1, \dots, y^t$ satisfy (5.2) with probability at least $1/n$, a random x differs on the first b bits from each $y^1, \dots, y^t$ and satisfies (5.3) with probability at least $1/n$ while the randomly fixed $u - n^d$ bits of w have the desired property with probability at least $1/n$ as well, the confidence of the learning algorithm is at least $1/n^3$. □

We give one more proof of the learning speedup which also addresses the issue of membership queries.

Theorem 20 (Alternative speedup). *Let $d \geq 2; k \geq 1$ and $\epsilon < 1$. Assume $\mathsf{Circuit}[n^{10dk}]$ is learnable by $\mathsf{Circuit}[2^{\epsilon n}]$ over the uniform distribution (possibly with membership queries) with confidence 1, up to error $1/n^5$. Then, circuits of size n^{dk} with n^d inputs are learnable by circuits of size 2^{Kn} over the uniform distribution with confidence $1/2^{Kn}$ up to error $1/2 - 2^{Kn}$, where K is an absolute constant.*

Proof. By a counting argument there exists H which is not $(1 - 1/n)$-approximable by circuits of size $2^{\epsilon n}$. Here, n is w.l.o.g. sufficiently big. By Lemma 7, learnability of $\mathsf{Circuit}[n^{10dk}]$ by $\mathsf{Circuit}[2^{\epsilon n}]$ up to error $1/n^5$ implies the existence of circuits of size $2^{O(n)}$ witnessing errors of circuits of size n^{10dk} with probability $\geq 1 - 2/n^4$. The conclusion thus follows by applying Theorem 5. The improved confidence and approximation parameter is the consequence of the fact that our witnessing circuits succeed in the first round, i.e. $t = 1$. □

Proof-search speedup. The core trick behind Theorem 19 can be formulated in the context of proof complexity. Assume that an n^{10dk}-size lower bound is provable in a proof system P by a proof of size $s(n)$. Then, a substitutional instance of the same P-proof of size $s(n)$ proves an m^k-size lower bound for circuits with $m = n^d$ inputs, on inputs given by the NW-generator from the proof of Theorem 19. Here, the base function of the NW-generator is not specified but represented by free variables encoding a circuit of size m^k.

Nonlocalizable hardness magnification. Theorem 19 and the original speedup of Oliveira and Santhanam can be interpreted as hardness magnification theorems. Hardness magnification is an approach to strong complexity lower bounds by reducing them to seemingly much weaker lower bounds developed in a series of recent papers [42, 36, 40, 34, 11, 12, 9, 8, 10, 35, 13, 32], see [8] for a more comprehensive survey. For example, it turns out that in order to prove that functions computable in nondeterministic quasipolynomial-time are hard for NC^1 it suffices to show that a parameterized version of the minimum circuit size problem MCSP is hard for $\mathsf{AC}^0[2]$. However, [8] identified a *locality barrier* which explains why direct adaptations of many existing lower bounds do not yield strong complexity lower bounds via hardness magnification. Essentially, the reason is that the existing lower bounds for explicit Boolean functions work often even for models which are allowed to use arbitrary oracles with $n^{o(1)}$-small fan-in (local oracles). This is easy to see in the case of $\mathsf{AC}^0[2]$ lower bounds: oracles of small fan-in can be simulated by polynomials of low degree. On the other hand, hardness magnification theorems typically yield (unconditional) upper bounds in the form of weak computational models extended with local oracles computing specific problems such as the abovementioned version of MCSP. In fact, even irrespective of hardness magnification it is important to develop lower bound methods which do not localize (i.e. lower bounds which do not remain valid in the presence of local oracles): proving the nonexistence of subexponential-size learning algorithms for $\mathsf{P/poly}$ would imply the nonexistence of $\mathsf{P/poly}$ natural properties against $\mathsf{P/poly}$ but it is not hard to see that natural properties against $\mathsf{P/poly}$ are computable by a single local oracle applied on a prefix of the input. Overcoming the locality barrier is thus essential for proving strong complexity lower bounds in general.[10]

Theorem 19, if read counterpositively, is a magnification of $O(n^{\epsilon \log n})$-size lower bounds for learning p-size circuits to $n^{O(\log n)}$-size lower bounds. This differs from previous hardness magnification theorems by avoiding localization: the size of the learner plays a crucial role in the reduction and therefore cannot be simply replaced by an arbitrary oracle. The same trick is behind non-blackbox worst-case to average-case reductions within NP of Hirahara [16]. To the best of our knowledge, the only other hardness magnification theorems with this property appeared in [8] and [17].[11] [8, Theorem 1], like Hirahara [16] and the speedup of Oliveira-Santhanam, is based on the result of Carmosino, Impagliazzo, Ka-

[10] Some known circuit lower bounds above the magnification threshold are provably nonlocalizable but they do not fit to the framework of the so called Hardness Magnification frontier [8], one reason being that they do not work for explicit and natural problems, cf. [8, 10]. For example, a nonlocalizable lower bound from [8] works for a function in E which is artificial in the sense that it is designed to avoid localization, not for a problem of independent interest such as MCSP. Oliveira [39] showed that near superlinear-size lower bounds for a version of MCSP defined w.r.t. a notion of randomized Kolmogorov complexity imply strong circuit lower bounds while the same problem is provably hard for probabilistic p-time. The lower bound of Oliveira works, however, only against uniform models of computation. Moreover, the magnification theorem concludes at best a 'non-explicit' lower bound of the form 'quasipolynomial-time QP being hard for $\mathsf{P/poly}$.' Similarly, an approach of Chen, Jin and Williams [10] via derandomizations and uniform obstructions appears to avoid the locality barrier but yields at best lower bounds of the form $\mathsf{QP} \not\subseteq \mathsf{P/poly}$.

[11] There are two more results which could be potentially classified as nonlocalizable hardness magnifications. A theorem of Buresh-Oppenheim and Santhanam [6, Theorem 1] is based on an exploitation of Nisan-Wigderson generators similar to that of [8] but it seems less practical in its current form, as it magnifies only lower bounds for nondeterministic circuits. The other result of Tal [54] shows that an average-case hardness for formulas of size s can be magnified to the worst-case hardness for slightly bigger

banets and Kolokolova [7]. However, the hardness magnification from [8] is still captured by the locality barrier: it asks for a lower bound for a version of MCSP while the localized version of the lower bound does not hold (as witnessed by other hardness magnification theorems). Theorem 19 does not seem to localize in this sense either: it asks for an $n^{\epsilon \log n}$-size lower bound on learning algorithms while there seems to be no reason to expect that p-size circuits are learnable by circuits of size $O(n^{\log n})$ extended with oracles of fan-in $n^{o(1)}$. (Such a localization would mean that p-size circuits are learnable in subexponential size.) The magnification theorems of Hirahara [17] face similar complications.[12]

Unfortunately, Theorem 19 does not reduce p-size lower bounds to, say, subquadratic lower bounds: It magnifies $n^{O(d)}s^3$-size lower bounds for learning functions with $m = n^d$ inputs (and circuit complexity m^k) to an s-size lower bound for learning functions with n inputs (and circuit complexity n^{10dk}). That is, a polynomial speedup w.r.t. the input-length of target functions is traded for a polynomial decrease of the circuit size of target functions. Ideally, we would like to magnify, say, $n^{1.9}$-size formula lower bound for learning circuits of size $n^{1.1}$ with n inputs to $n^{O(1)}$-size formula lower bounds for learning circuits of size $n^{2.1}$ with n inputs. If the existing methods for proving the required formula lower bounds were applicable to prove subquadratic formula lower bounds for learning algorithms (note that such lower bounds are allowed to localize and naturalize), such a strengthening of Theorem 19 would lead to explicit NC^1 lower bounds.

6 Concluding remarks and open problems

The methods for deriving learning algorithms from circuit lower bounds presented in this paper might be improvable in many ways.

Safe cryptography or efficient learning. Perhaps the most appealing question asks for bridging cryptography and learning theory. Showing that efficient learning follows from breaking pseudorandom generators, i.e. answering positively Question 8, would establish a remarkable win-win situation. As discussed in Section 4.4 the question is closely related to a problem of Rudich about turning demibits to superbits.

formulas. A problem is that [54] magnifies at best to an s^2-size lower bound. Moreover, if we wanted to strenghten it further by connecting it with another magnification theorem, it is not clear how to preserve the nonlocalizability - the weak lower bound obtained via [54] would likely localize.

[12] Hirahara [17, Theorem 11 and 13] proves two types of magnification theorems. The first type essentially adapts the result from [8] in the context of weaker computational models. The second type extends it by introducing metacomputational circuit lower bound problems MCLPs and showing that weak lower bounds for MCLPs can be magnified as well. MCLPs are not solvable by any algorithm whatsoever unless standard hardness assumptions break. This implies that there is no unconditional upper bound for MCLPs and the locality barrier does not apply. Unfortunately, we do not have any interesting lower bound for MCLPs either. The corresponding magnification theorems thus do not establish a Hardness Magnification frontier [8]. Nevertheless, as suggested in [17], developing such methods might be a way to strong lower bounds.

Instance-specific learning vs PAC learning. Circuit lower bounds correspond to a simple instance-specifc learning model described in Section 3.1. Can we improve our understanding of the model and its relation to PAC learning? In particular, can we determine how much we can learn from a single circuit lower bound? A possible formalization of the problem is given by Question 4.

Connections to proof complexity. The present paper brings several methods from proof complexity to learning theory. It seems likely that these connections can be strengthened. A particularly relevant part of proof complexity is the theory of proof complexity generators, cf. [24]. An interesting conjecture in the area due to Razborov [49] implies a conditional hardness of circuit lower bounds in strong proof systems. In other words, Razborov's conjecture asks for turning short proofs of circuit lower bounds into upper bounds breaking standard hardness assumptions.

Notably, strengthening Theorem 5 by allowing white-box access in the witnessing of lower bounds would lead to a conditional unprovability of p-size lower bounds for SAT in Cook's theory PV_1. A complication is that under standard hardness assumptions such a witnessing exists. That is, in order to obtain the conditional unprovability, one might need to exploit the PV_1-provability in a deeper way. Nevertheless, this suggests a simplified version of Question 8: Can we prove a disjunction stating the PV_1-consistency of the existence of strong pseudorandom generators or the PV_1-consistency of efficient learning? Since, by witnessing theorems in PV_1, both the PV_1-provability of the non-existence of pseudorandom generators and the PV_1-provability of the impossibility of efficient learning imply uniform efficient algorithms witnessing these facts, it could be possible to combine them with a version of uniform MinMax [56] to get a contradiction.

Nonlocalizable hardness magnification near the existing lower bounds. Can we push forward the program of hardness magnification by strengthening the magnification from Theorem 19 to a setting in which strong circuit lower bounds follow from lower bounds near the already existing ones? The importance of the question stems from the necessity of developing nonlocalizable magnification theorems or nonlocalizable constructive lower bound methods as discussed in Section 5.

SAT solving circuit lower bounds. It would be interesting to investigate practical consequences of the provability of circuit lower bounds. Circuit lower bounds for explicitly given Boolean functions are coNP statements which means that they are encodable into propositional tautologies resp. SAT instances. Could SAT solvers be successful in proving interesting instances of circuit lower bounds for some fixed input lengths? At present the existing SAT solvers are not able to deal with circuit lower bounds for circuits with more than 13 gates, cf. [29], so one would need to develop new methods to get more interesting outcomes. If successful, this could provide an experimental verification of central results and conjectures from complexity theory such as $\mathsf{P} \neq \mathsf{NP}$ up to some finite domain. As discussed in the present paper, efficient algorithms proving circuit lower bounds can be also transformed into learning algorithms, which provides a separate motivation for this line of research.

In particular, SAT solving of circuit lower bounds could lead to an interesting comparison with the research on neural networks. The task of training a neural network is to design a circuit C of size s, typically with a specific architecture, coinciding with some training input samples $(y_i, f(y_i))$, and apply it to predict the value $f(y)$ on a new input y. As discussed in Section 3.1, this problem can be addressed by proving a circuit lower bound. Since proving a circuit lower bound can give us a reliable instance-specific prediction one could try to use SAT solvers to verify outcomes of neural networks. More generally, one could try to simulate neural networks by SAT solving circuit lower bounds. A potential advantage of SAT solvers is that they do not need to construct a circuit coinciding with training data - it is enough to prove its properties (lower bounds). On the other hand, SAT solvers need to prove a universal statement which might turn out to be even harder.

Acknowledgements

I would like to thank Rahul Santhanam for many inspiring discussions which, in particular, motivated me to prove Theorem 5. I am indebted to Susanna de Rezende and Erfan Khaniki for many illuminating discussions during the development of the project. I would also like to thank V. Kanade for helpful comments on the existing learning models and L. Chen, S. Hirahara, V. Kabanets, J. Krajíček, I.C. Oliveira and anonymous reviewers for helpful comments on a draft of the paper and related topics. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodovska-Curie grant agreement No 890220. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] Althöfer I.; *On sparse approximations to randomized strategies and convex combinations*; Linear Algebra and its Applications, 199(1):339-355, 1994.

[2] Amit N., Goldwasser S., Paradise O., Rothblum G.; *Models that prove their own correctness*; preprint (available at ECCC), 2024.

[3] Atserias A.; *Distinguishing SAT from polynomial-size circuits, through black-box queries*; Computational Complexity Conference (CCC), 2006.

[4] Binnendyk E., Carmosino M., Kolokolova A., Ramyaa R.., Sabin M.; *Learning with distributional inverters*; Algorithmic Learning Theory (ALT), 2022.

[5] Blum A., Furst M., Kearns J., Lipton R.; *Cryptographic primitives based on hard learning problems*; International Cryptology Conference (CRYPTO), 1993.

[6] Buresh-Oppenheim J., Santhanam R.; *Making hard problems harder*; Computational Complexity Conference (CCC), 2006.

[7] Carmosino M., Impagliazzo R., Kabanets V., Kolokolova A.; *Learning algorithms from natural proofs*; Computational Complexity Conference (CCC), 2016.

[8] Chen L., Hirahara S., Oliveira I.C., Pich J., Rajgopal N., Santhanam R.; *Beyond natural proofs: hardness magnification and locality*; Innovations in Theoretical Computer Science (ITCS), 2020.

[9] Chen L., Jin C., Williams R.; *Hardness magnification for all sparse* NP *languages*; Foundations of Computer Science (FOCS), 2019.

[10] Chen L., Jin C., Williams R.; *Sharp threshold results for computational complexity*; Symposium on Theory of Computing (STOC), 2020.

[11] Chen L., McKay D., Murray C., Williams R.; *Relations and equivalences between circuit lower bounds and Karp-Lipton theorems*; Computational Complexity Conference (CCC), 2019.

[12] Chen L., Tell R.; *Bootstrapping results for threshold circuits "just beyond" known lower bounds*; Symposium on Theory of Computing (STOC), 2019.

[13] Cheragchi M., Hirahara S., Myrisiotis D., Yoshida Y.; *One-tape Turing machine and read-once branching program lower bounds for MCSP*; preprint, 2020.

[14] Goldberg H., Kabanets V.; *Improved learning from Kolmogorov complexity*; Computational Complexity Conference (CCC), 2023.

[15] Gutfreund D., Shaltiel R., Ta-Shma A.; *If NP languages are hard in the worst-case then it is easy to find their hard instances*; Computational Complexity Conference (CCC), 2005.

[16] Hirahara S.; *Non-black-box worst-case to average-case reductions within* NP; Foundations of Computer Science (FOCS), 2018.

[17] Hirahara S.; *Non-disjoint promise problems from meta-computational view of pseudorandom generator constructions*; Computational Complexity Conference (CCC), 2020.

[18] Hirahara S., Nanashima M.; *Learning in Pessiland via inductive inference*; Foundations of Computer Science (FOCS), 2023.

[19] Ilango R., Loff B., Oliveira I.C.; *NP-hardness of circuit minimization for multi-output functions*; Computational Complexity Conference (CCC), 2020.

[20] Karchmer A.; *Average-case PAC-learning from Nisan's natural proofs*; preprint (available at ECCC), 2023.

[21] Krajíček J.; *Bounded arithmetic, propositional logic, and complexity theory*; Cambridge University Press, 1995.

[22] Krajíček J.; *Dual weak pigeonhole principle, pseudo-surjective functions and provability of circuit lower bounds*; Journal of Symbolic Logic, 69(1):265-286, 2004.

[23] Krajíček J.; *On the proof complexity of the Nisan-Wigderson generator based on a hard* $\mathsf{NP} \cap \mathsf{coNP}$ *function*; Journal of Symbolic Logic, 11(1):11-27, 2011.

[24] Krajíček J.; *Forcing with random variables and proof complexity*; Cambridge University Press, 2011.

[25] Krajíček J.; *Pseudo-finite hard instances for a student-teacher game with a Nisan-Wigderson generator*; Logical Methods in Computer Science, 8(3:9), 2012.

[26] Krajíček J.; *On the computational complexity of finding hard tautologies*; Bulletin of the London Mathematical Society, 46(1):111-125, 2014.

[27] Krajíček J.; *Proof complexity*; Cambridge University Press, 2019.

[28] Krajíček J., Pudlák P., Takeuti G.; *Bounded arithmetic and the polynomial hierarchy*; Annals of Pure and Applied Logic, 52:143-153, 1991.

[29] Kulikov A.S., Slezkin N.; *SAT-based circuit local improvement*; preprint, 2021.

[30] Li L., Littman M., Walsh T.; *Knows what it knows: a framework for self-aware learning*; International Conference on Machine Learning (ICML), 2008.

[31] Linial N., Mansour Y., Nisan N.; *Constant depth circuits, Fourier transform, and learnability*; Journal of the Association for Computing Machinery; 40(3):607-620, 1993.

[32] Liu Y., Pass R.; *Cryptography from sublinear-time average-case hardness of time-bounded Kolmogorov complexity*; Symposium on Theory of Computing (STOC), 2021.

[33] Lipton R.J., Young N.E.; *Simple strategies for large zero-sum games with applications to complexity theory*; Symposium on Theory of Computing (STOC), 1994.

[34] McKay D., Murray C., Williams R.; *Weak lower bounds on resource-bounded compression imply strong separations of complexity classes*; Symposium on Theory of Computing (STOC), 2019.

[35] Modanese A.; *Lower bounds and hardness magnification for sublinear-time shrinking cellular automata*; preprint, 2020.

[36] Müller M., Pich J.; *Feasibly constructive proofs of succinct weak circuit lower bounds*; Annals of Pure and Applied Logic, 2019.

[37] Newman I.; *Private vs common random bits in communication complexity*; Information Processing Letters, 39:67-71, 1991.

[38] Nisan N., Wigderson A.; *Hardness vs. randomness*; J. Comp. Systems Sci., 49:149-167, 1994.

[39] Oliveira I.C.; *Randomness and intractability in Kolmogorov complexity*; International Colloquium on Automata, Languages and Programming (ICALP), 2019.

[40] Oliveira I.C., Pich. J., Santhanam R.; *Hardness magnification near state-of-the-art lower bounds*; Computational Complexity Conference (CCC), 2019.

[41] Oliveira I.C., Santhanam R.; *Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness*; Computational Complexity Conference (CCC), 2017.

[42] Oliveira I.C., Santhanam R.; *Hardness magnification for natural problems*; Foundations of Computer Science (FOCS), 2018.

[43] Pich J.; *Nisan-Wigderson generators in proof systems with forms of interpolation*; Mathematical Logic Quarterly, 57(4), 2011.

[44] Pich J.; *Circuit lower bounds in bounded arithmetics*; Annals of Pure and Applied Logic, 166(1):29-45, 2015.

[45] Pich J.; *Mathesis universalis*; Literis, 2016.

[46] Pich J., Santhanam R.; *Strong co-nondeterministic lower bounds for* NP *cannot be proved feasibly*; Symposium on Theory of Computing (STOC), 2021.

[47] Pich J., Santhanam R.; *Towards* $\mathsf{P} \neq \mathsf{NP}$ *from Extended Frege lower bounds*; arXiv, 2023.

[48] Razborov A.A; *Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic*, Izvestiya of the Russian Academy of Science, 59:201-224, 1995.

[49] Razborov A.A.; *Pseudorandom generators hard for k-DNF Resolution and Polynomial Calculus*; Annals of Mathematics, 181(2):415-472, 2015.

[50] Razborov A.A, Rudich S.; *Natural Proofs*; Journal of Computer and System Sciences, 55(1):24-35, 1997.

[51] Rivest R., Sloan R.; *Learning complicated concepts reliably and usefully*; Conference on Artificial Intelligence (AAAI), 1988.

[52] Rudich S.; *Super-bits, demi-bits, and NP/qpoly-natural proofs*; Journal of Computer and System Sciences, 55(1):24-35, 1997.

[53] Santhanam R.; *Pseudorandomness and the Minimum Circuit Size Problem*; Innovations in Theoretical Computer Science (ITCS), 2020.

[54] Tal A.; *Computing requires larger formulas than approximating*; Symposium on Theory of Computing (STOC), 2017.

[55] Vadhan S.; *Learning versus refutation*; Conference on Learning Theory (COLT), 2017.

[56] Vadhan S., Zheng C.J.; *A uniform Min-Max theorem with applications in Cryptography*; International Cryptology Conference (CRYPTO), 2013.