Restricted Distribution Automatizability in PAC-Semantics

Brendan Juba*
Harvard University
bjuba@alum.mit.edu

April 1, 2014

Abstract

We consider the proof search (*"automatizability"*) problem for propositional proof systems in the context of *knowledge discovery* (or *data mining* and *analytics*). Discovered knowledge necessarily features a weaker semantics than usually employed in mathematical logic, and in this work we find that these weaker semantics may result in a proof search problem that seems easier than the classical problem, but that is nevertheless nontrivial. Specifically, if we consider a knowledge discovery task corresponding to the unsupervised learning of parities over the uniform distribution from partial information, then we find the following:

- Proofs in the system *polynomial calculus with resolution (PCR)* can be detected in quasipolynomial time, in contrast to the $n^{O(\sqrt{n})}$-time best known algorithm for classical proof search for PCR.
- By contrast, a quasipolynomial time algorithm that distinguishes whether a formula of PCR is satisfied a $1-\epsilon$ fraction of the time or merely an ϵ-fraction of the time (for polynomially small ϵ) would give a randomized quasipolynomial time algorithm for NP, so the use of the promise of a small PCR *proof* is essential in the above result.
- Likewise, if integer factoring requires subexponential time, we find that bounded-depth Frege proofs cannot be detected in quasipolynomial time.

The final result essentially shows that negative results based on the hardness of interpolation [29, 13, 11] persist under this new semantics, while the first result suggests, in light of negative results for PCR [22] and resolution [2] under the classical semantics, that there are intriguing new possibilities for proof search in the context of knowledge discovery and data analytics.

*Supported by ONR grant number N000141210358.

1 Introduction

Consider the following *goal-driven knowledge discovery (data mining and analytics)* problem proposed by Juba [24], building on earlier work by Khardon and Roth [27, 28]. We have access to a data set consisting of partially specified examples $\rho^{(1)}, \ldots, \rho^{(m)}$, i.e., each $\rho^{(i)} \in \{0, 1, *\}^n$, where $*$ is an unspecified value. We assume that these examples have been produced by first drawing a complete example $x^{(i)} \in \{0, 1\}^n$ from an unknown, "ground truth" distribution D, and that some of the attributes have been hidden by a second random process M. That is, each $\rho^{(i)}$ is drawn i.i.d. from a distribution $M(D)$ with some well-defined "ground truth" $x^{(i)}$ for the values of all of the attributes. We are now given a *query* Boolean formula φ, and our objective is to decide whether the data set *provides empirical evidence* that φ is satisfied with high probability over D in the following sense. We wish to guarantee that whenever there are some formulas $\psi_1, \ldots, \psi_k$ that can be verified to simultaneously hold with high probability on D using our partial examples,[1] that then complete a *proof* of the query φ, that we then *certify* φ as satisfied with high probability under D. At the same time, we also wish to guarantee that if the query φ is not satisfied with high probability under D, then we are unlikely to certify it as being so. (We will discuss the motivation for this problem shortly.)

In this work, we will be considering this problem from the angle of *proof complexity*: if our query testing problem can be solved (relatively) efficiently for a proof system, we will say that the proof system is *"PAC-automatizable"* (the formal definition appears in Section 2.3). As we will review next, this *integrated* data analytics approach is not only sufficient to capture a variety of applications, but moreover provides more power (in multiple respects) than a standard, two-stage, "learn-then-analyze" approach. In particular, in this work, we will see evidence that there may be advantages to the use of integrated algorithms from the standpoint of the power of the proof systems: For a *limited* but *demonstrably nontrivial special case* of distributions over partial examples $M(D)$, we obtain a substantial speedup for query testing using the proof system *"polynomial calculus with resolution" (PCR)*, which reasons about the solutions to systems of (arbitrary-degree) polynomial equations. As we will argue, this suggests that data-driven applications will benefit from a deeper integration of learning into the core algorithms.

1.1 Motivation and context

In *knowledge discovery* and *data analytics*, the extraction of *knowledge* from a data set is not an end goal, only an intermediate step. The final goal is to make a *decision* based on the data, perhaps about the allocation of resources in a business or course of treatment in medicine. In this work, we are modeling such decisions as a Boolean query formula. Even for the relatively simple proof system of *resolution* in which these queries are DNFs, such a representation is rich enough to capture some relatively interesting problems such as *planning in a STRIPS environment* [25]. Also, although we do not directly address such proof systems in this work, this broader perspective allows us to consider integer linear program queries against the *cutting planes* proof system, for example, which would capture a variety of real-world optimization problems (and thus presents an intriguing challenge for future work).

[1] Verified here by plugging in the partial example for the variables of the formulas, performing the natural connective-wise simplification of the formula, and checking if the resulting simplified formula is the constant "true." See Section 2.2 for a full definition.

The current state of the art in *data science* is not a "closed loop," however. The business of extracting knowledge from the data is the responsibility of a human "data scientist" who, at his or her discretion, opts to invoke one or more of the usual machine learning algorithms on a given data set. These algorithms may be a "supervised" algorithm like linear regression, in search of a linear relationship among some of the data attributes, or perhaps an "unsupervised" algorithm that searches for deviations from a product distribution or some other ad-hoc measure of "interestingness" of properties of the data. In each case, the algorithms can be viewed as producing a *rule* satisfied by the data, e.g., $\psi(x, \vec{y}) = [x = \langle \vec{\beta}, \vec{y} \rangle + \beta_0 \pm \epsilon]$ (properly, a pair of linear inequalities) in the case of linear regression. There may be exponentially many such rules, so the data scientist cannot realistically hope to list them all. Without further guidance, there is therefore no guarantee that the data scientist will manage to discover all of the *relevant* rules necessary to inform the final decision, even when there exists a (polynomially) small list of such rules.

Moreover, the machine learning algorithms employed by the data scientist may be fundamentally incapable of producing the rules required to inform a decision. Recent work by Daniely et al. [15] provides new evidence that DNF/CNF representations may not be learnable by *any* efficient algorithm. Moreover, this result implies [26] that supervised learning algorithms, even for the relatively weak class of *conjunctions*, may be incapable of tolerating adversarial noise (*"agnostic"* learning). This is troubling, as there is no reason to expect that the kinds of simple representations that can be efficiently learned and reasoned about actually capture the "ground truth" (meaning here, hold with probability 1 over D). For example, in the case of STRIPS environments (originally proposed by Fikes and Nilsson [21]) that we mentioned previously, an environment is modeled as a list of deterministic rules of the form "(partial) state and action implies effect" (each formally encoded as clauses). We would only expect these rules to serve as *approximations* to the complicated, true dynamics of any real-world environment [31]. This is, nearly verbatim, the setup assumed in agnostic learning [26], and so seems to be beyond the reach of (stand-alone) learning algorithms.

Perhaps surprisingly, it turns out that these difficulties are an artifact of the separation of the learning algorithm from the query. A result by Juba [24], building on earlier work by Khardon and Roth [27], shows that for nearly every proof system in the literature (those that are *"natural"* in the sense of Beame et al. [6]), the problem of (*"implicitly"*) learning all of the relevant rules to certify a query is feasible, even under an agnostic learning model, *provided that we give up on identifying the rules explicitly.* Indeed, even in an "agnostic" model, the query itself presents merely a problem of *approximate counting* or *testing* using examples: we are only asking the probability that a query φ is satisfied on the complete examples from D. This approximate counting/testing problem then does not require that one actually solve the (often intractable) optimization problem of identifying rules with a near-optimal error rate. Nevertheless, we stress again, *logical queries* against such implicitly learned representations suffice to solve other problems, such as producing plans for STRIPS environments [25] for example.

Khardon and Roth [27] were the first to observe that such integration of logical queries and learning may be beneficial. They showed that, in a *complete* information setting (where the examples are taken from D directly), one can efficiently answer *all* $O(\log n)$-CNF *entailment queries* against any DNF representation that is always satisfied over D, even though it is not known how to learn DNFs in such a model, and such queries may be NP-hard. We stress that their formulation makes *no* reference to *proofs* of the queries from the "learned" DNF: their algorithm decides whether the query is entailed (by such a DNF) or falsified with noticeable probability. While their result is therefore quite striking from both the standpoint of learning theory and automated reasoning, its

applicability is severely limited by the requirement of complete examples. This not only denies us the ability to model data sets in which some "ground truth" attributes cannot be directly observed, as for example the presence of a disease might not be in the case of medicine or the true preferences of a user might not be in the case of internet advertising. It also prevents us from setting up an optimization problem in which some attributes, capturing an optimal allocation (say), are likewise unknown. Khardon and Roth, keeping in the spirit of their first approach, proposed an algorithm for using partial examples [28] that avoided the use of theorem-proving techniques. This approach could only efficiently handle queries consisting of k-CNFs (for small k), however.

Khardon and Roth's work raises the question of whether or not the consideration of proof systems and theorem-proving are essential to our query testing problem. Although Juba [24] showed how to make use of partial examples for richer query representations by restricting our attention to testing for the existence of proofs of the query, it is not immediately clear that this introduction of proofs was essential in general. In this work, we will address this question by showing that even for rather benign partial information settings, the strength of proof we are testing for may impact the complexity of the problem dramatically: for the same query representation, the query testing problem may be NP-hard without any proof complexity promise (Theorem 34), as hard as breaking Diffie-Hellman key exchange[2] under a weak proof complexity promise (Theorem 33), and solvable in quasipolynomial time under the stronger proof complexity promise of having a (quasi)polynomial size PCR proof (Theorem 28).

We view this final result as particularly significant for two reasons. First, it stands in contrast to what is believed to be possible for classical proof search, i.e., without access to partial examples: The best known algorithm for PCR theorem-proving, due to Clegg et al. [14], runs in time $n^{O(\sqrt{n})}$ for polynomial-size proofs over n variables. Even the weaker system of resolution is conjectured to not have such quasipolynomial-time theorem-proving algorithms (cf. Alekhnovich and Razborov [2]). Therefore, it suggests that the setting in which learning from partial examples is integrated into proof search is significantly different from the classical setting, and specifically that proof search may actually become easier in such an integrated setting. Second, in the context of theorem-proving (or "automatizability" in the usual language of proof complexity), a quasipolynomial time algorithm is relatively efficient. Even the relatively weak system of treelike resolution, in which intermediate derivations are not reused, is only known to have quasipolynomial time algorithms [7, 9]. Nevertheless, until the development of modern "clause learning" SAT-solvers, algorithms for finding treelike resolution proofs (via variants of DPLL [17, 16] which are not necessarily even so efficient) were widely used. Even quasipolynomial-time algorithms (as opposed to truly polynomial time algorithms) for richer proof systems may be of great significance.

1.2 Setting and results

In this work, we will focus primarily on a single, common family of query representations against a class of distributions encoding a common family of learning problems. We will obtain positive and negative results for this same class by varying the strength of the proofs we aim to test for. The primary class of queries we consider are given by a system of multivariate polynomial constraints, where the polynomials have rational coefficients. We restrict our attention to the Boolean solutions to these polynomials by assuming that for each indeterminate x_i there is a constraint $[x_i^2 - x_i = 0]$. Naturally, the query is satisfied when the complete example drawn from the distribution D satisfies

[2]Therefore, as hard as integer factoring [10].

this system of equations, and we wish to test whether or not the query is satisfied with high probability, given only access to partial examples.

Given the way our query testing problem is set up – we distinguish queries that we can "certify true" using our partial examples from queries that are "inconclusive" – a partial information model in which *all* information may be hidden is trivially equivalent to classical logical reasoning. The "data" in such a case is essentially nonexistent, which is clearly not the setting we had originally envisioned. We wish to restrict our attention to cases where the data for the learning problem provides some information. In the primary special case we consider, our partial examples will be produced in the following, standard way. For each ith attribute of each complete example drawn from the ground truth distribution D, a μ-biased coin (for some constant $\mu \in (0,1)$) is tossed to determine whether or not the attribute is replaced by a $*$ in the resulting partial example. This model first appeared in learning theory in the work of Decatur and Gennaro [18], and more recently was employed as the partial information model for *Population Recovery* [20, 41, 5, 34] (which we will discuss later). Elsewhere, this model is known as *"masking completely at random (MCAR)."*

The main learning problems we consider are an "unsupervised" version of *uniform distribution parity learning.* Recall that in the usual, supervised PAC-learning model [39], there is a distinguished "label" attribute x_ℓ that is guaranteed to be determined by some function f of the other attributes, where f is adversarially chosen from some pre-determined "concept class" $\mathcal{C}$. "Parity learning" then means that $\mathcal{C}$ is the class of parity functions: $x_\ell = \bigoplus_{i \in S} x_i$ for some $S \subseteq [n]$. In "uniform distribution" learning (in contrast to the original "distribution-free" model), the examples $x_1, \ldots, x_n$ are chosen uniformly at random, and x_ℓ is then determined by the unknown parity function. Equivalently, we could say that there is a *rule* (or *constraint*) $[\bigoplus_{i \in S \cup \{\ell\}} x_i = 0]$ and our examples $(x_1, \ldots, x_n, x_\ell)$ are chosen uniformly at random from the (Boolean) satisfying assignments to this rule. In this formulation, there is no longer any distinguished "label" bit. Indeed, we can naturally consider a *system* of such parity constraints, more generally of the form $[\bigoplus_{i \in S_j} x_i = b_j]$ for $S_j \subseteq [n]$ and $b_j \in \{0,1\}$, which define an $\mathbb{F}_2$-affine subspace. Our examples are now drawn uniformly at random from this affine subspace. We will simply refer to the distributions underlying these unsupervised learning problems as *"affine distributions"*. We chose to focus on this family of problems because it is mathematically simple and problems involving parity learning or parity formulas have provided hard examples in both proof complexity [38] and learning theory (in particular, here, in the work of Ben-David and Dichterman [8] on learning from partial information). Our choice is vindicated by the existence of both positive and negative results for the same learning problem, for different choices of the strength of reasoning problem. We briefly note that our positive result applies to a somewhat broader class of problems that we defer to the body of the work.

1.2.1 Statement and discussion of results

We can now state the results. Recall that we assume that we are given as inputs a query formula (given by a system of multivariate polynomial constraints) and partial examples drawn from some common distribution $M(D)$. We wish to soundly certify when the query is satisfied with high probability by the underlying distribution D over complete examples, and we will seek quasipolynomial time algorithms for these problems. We first note that this is NP-hard in general, even for affine distributions masked completely at random:

Theorem 1 (Basic negative result, cf. Theorem 34) *Let $M(D)$ be an affine distribution masked completely at random (with constant $\mu \in (0,1)$). Unless* NP *has quasipolynomial-time randomized*

algorithms, there is no quasipolynomial time algorithm (in n and $1/\epsilon$) that uses examples from $M(D)$ to distinguish systems of multivariate polynomials that are satisfied with probability $1-\epsilon$ by D from those that are satisfied with probability at most ϵ.

By contrast, if we only seek to test for the existence of a small proof from some rules (premises) that are verifiable from the partial information, then a quasipolynomial time algorithm sometimes *does* exist, depending on the strength of the proof system. Recall that we say that a proof system is *"PAC-automatizable"* (in time T) if a sufficiently fast algorithm for the following problem exists:

- *Reject* queries that are not satisfied with high probability under the distribution on complete examples D.
- *Accept* queries for which there exist some formulas $\psi_1, \ldots, \psi_k$ that can be checked in a standard way (defined precisely in Section 2.2) using the partial examples from $M(D)$, that in turn complete a polynomial-size proof of the query in the proof system.

In this way, the algorithm implicitly learns some $\psi_1, \ldots, \psi_k$ using examples from $M(D)$, and tests for a proof of the query from such learnable formulas.

Theorem 2 (Positive result, cf. Theorem 28) *Polynomial calculus with resolution (PCR) is PAC-automatizable for affine distributions masked completely at random in quasipolynomial time.*

Comparing Theorems 34 and 28, we see that the *promise* of some structure, such as provided by a small PCR proof, crucially reduces the complexity of our query testing problem, even under a rather benign partial information model and limited family of learning problems. This stands in contrast to Khardon and Roth's *complete* information setting [27] in which entailment queries for relatively rich representations could be answered without the need for any considerations of proof complexity.

We stress that the *"integrated"* formulation of this problem is crucial: in certifying a query, the algorithm is not required to produce either an explicit representation of the learned premises $\psi_1, \ldots, \psi_k$, or an explicit proof of the query from these formulas. Whereas earlier we discussed the advantages of such an integrated formulation from the standpoint of learning theory, here we note that it provides a means to avoid difficulties in proof complexity. In particular, Galesi and Lauria [22] (building on earlier work by Alekhnovich and Razborov for resolution [2]) gave evidence that the proof search problem for PCR is intractable.[3] These works rely on a reduction that uses the *size of the smallest proof* to solve a hard optimization problem. Since the integrated formulation (as we will see) avoids producing an explicit, complete proof of the query, it cannot be invoked in such reductions in any obvious way.

Nevertheless, we can also show that some of the stronger negative results for proof search, namely those based on the infeasibility of "interpolation," as pioneered by Krajíček and Pudlák [29], can be carried over to negative results for our PAC-automatizability problem. Specifically, Bonet et al. [11], building on earlier work by Bonet, Pitassi, and Raz [13], showed that the proof search problem for *"bounded depth Frege systems"* (recalled in Section 2.3.2) is intractable if factoring requires subexponential time. We obtain an analogue of this result, even when given examples from an affine distribution that is masked completely at random.

[3] Although the technique used in those works cannot address the question of whether or not quasipolynomial-time algorithms for the proof search problem exist, the authors explicitly conjecture that such algorithms do not exist.

Theorem 3 (Negative result, cf. Theorem 33) *If integer factoring requires subexponential time, then bounded-depth Frege is not PAC-automatizable for affine distributions masked completely at random in quasipolynomial time.*

Contrasting Theorem 33 with Theorem 28, we observe that the *strength* of the proof complexity promise given to our algorithm crucially affects its time complexity. Although bounded-depth Frege is a proof system in which the lines of the proof are each given by an AC_0-circuit[4] and so the queries addressed by such proofs are *not* a system of multivariate polynomial constraints, we note that this is not the source of the difficulty. Indeed, these formulas comprising the proof (and therefore the corresponding class of queries) can be translated into a system of quasipolynomial-size multivariate polynomial constraints at the cost of a small increase in the error. This is accomplished by a technique presented by Aspnes et al. [3] for characteristic zero fields, building on an earlier, classic work by Razborov [36] that used such a transformation of AC_0 circuits to polynomials over fields of finite characteristic. Thus, the promise of a small bounded-depth Frege proof of a query implies that the query has an encoding as a system of multivariate polynomials, and so the difficulty does not lie in the use of a different query representation.

Relationship to Population Recovery. *Population recovery* was first studied by Dvir et al. [20] and essentially solved in a sequence of later works [41, 5, 34]. In these works, partial examples drawn from an unknown distribution are presented, and the objective is to reconstruct the distribution (or merely the "heavy" portion of the distribution, in the case of the work by Batman et al. [5]) up to some additive error. As mentioned previously, we use the same partial information model as employed in these works. Naturally, if one can recover a sufficient measure of the distribution, then one can solve our approximate query testing problem directly, so whenever these algorithms apply, no proof complexity promise is required. On the other hand, population recovery is only efficient when either the support of the distribution is small (e.g., in Wigderson and Yehudayoff [41] or Moitra and Saks [34]) or when the distribution has almost all of its mass concentrated on a small number of heavy assignments, in the work by Batman et al. [5]. In general, our affine distributions can have high entropy and large support, and hence these algorithms for population recovery are not efficient in the setting we consider here. Indeed, we know that the query testing problem is sometimes NP-hard for affine distributions (in the absence of a proof complexity promise), so we don't expect the population recovery algorithms to solve our problem in general in any straightforward way. The bottom line is that PAC-automatizability and population recovery, in spite of presenting learning tasks for the same partial information model, are generally incomparable problems.

1.3 Techincal overview

We now provide an overview of how our results are obtained, highlighting the techniques involved. In what follows, we will denote examples drawn from a distribution D masked completely at random with bias μ by $M_\mu(D)$.

[4]Where, recall AC_0 is the class of polynomial-size, unbounded fan-in circuit families of *constant depth*; here, we mean that we consider proofs with some externally fixed constant depth bound.

1.3.1 Overview of the positive result, Theorem 28

The starting point for our positive results are the prior work by Juba [24], showing how PAC-automatizability can be reduced to classical automatizability whenever the proof system is *natural* (again, in the sense of Beame, Kautz, and Sabharwal [6]). That is: it provides a simple technique for searching over conjunctions of premises ψ such that $\psi|_\rho$ simplifies to the constant 1 for $\rho \in M_\mu(D)$ with probability $(1-\epsilon+\gamma)$ (we say ψ is $(1-\epsilon+\gamma)$-*testable*) and such that there is a proof of the query φ from ψ in the given proof system. The observation is that for a natural proof system, when every step of the proof is hit by the restriction ρ, the desired premises ψ will vanish from the proof with high probability. Thus, it suffices to check the provability of $\varphi|_\rho$ on various partial examples ρ drawn from $M_\mu(D)$, and the testable premises are incorporated "for free." We won't focus further on these classes of testable premises in this work; instead, our focus in this work will be on how these restrictions drawn from $M_\mu(D)$ cause polynomial calculus with resolution proofs to simplify.

The uniform distribution provides an informative special case of how such simplification occurs. Consider any monomial of degree at least d: in a partial example ρ drawn from $M_\mu(U_n)$, each literal ℓ_i in the monomial is set to 0 with probability $\mu/2$. Therefore, when the monomial is hit by the restriction ρ, it simplifies to 0 with probability $1-(1-\mu/2)^d$; if $d \geq \frac{2}{\mu}\ln\frac{1}{\gamma}$, then the monomial only "survives" with probability γ. So, in a polynomial calculus proof of size $P(n)$ (consisting of $P(n)$ monomials), a union bound over the monomials yields that with probability $1-\delta$, only monomials of degree at most $O(\frac{1}{\mu}\ln\frac{n}{\delta})$ survive. The algorithm of Clegg, Edmonds, and Impagliazzo [14] shows that degree-d polynomial calculus is automatizable in time $n^{O(d)}$; therefore, the basic reduction of our PAC-automatizability problem (for the uniform distribution) produces (YES-)instances which are automatizable in quasipolynomial time.

Affine distributions are not quite so simple. Nevertheless, we show that when polynomial calculus proofs are hit with restrictions from affine distributions, the surviving monomials have a special structure that similarly allows the proofs to be simulated in low degree. We observe that affine distributions feature a *"bias gap":* conditioned on any partial assignment of the variables, the assignment to any single further variable either remains unbiased or is fixed to a single value. (In the body of the work, we will see how this condition can be relaxed slightly, generalizing our algorithm to a larger class of distributions.) On account of this bias gap, we show that a monomial of logarithmic degree can only survive (with probability $\delta/P(n)$ for a size $P(n)$ proof, again) when, conditioned on partial assignments that set $O(\log n/\delta)$ of the indeterminates to 1, other indeterminates are also fixed to take value 1. This property follows from a couple of lemmas. The first lemma generalizes our observation about the uniform distribution to monomials in which the variables are unbiased:

Lemma 4 (cf. Lemma 23) *Let $x^{\vec{\alpha}}$ be a monomial of degree at least $\frac{2}{\mu}\ln\frac{1}{\delta}$ such that for any literal ℓ of $x^{\vec{\alpha}}$ and any submonomial $x^{\vec{\alpha}'}$ of $x^{\vec{\alpha}}$ without ℓ, ℓ is unbiased conditioned on $x^{\vec{\alpha}'}$ surviving. Then $x^{\vec{\alpha}}|_\rho = 0$ on $\rho \in M_\mu(D)$ with probability $1-\delta$.*

The second lemma considers when the variables may be fixed to take value 0:

Lemma 5 (Lemma 25) *Let $x^{\vec{\alpha}}$ be a monomial of degree $2d$ for $d \geq \frac{1}{\mu}\ln\frac{1}{\delta}$ such that for every submonomial of $x^{\vec{\alpha}}$ of degree d there is some further submonomial $\ell x^{\vec{\alpha}'}$ such that conditioned on $x^{\vec{\alpha}'}$ surviving, ℓ is set to 0. Then with probability $1-\delta$, there is an unmasked variable of $x^{\vec{\alpha}}$ that is fixed to zero under $\rho \in M_\mu(D)$.*

Together, these simple lemmas indeed establish our claimed "structure" of the surviving monomials: As a consequence of the bias gap in affine distributions, every sufficient degree ($\frac{4}{\mu}\ln\frac{P(n)}{\delta}$) submonomial of a surviving monomial in our polynomial calculus proof must have some variable that is fixed to 1, conditioned on the rest of the submonomial surviving. Indeed, otherwise either there is a degree $\frac{2}{\mu}\ln\frac{P(n)}{\delta}$ submonomial satisfying the conditions of the first lemma, or else the conditions of the second lemma are satisfied, and either way the monomial cannot survive with probability greater than $\delta/P(n)$, so by a union bound, none survive.

We can exploit this structure of the monomials now by learning *all* of the small-degree monomial constraints on the distribution's support. Learning these monomials is fairly straightforward: Logarithmic-degree monomials are simultaneously unmasked (under M_μ) with inverse-polynomial probability, and since the masking is independent of the underlying assignment drawn from D, we can obtain unbiased "complete" examples for each monomial, which allows us to apply standard learning techniques, summarized in Lemma 26. The heart of our analysis now is that we show that these small monomial constraints allow us to simulate (the surviving portion of) an arbitrary PCR proof in low degree: We know that every logarithmic-width submonomial of the surviving monomials has a member that is fixed to 1 conditioned on the rest of the monomial surviving; then taking the *complement* of this member (recalling that this is PCR, so we have such complement indeterminates) yields a logarithmic-width monomial that is consistently 0, and hence is among the learned monomial constraints. We can therefore eliminate these variables that are fixed to 1 by resolution steps using our learned monomial constraints. That is, if we quotient out the ideal generated by the small-degree monomial constraints, every step of the restricted PCR proof has a low-degree representative.

The only complication now is that we need to ensure that we can derive the low-degree representatives of subsequent steps of the (restricted) proof by a low-degree PCR derivation. For example, we might try to maintain the same "reduced" monomials across different parts of the proof—say some canonical reduction such as lex. order. The difficulty with this approach is that the multiplication rule of polynomial calculus may change the canonical reduction substantially. What turns out to work is that we suppose that we only make greedy reduction steps on each monomial of the proof according to *some* arbitrary order using our low-degree monomials of the ideal, and we are given that the resulting representative is of low degree. We then show that there is a low-degree PCR derivation of one reduction ordering from any other in Lemma 27: We show that the *product* of two corresponding reduced monomials of a proof step can be derived in low degree by following one reduction order in reverse and eliminating those indeterminates missing from the product from the rest of the learned, low-degree monomial constraints of the ideal. Either of the reduced monomials can be derived in low degree from their product using these modified monomials from the ideal (with the unnecessary variables already eliminated) to perform the reduction. If the reduced monomials and the learned monomial constraints all have degree d, then the overall derivation requires only degree $3d$. This completes the proof: there is a low-degree simulation of the surviving PCR proof, which is therefore automatizable by the algorithm of Clegg, Edmonds, and Impagliazzo [14]. As we obtain a degree bound of $d = O(\frac{1}{\mu}\log\frac{n}{\delta})$ on the polynomials involved, it turns out that the overall algorithm runs in time $m \cdot n^{O(d)}$, where $m = \frac{2}{\gamma^2}\ln\frac{2}{\delta}$; the initial learning of all of the degree-d monomials can be done using $n^{O(d)}$ time and examples, so the time spent on the proof search dominates.

1.3.2 Overview of the negative results, Theorems 34 and 33

The main observation underlying both of the negative results is that affine distributions present hard examples because sufficiently large parity constraints are invisible when even one variable participating in the parity is masked: formally, if all of the constraints are sufficiently large, then the masked affine distribution is indistinguishable from masked examples of the uniform distribution. (Indeed, this is the main content of Ben-David and Dichterman's result for the RFA model [8].) When the query can compute the corresponding large parities, these parity functions are constrained to take a hidden value (depending on the choice of affine distribution) with probability 1. Intuitively, since settings of the parities are hidden from an algorithm that only has access to partial examples, and they can encode an arbitrary fixed assignment, the algorithm will need to somehow "rule out" the possibility that *any* such hidden assignment falsifies the query. The formalization of this intuition proceeds, naturally, by showing (in Lemma 31) that for any fixed setting of hidden variables we desire, an algorithm that has no access to any examples can always simulate access to masked examples of an affine distribution with parity encodings of these hidden values by simply masking examples of the uniform distribution.

Towards showing the NP-hardness of polynomial queries, we take an arbitrary 3DNF and substitute parity encodings for each of the variables: if the 3DNF is a tautology, so is this substitution instance, and otherwise there is an affine distribution (indistinguishable from the uniform distribution under masking) in which it is falsified with probability 1. It is immediate that distinguishing whether such substitution instances are satisfied with probability 1 or 0 is NP-hard, even given masked examples. To fool quasipolynomial-time algorithms, we only need polylogarithmic-size parity constraints, and the work of Bonet et al. [11] in particular establishes that polylogarithmic size parity formulas have AC_0 circuits. We then convert these AC_0 circuits down to polynomial queries using randomized polynomial encodings (of Razborov for finite characteristic [36] and Aspnes et al. for characteristic zero [3]), at the cost of introducing some small error ϵ and an increase to quasipolynomial size monomial expansion representations. The result is a randomized quasipolynomial-time reduction, showing that it is NP-hard to distinguish whether the resulting polynomial queries are satisfied with probability at least $1 - \epsilon$ or at most ϵ. This is Theorem 34.

The argument that bounded depth Frege is not PAC-automatizable in quasipolynomial time is similar: the basic building block is the result by Bonet, Pitassi, and Raz [13] that automatizing TC_0-Frege gives an attack that breaks the Diffie-Hellman key exchange protocol [19], which in turn yields integer factoring [10]. Here, we use parity encodings to hide the unknown values of the Diffie-Hellman protocol.[5] Again, Lemma 31 is indeed saying that there is a simulator that generates such "leakage" as provided by our parital examples, so any algorithm that PAC-automatizes these substitution instances (using examples) can be combined with the simulator to break the security of the Diffie-Hellman protocol. This is summarized in Theorem 32, showing a range of hardness results for PAC-automatizing TC_0-Frege for a corresponding range of security assumptions for Diffie-Hellman key exchange. Now, to obtain the promised hardness of bounded-depth Frege based on integer factoring requiring subexponential time, we consider the reduction of Bonet et al. [11], showing how TC_0-Frege proofs in which the parity and threshold gates have polylogarithmic fan-in can be converted to bounded-depth Frege proofs. With a little more care, we

[5] The intuition here is that our masking process is "leaking" some of the secrets of the two parties, so we use a leakage-resilient encoding of these values. Since the masking completely at random is such a weak form of leakage, the parity encoding is secure. This use of the parity encoding originally appeared in the work by Ishai, Sahai, and Wagner [23].

show that the reduction can convert *quasipolynomial-time* algorithms for automatizing bounded-depth Frege to arbitrarily good subexponential-time algorithms for integer factoring. By invoking Lemma 31 again to simulate access to the affine distributions encoding the secret values in the Diffie-Hellman instances used in the factoring reduction, we therefore show that quasipolynomial time PAC-automatizing bounded-depth Frege indeed likewise yields $2^{O(n^\eta)}$-time algorithms for factoring for every $\eta > 0$. As in the original result for classical non-automatizability of bounded-depth Frege by Bonet et al., this contradicts an assumption that there is some absolute smallest $\eta > 0$, and Theorem 33 follows.

2 Our setting: background and approach

2.1 PAC-Semantics

PAC-Semantics (for a logic) was introduced by Valiant [40] to capture the kind of validity possessed by statements produced by PAC-learning algorithms; for example, if one uses a PAC-learning algorithm to learn a conjunction $\wedge_j x_{i_j}$ to match the "labels" given by another variable x_t from examples from a distribution D, then the formula $[\wedge_j x_{i_j} \equiv x_t]$ is (probably) approximately valid in the following sense:

Definition 6 ($(1-\epsilon)$-valid) *Given a distribution D over $\{0,1\}^n$, we say that a Boolean formula φ is* $(1-\epsilon)$-valid *if $\Pr_{x \in D}[\varphi(x) = 1] \geq 1 - \epsilon$. If $\epsilon = 0$, we say that the formula is* perfectly valid.

Of course, the definition makes sense for other kinds of formulas, not just the kind of equivalences one obtains in this way from supervised learning. It is not hard to show (by a union bound) that any classical logical inference can be applied to formulas possessing this weaker kind of validity, as long as we allow for further loss in the approximation.[6]

Proposition 7 (Classical reasoning in PAC-Semantics [24]) *Let $\psi_1, \ldots, \psi_k$ be formulas such that each ψ_i is $(1-\epsilon_i)$-valid under a common distribution D for some $\epsilon_i \in [0,1]$. Suppose that $\{\psi_1, \ldots, \psi_k\} \models \varphi$ (in the classical sense). Then φ is $1-\epsilon'$-valid under D for $\epsilon' = \sum_i \epsilon_i$.*

The main problem that we wish to consider, introduced in prior work [24], is that of certifying the high validity of a given *query* formula using *examples* from an unknown distribution. In the present work, we will focus primarily on a special case of this problem in which the learning problem is essentially unsupervised learning of parity formulas under the uniform distribution. More precisely, these are uniform distributions over a $\mathbb{F}_2$-affine subspace, which we refer to as *affine distributions*:

Definition 8 (Affine distribution) *For any solvable linear system over $\mathbb{F}_2$ $Ax = b$, the distribution over $\{0,1\}^n$ that is uniform over solutions to the linear system is an* affine distribution.

Affine distributions are an "unsupervised" generalization of the problem of learning parity formulas under the uniform distribution in the following sense. One way of viewing such problems is that there is an unknown parity constraint over the example and label bits, and the examples are generated by choosing solutions to this equation uniformly at random. The difference is that in an affine distribution, there may be many constraints, and there is no distinguished label bit. (This is the sense in which it is unsupervised.)

[6] It also is not hard to show that as long as the distributions are arbitrary, the union bound is tight here [24].

We will give both positive and negative results for our query testing problem against affine distributions. We will actually obtain our positive results for a more general class of distributions (and hence learning problems) in which the (conditional) biases of variables is either strong or weak (and not of moderate strength):

Definition 9 (Bias gap) *We will say that a distribution D over $\{0,1\}^n$ has a* width-w $(\beta, 1-\gamma)$ bias gap *if for any monomial of $k \leq w$ literals $\prod_{i=1}^{k} \ell_i$ that takes value 1 with nonzero probability and any variable x, either $\Pr[x = 1 | \prod_{i=1}^{k} \ell_i = 1] \geq \beta$ and $\Pr[x = 0 | \prod_{i=1}^{k} \ell_i = 1] \geq \beta$ or else for some $b \in \{0,1\}$, $\Pr[x = b | \prod_{i=1}^{k} \ell_i = 1] \geq 1-\gamma$. In the former case, we say that x is β-*balanced for the monomial $\ell_1 \cdots \ell_k$, *and in the latter we say that, respectively, x (for $b = 1$) or $\neg x$ (for $b = 0$) is $1-\gamma$-*implied by $\ell_1 \cdots \ell_k$, *denoted as $\ell_1 \cdots \ell_k \rightsquigarrow x$ (or $\ell_1 \cdots \ell_k \rightsquigarrow \neg x$, respectively).*

Bias gap distributions are reasonably natural. For example, the simple *"pure document"* probabilistic corpus model introduced by Papadimitriou et al. [35] to analyze Latent Semantic Indexing generally features a nontrivial bias gap:

Example 10 (Pure document topic model [35]) *The* pure document topic model *is a probabilistic model of document generation. We suppose that documents are represented by the set of words appearing in them. A document is then generated in a two-stage process in which a (latent)* topic variable *is first sampled, where each outcome of the topic variable is associated with one member of a family of disjoint sets of* primary terms*. The set of words associated with this topic is obtained by the union of these primary terms with a set of generic terms (shared across topics); each word has an associated probability in the range $[\epsilon, \tau]$ for some small constants ϵ and τ. The document itself is then sampled by independently tossing a biased coin for each word in the topic set, including it with its given probability. If the overall probability of the topic words appearing in the document is at most $\delta \ll \epsilon$, which holds, for example, if each topic has sufficiently small probability of being chosen (relative to ϵ/τ), then the distribution has a width-w $(\epsilon, 1-\delta(1+\delta w))$-bias gap.*

The class of bias gap distributions subsumes that of affine distributions. Indeed, affine distributions are a particularly simple example of a large collection of distributions that have a very strong bias gap of $(1/2, 1)$ up to width n. Affine distributions turn out to have a bias gap, since intuitively, if a monomial determines the value of another variable in a linear constraint, that literal is easily seen to be 1-implied, and otherwise the literal turns out to be uniformly distributed (and thus, 1/2-balanced).

Definition 11 (Constraints on monomials) *We say that there is a* constraint *on a monomial $x^{\vec{\alpha}}$ in an affine distribution given by the linear system $Ax = b$ if there is a linear combination of the rows of A such that the only nonzero entries are in indices i for which the corresponding variable appears in a variable of $x^{\vec{\alpha}}$.*

Lemma 12 *Let $x^{\vec{\alpha}}$ be a monomial and D be an affine distribution such that there are no constraints on $x^{\vec{\alpha}}$. Then the marginal distribution over the variables appearing in $x^{\vec{\alpha}}$ is uniform.*

Proof: Let $Ax = b$ be the linear system defining D. Let y denote the variables in $x^{\vec{\alpha}}$ and z denote the variables not appearing in $x^{\vec{\alpha}}$, and let A' be the submatrix of A formed by columns corresponding to variables in $x^{\vec{\alpha}}$, and A'' be the submatrix formed by columns of A corresponding

to variables not in $x^{\vec{\alpha}}$. Note that by converting the matrix into reduced form, we may verify that if for some y^* there were no solutions to the system $A''z = (b + A'y^*)$, then there must be some linear combination of the rows of A'' yielding the 0 vector. The corresponding linear combination of the rows of the linear system $Ax = b$ would be a constraint on $x^{\vec{\alpha}}$, which does not exist by hypothesis. Therefore, for every y^*, the system $A''z = (b + A'y^*)$ has solutions, and furthermore, since A'' is the same for every assignment y^*, it always has the same number of solutions. Thus, as D is uniform over these solutions (and y^*), the marginal distribution over the variables in $x^{\vec{\alpha}}$ is indeed uniform as claimed. ■

2.2 Partial examples

Answering queries using (complete) examples drawn from a distribution D is a trivial approximate counting task. By contrast, we will see that the task is much more difficult when some of the variables' settings are deleted from the examples, resulting in *partial* example assignments. Formally, we use the following model due to Michael [33]:

Definition 13 (Partial examples and masking) *A* partial example *ρ is an element of $\{0, 1, *\}^n$. We say that a partial example ρ is* consistent *with an assignment x if whenever $\rho_i \neq *$, $\rho_i = x_i$.*

A mask *is a function taking assignments to consistent partial examples. A* masking process *is a mask-valued random variable. We denote the distribution obtained by applying a masking process M to a distribution over assignments D by $M(D)$.*

Notice, this general definition permits arbitrary correlations between the hiding of indices by M in the partial example and the underlying example actually drawn from D. One cannot hope to beat worst-case performance at reasoning when given only examples where M and D are arbitrary: Any arbitrary setting of the masked indices may have appeared with (conditional) probability 1, so one must rule out the existence of any falsifying assignments before it is safe to report that a query is (highly) valid, and such a case is easily seen to be essentially equivalent to classical reasoning. The problem becomes interesting when the masking process is sufficiently restricted to permit generalization beyond the revealed partial information; one natural example of such a process that appears in many other works (starting with a work of Decatur and Gennaro [18] in learning) is the following:

Definition 14 (μ-independent masking process) *For any $\mu \in (0, 1)$, the* μ-independent masking process *(denoted M_μ) produces a mask m by tossing an independent μ-biased coin for each $i \in [n]$, putting $m(x)_i = x_i$ for all x if it comes up heads (with probability μ), and otherwise (with probability $1 - \mu$) putting $m(x)_i \equiv *$.*

We will be interested in the regime where μ is a constant (e.g., 1%).

An important notion in proof complexity is that of a *restriction* of a formula; we can naturally interpret our partial examples as restrictions as follows:

Definition 15 (Restriction) *Given a formula $\varphi(x_1, \ldots, x_n)$ defined over linear threshold and parity connectives and a partial example $\rho \in \{0, 1, *\}^n$ we define* the restriction of φ under ρ, *denoted $\varphi|_\rho$, as follows by induction on the construction of φ:*

- *For any Boolean constant b, $b|_\rho = b$.*
- *For any variable x_i, if $\rho_i = *$, then $x_i|_\rho = x_i$, and otherwise (for $\rho_i \in \{0, 1\}$), $x_i|_\rho = \rho_i$.*

- *For a parity connective over $\psi_1, \ldots, \psi_k$, if $\ell \geq 1$ of the ψ_i (indexed by $i_1, \ldots i_\ell$) do not simplify to Boolean values under ρ, then (indexing the rest by $j_1, \ldots, j_{k-\ell}$)*

$$\oplus(\psi_1, \ldots, \psi_k)|_\rho = \oplus(\psi_{i_1}|_\rho, \ldots, \psi_{i_\ell}|_\rho, (\psi_{j_1}|_\rho \oplus \cdots \oplus \psi_{j_{k-\ell}}|_\rho))$$

  and otherwise it simplifies to a Boolean constant, $\oplus(\psi_1, \ldots, \psi_k)|_\rho = \psi_1|_\rho \oplus \cdots \oplus \psi_k|_\rho$.
- *A linear threshold connective $[\sum_{i=1}^k c_i \psi_i \geq b]$, $(c_1, \ldots, c_k, b \in \mathbb{Q})$ simplifies to 1 if*

$$\sum_{i:\psi_i|_\rho = 1} c_i + \sum_{i:\psi_i|_\rho \notin \{0,1\}} \min\{0, c_i\} \geq b$$

  simplifies to 0 if

$$\sum_{i:\psi_i|_\rho = 1} c_i + \sum_{i:\psi_i|_\rho \notin \{0,1\}} \max\{0, c_i\} < b$$

  and otherwise is given by

$$\left[\sum_{i:\psi_i|_\rho \notin \{0,1\}} c_i(\psi_i|_\rho) \geq \left(b - \sum_{i:\psi_i|_\rho = 1} c_i \right) \right].$$

That is, $\varphi|_\rho$ is a formula over the variables x_i such that $\rho_i = *$. We chose to define partial evaluation over this atypical basis of connectives because it enables us to define partial evaluation of *both* arithmetic formulas (which play a central role here) and the standard Boolean basis in a natural way. Recalling that our domain is Boolean, we define a monomial to be the AND of the literals, where we define AND and OR using the threshold connective and NOT using the parity connective in the natural way. We can now define a polynomial constraint $P(x) = 0$ using the conjunction of two linear-threshold connectives, $[P(x) \geq 0] \wedge [-P(x) \geq 0]$.

Definition 16 (Witnessing) *We define a formula φ to be* witnessed to evaluate to true *on a partial assignment ρ if $\varphi|_\rho$ simplifies to (the Boolean constant)* 1.

The constraints we will aim to learn are those witnessed to be true on *most* examples from $M_\mu(D)$.

2.3 Propositional proof systems

We begin by defining our central notion of interest:

Definition 17 (PAC-automatizability) *When we say that a proof system is* PAC-automatizable in time $T(N, 1/\epsilon, 1/\gamma, 1/\delta)$, *we mean that there is an algorithm that is given φ, $\epsilon, \gamma, \delta > 0$, and N as input and obtains samples from $M(D)$ for a given distribution D and masking process M. This algorithm runs in time $T(N, 1/\epsilon, 1/\gamma, 1/\delta)$ and with probability $1-\delta$ distinguishes φ that are $(\epsilon+\gamma)$-valid from φ that have a refutation of size N in the system from additional premises $\psi_1, \ldots, \psi_k$ such that $\psi_1 \wedge \cdots \wedge \psi_k$ simplifies to true under partial examples drawn from $M(D)$ with probability at least $1 - \epsilon + \gamma$.*[7]

In this work we will show that some proof systems are PAC-automatizable with a quasipolynomial time complexity for $M_\mu(D)$ when D is an affine distribution, while others are not.

[7] It turns out to be more natural and convenient to state this definition in terms of refutations of φ as we have done here, since this is how we will use the algorithms in most circumstances. We could equivalently (but less conveniently) have said that we were distinguishing when $\neg\varphi$ has a proof from when $\neg\varphi$ is *not* $(1 - \epsilon - \gamma)$-valid (where φ is the query actually provided to the algorithm).

2.3.1 Polynomial calculus

In polynomial calculus, originally introduced by Clegg et al. [14], formulas have the form of polynomial equations over an arbitrary nontrivial field $\mathbb{F}$ (for the present purposes, assume $\mathbb{F}$ is $\mathbb{Q}$, the field of rationals), and we are interested in their Boolean solutions. A set of hypotheses is thus a system of equations, and polynomial calculus enables us to derive new constraints that are satisfied by any Boolean solutions to the original system. More formally, for our Boolean variables $x_1, \ldots, x_n$, our formulas are equations of the form $[P(x) = 0]$ for $P \in \mathbb{F}[x_1, \ldots, x_n]$ (i.e., formal multivariate polynomials over the field $\mathbb{F}$ with indeterminates given by the variables). We require that the polynomials are represented as a sum of monomials: that is, every line is of the form $\sum_{\vec{\alpha} \in \mathbb{N}^n} c_{\vec{\alpha}} \prod_{i:\alpha_i \neq 0} x_i^{\alpha_i} = 0$ for coefficients $c_{\vec{\alpha}} \in \mathbb{F}$, where the products $\prod_{i:\alpha_i \neq 0} x_i^{\alpha_i}$ are the *monomials* corresponding to the degree vector $\vec{\alpha}$. For each variable, the proof system has a *Boolean axiom* $[x^2 - x = 0]$ (asserting that $x \in \{0, 1\}$). The rules of inference are *linear combination*, which asserts that for equations $[P(x) = 0]$ and $[Q(x) = 0]$, for any coefficients a and b from $\mathbb{F}$, we can infer $[a \cdot P(x) + b \cdot Q(x) = 0]$; and *multiplication*, which asserts that for any variable (indeterminate) x and polynomial equation $[P(x) = 0]$, we can derive $[x \cdot P(x) = 0]$. A *refutation* in polynomial calculus is a derivation of the polynomial 1, i.e., the contradictory equation $[1 = 0]$. We note that without loss of generality, we can restrict our attention to formulas in which no indeterminate appears in a monomial with degree greater than one—such monomials are *multilinear*. Intuitively this is so because the Boolean axioms assert that a larger power can be replaced by a smaller one.

In this work, we focus on an extension of polynomial calculus that can simulate resolution, known as *polynomial calculus with resolution (PCR)* that first appeared in the work of Alekhnovich et al. [1]. We introduce a new indeterminate $\bar{x}$ for each variable x, related by the *complementarity axiom* $[x + \bar{x} - 1 = 0]$ (forcing $\bar{x} = \neg x$). That is, roughly speaking, our indeterminates now correspond to *literals* (and we will abuse notation by speaking of the monomials as products of literals elsewhere). Monomials now can encode clauses, with the degree of the (multilinear) monomial equal to the width of the clause. We can *simulate a resolution step* (i.e., the cut rule) between one monomial of the form $\ell x^{\vec{\alpha}}$ from a larger polynomial constraint $[P(x) = 0]$ and a second monomial constraint of the form $[\neg \ell x^{\vec{\beta}} = 0]$ where $\vec{\beta} \leq \vec{\alpha}$ as follows. Using the multiplication rule on the complementarity axiom for ℓ, we can obtain $[\ell x^{\vec{\alpha}} + \neg \ell x^{\vec{\alpha}} - x^{\vec{\alpha}} = 0]$, and likewise using multiplication on the monomial $\neg \ell x^{\vec{\beta}}$, we can obtain the monomial constraint $[\neg \ell x^{\vec{\alpha}} = 0]$. Now, by adding an appropriate multiple of this constraint to $[P(x) = 0]$, and then subtracting the same multiple of $[\ell x^{\vec{\alpha}} + \neg \ell x^{\vec{\alpha}} - x^{\vec{\alpha}} = 0]$ from the result we obtain the constraint $[P'(x) = 0]$ in which the monomial $x^{\vec{\alpha}}$ is substituted for the monomial $\ell x^{\vec{\alpha}}$ i.e., ℓ is eliminated. Notice, we can obtain $[P(x) = 0]$ from $[P'(x) = 0]$ and the monomial constraint $\neg \ell x^{\vec{\beta}}$ again by repeating the derivation except subtracting the multiple of $\neg \ell x^{\vec{\beta}}$ and adding the mutiple of $[\ell x^{\vec{\alpha}} + \neg \ell x^{\vec{\alpha}} - x^{\vec{\alpha}} = 0]$, thus "reversing" the elimination of ℓ. This will be important since we can't freely use multiplication to simulate weakening of the monomials of $P(x)$ independently.

Following standard convention, we define the *size* of a PCR proof to be the *number of monomials* appearing in the proof. We also define the *degree* of the proof to be the maximum degree of any monomial appearing in the proof. The main result of Clegg et al. [14] establishes that the bounded-degree fragment of polynomial calculus (and PCR) is automatizable:

Theorem 18 (Theorem 3 of Clegg et al. [14]) *Degree-d PCR is automatizable in time $O((n^d + \ell)n^{2d})$ where n is the number of indeterminates and ℓ is the number of initial polynomials.*

We will be interested in the result of "plugging in" a partial assignment to each step of a PCR

refutation:

Definition 19 (Restricted proof) *Given a PCR refutation* Π *and partial assignment* ρ*, the* restriction *of* Π *under* ρ*, denoted* $\Pi|_\rho$*, is the proof obtained by substituting* $\varphi|_\rho$ *for each line* φ *of* Π*.*

A property shared by most propositional proof systems is that a restriction maps a proof to a proof of the restriction of the initial premises—indeed, Beame et al. [6] called such proof systems "natural." It is not hard to show that PCR has this property:

Proposition 20 (PCR is natural) *For any PCR refutation* Π *and any partial assignment* ρ*, the restriction* $\Pi|_\rho$ *is also a PCR refutation.*

We will see the value of Proposition 20 illustrated simply in Section 3.1. (It will, of course, also play a key role in establishing the main theorem, Theorem 28.)

2.3.2 Bounded-depth Frege

We will use the standard bounded-depth Frege sequent systems of propositional logic defined by Maciel and Pitassi [30]. In these systems, each line is of the form $A_1, \ldots, A_s \to B_1, \ldots, B_t$ (that is, the conjunction of the A_i's implies the disjunction of the B_j's) where each A_i and B_j is a bounded-depth formula from the appropriate class; we will consider two such classes in this work, AC_0 and TC_0. These classes both use the connectives $\vee$, $\wedge$, and $\neg$, and TC_0 additionally features the $\oplus_b$ connectives that are true iff the number of inputs that are true modulo 2 is $b \in \{0, 1\}$, and the Th_k connective, a threshold connective that is true iff at least k of the inputs are true. All of these connectives (except $\neg$) have unbounded fan-in. We define the *depth* of a formula to be the maximum depth of nesting of these connectives; the depth of a proof is then the maximum depth of any formula appearing in the proof. The *size* of the proof is the sum of the sizes of all of the formulas appearing in the proof. In the systems we consider, the depths will be bounded by some absolute constant (independent of the number of variables n) and the size of the proofs (and hence also their lengths) will be bounded by some polynomial in the number of variables. A complete description of the axioms and rules of inference are included in Appendix A.

The main result of Bonet et al. [11] is essentially a translation from TC_0-Frege proofs to AC_0-Frege proofs:

Theorem 21 (Theorem 6.1 of Bonet et al. [11]) *Suppose that* $\Gamma \to \Delta$ *has a* TC_0*-Frege proof of size polynomial in* n *in which the threshold and parity connectives all have fan-in bounded by* $O(\log^k n)$*. Then there is an* AC_0 *formula equivalent to* $\Gamma \to \Delta$ *that is polynomial-time computable from* $\Gamma \to \Delta$ *and has an* AC_0*-Frege proof of size greater by a factor of at most* $O(n^K)$ *where* K *depends only on* k*.*

Actually, Bonet et al. give specific definitions of such threshold and parity connectives. They do not explicitly state or argue for the efficient computation of the transformation (of the conclusion $\Gamma \to \Delta$) but this is essentially immediate. This translation enables non-automatizability for (sufficiently simple) specific TC_0-Frege formulas to be carried over to non-automatizability for AC_0-Frege formulas.

Substitutions. A *substitution* is a mapping from formulas to formulas defined by its action on free variables, taking them to arbitrary propositional formulas. For a substitution θ and propositional formula φ, we typically denote the result of applying θ to φ by $\theta\varphi$. Since the rules of inference for Frege systems remain instances of the same rules under any substitution, the following (essentially standard) fact is easily established:

Proposition 22 *Let θ be any substitution taking variables to depth-d_1 formulas, and suppose that there is a depth-d_2 Frege proof of φ from $\{\psi_1, \ldots, \psi_k\}$. Then there is a depth-$(d_1 + d_2)$ Frege proof of $\theta\varphi$ from $\{\theta\psi_1, \ldots, \theta\psi_k\}$.*

In particular, we will be substituting formulas consisting of a parity connective over variables for the variables of the original formula. This increases the depth of a formula by one and increases the size by at most a factor of n' (where there are n' variables in the substitutions). It thus takes TC_0-Frege proofs to TC_0-Frege proofs.

3 Automatizability of PCR

3.1 Automatizability for the uniform distribution

We start with the simpler special case of the uniform distribution (which we denote U_n), which will introduce a useful lemma for the general case, and help develop some intuition. The key observation here is that monomials that have sufficiently high degree are almost always witnessed to evaluate to zero:

Lemma 23 *Let $x^{\vec{\alpha}}$ be a monomial such that for any literal ℓ of $x^{\vec{\alpha}}$ and any submonomial $x^{\vec{\alpha}'}$ of $x^{\vec{\alpha}}$ without ℓ, ℓ is β-balanced for $x^{\vec{\alpha}'}$, and suppose $x^{\vec{\alpha}}$ has degree at least $\frac{1}{\mu\beta}\ln\frac{1}{\delta}$. Then $[x^{\vec{\alpha}} = 0]$ is witnessed to evaluate to true on $M_\mu(D)$ with probability $1 - \delta$.*

Proof: We note that ρ drawn from $M_\mu(D)$ are of the form $\rho = m(x)$ for m drawn from M_μ and x drawn from D independently. Suppose we construct a sequence of literals and submonomials of $x^{\vec{\alpha}}$ as we sample m and x in the following way: put $\vec{\alpha}_0 = \vec{0}$, and fix the entries of m in order until we encounter some unmasked entry corresponding to some literal ℓ_i in $x^{\vec{\alpha}}$; we then put ℓ_i in $x^{\vec{\alpha}_{i+1}} = x^{\vec{\alpha}_i}\ell_i$. Each literal of $x^{\vec{\alpha}}$ is thus included in some $\vec{\alpha}_i$ independently with probability μ. Now, $[x^{\vec{\alpha}} = 0]$ is witnessed true on ρ precisely when some ℓ_i is set to 0 in x, and if the first $i-1$ literals are set to 1 in x, then since ℓ_i is β-balanced for $x^{\vec{\alpha}_i}$, each ℓ_i is set to 0 in x with probability at least β when the first $i-1$ take value 1. Thus, the probability that none of these literals in a monomial of degree d is set to 0 by ρ is at most $(1-\mu\beta)^d$, which for $d \geq \frac{1}{\mu\beta}\ln\frac{1}{\delta}$ is at most δ. ■

Since the uniform distribution is 1/2-balanced, Lemma 23 establishes that any high degree monomials that appear in a PCR refutation are witnessed to evaluate to zero with high probability over $M_\mu(U_n)$, and hence in a proof Π of size $p(n)$, a union bound establishes that every monomial of degree $\Omega(\frac{1}{\mu}\log\frac{p(n)}{\delta}))$ is substituted by 0 in the restriction $\Pi|_\rho$. Thus, the degree-based algorithm of Clegg et al. [14] that searches for refutations using monomials of degree at most d (and thus runs in time $O(n^{3d})$) finds a refutation when one exists with probability $1-\delta$. The following theorem is then almost immediate:

Theorem 24 *For any $\mu \in (0,1)$, PCR is PAC-automatizable for $M_\mu(U_n)$ in quasipolynomial time.*

Proof: The algorithm takes a sample of partial examples of size $m = \frac{1}{\gamma^2} \log \frac{1}{\delta}$ and for each partial assignment $\rho^{(i)}$ uses the degree-based algorithm for $d = O(\frac{1}{\mu} \log \frac{m \cdot p(n)}{\delta})$ to check for a degree-d refutation of $\varphi|_{\rho^{(i)}}$; if more than an ϵ fraction of these refutations fail, the algorithm rejects and otherwise it accepts. It is immediate that (for constant μ) the algorithm runs in quasipolynomial time in n, $1/\gamma$, $\log 1/\epsilon$, and $1/\delta$. We thus turn to considering correctness.

If φ is $\epsilon + \gamma$ valid, then Hoeffding's inequality shows that with probability $1 - \delta$, at least an ϵ fraction of the examples drawn from U_n satisfy φ; since this satisfying assignment $x^{(i)}$ is consistent with any partial assignment $\rho^{(i)}$ in the support of $M_\mu(x^{(i)})$, $\varphi|_{\rho^{(i)}}$ is satisfiable, and hence by the soundness of PCR, no refutation exists for these partial examples, and we see that the algorithm rejects.

In the second case, we note first that (again by Hoeffding's inequality) with probability $1 - \delta/2$, every polynomial constraint of the unknown formula ψ is witnessed to evaluate to true in at least a $1 - \epsilon$-fraction of the partial examples. Let Π be the size $p(n)$ refutation of $\psi \wedge \varphi$. It then follows from Lemma 23 (cf. the above discussion) that with probability $1 - \delta/2$, for this $1 - \epsilon$ fraction of the $\rho^{(i)}$ (out of m), $\Pi|_{\rho^{(i)}}$ is a degree-d refutation of $\varphi|_{\rho^{(i)}}$ since the polynomials from ψ and monomials of Π of degree greater than d all simplify to 0. Thus, in this case, the algorithm accepts with probability $1 - \delta$, as needed. ■

3.2 Augmenting the degree-based algorithm with learning

While the uniform distribution illustrates how the *reasoning* problem may become easier in the context of a distribution, there is no *learning* problem if the distribution is known to be uniform: a given formula indicates which settings of the variables are "positive" or "negative" examples, and the entire learning question for a given formula merely concerns approximate counting of the negative examples. We now turn to considering our learning problem. (We remind the reader than an overview of our argument appeared in Section 1.3.1.)

Distributions with a bias gap turn out to be easy to work with because we only need to consider low-degree monomials: for starters, Lemma 23 guarantees that (sub)monomials of sufficient degree ($\Omega(\frac{1}{\mu\beta} \log \frac{1}{\delta})$ here) for which every variable is balanced in D (for every small subset of the rest) are witnessed with high probability. Naturally, in an affine distribution, we can make a similar claim if all of the variables of a sufficiently large monomial are each involved in some constraint that 1-implies one of their negations. More generally:

Lemma 25 *Let $x^{\vec{\alpha}}$ be a monomial of degree $2d$ for $d \geq \frac{1}{\mu} \ln \frac{1}{\delta}$ and D be a distribution with a width-d $(\beta, 1-\gamma)$ bias gap such that for every submonomial of $x^{\vec{\alpha}}$ of degree d there is some further submonomial $\ell x^{\vec{\alpha}'}$ with $x^{\vec{\alpha}'} \rightsquigarrow \neg\ell$. Then with probability $1 - \delta$, there is an unmasked literal of $x^{\vec{\alpha}}$ that is zero with probability $(1 - \gamma)$ under D.*

Proof: Let $x^{\vec{\alpha}_0}$ be the first d literals of $x^{\vec{\alpha}}$ (in some order), and suppose m is drawn from M_μ and x is drawn from D. Given $x^{\vec{\alpha}_i}$ (of degree d), some ℓ_i for the submonomial $\neg\ell_i x^{\vec{\alpha}'_i}$ is $(1-\gamma)$-implied by $x^{\vec{\alpha}'_i}$. Thus, with probability at least $1 - \gamma$, either ℓ_i is 0 in x or some other literal of $x^{\vec{\alpha}'_i}$ is 0 in x. We let ℓ'_i be that other literal of $x^{\vec{\alpha}'_i}$ in the latter case, and let it be ℓ_i in the former case. Now, we construct $\vec{\alpha}_{i+1}$ by removing ℓ'_i from $\vec{\alpha}_i$ and taking the next literal of $x^{\vec{\alpha}}$ in our ordering, and we repeat. Note that since $x^{\vec{\alpha}}$ has degree at least $2d$, we find at least d such literals ℓ'_i. With probability at least $1 - (1-\mu)^d \leq 1 - \delta$, at least one of these literals is unmasked. The first such unmasked literal is as needed. ■

The final case to consider is when there is a submonomial of degree at most $\frac{1}{\mu\beta}\ln\frac{1}{\delta}$ (noting $\beta \leq 1/2$) for which there is a literal that is implied by the rest of the monomial. In this case, as we elaborate on next, there is a small monomial that we can learn that can be used to reduce the degree of the first monomial by a resolution step. If the monomial has sufficient degree and yet is not witnessed to evaluate to zero, it must be because there are many such literals. Using these learned monomials to eliminate the corresponding literals from the original monomial enables us to find an equivalent constraint that satisfies our degree bound. We will make this precise in the proof of Theorem 28.

On the two kinds of premises, ψ_0 and ψ_1. In contrast to the definition of PAC-automatizability, Algorithm 1 supports the learning of *two* kinds of hypotheses, denoted ψ_0 and ψ_1 in the theorem statement. ψ_1 is any system of polynomials that is "witnessed satisfied" with high probability, and thus corresponds to the same kind of premise that is required by the definition of PAC-automatizability (and hence Theorem 28 implies Theorem 2); ψ_0 is the new feature. The conditions on ψ_0 essentially guarantee that any conjunction of monomials (i.e., a CNF) featuring near-perfect agreement with the data are learnable, capturing to the guarantee provided by standard, "realizable" learning of CNFs. Although in contrast to the usual learning set-up we allow some potentially noticeable counterexamples to the target hypothesis, these counterexamples are required to be very rare relative to the accuracy parameter γ given to the algorithm. Since the parity constraints in an affine distribution hold perfectly, the learnability of such a ψ_0 establishes the learnability of (a CNF representation of) the parity constraints of an affine distribution in the context of a PCR proof. We remark that our previous work [24] could not exploit such hypotheses in general, due to the fact that we did not restrict our attention to examples masked completely at random (i.e., to $M = M_\mu$).

Analysis of Algorithm 1. We first note that the learned monomials are highly valid under D and contain a constraint $\ell_1 \cdots \ell_{k-1} \cdot \neg\ell_k = 0$ whenever $\ell_1 \cdots \ell_{k-1} \rightsquigarrow \ell_k$ for $k \leq d$:

Lemma 26 *Suppose D is a distribution with a width-d $(\beta, 1 - \frac{\gamma}{4(2n+1)^d})$ bias gap. Let ψ be the conjunction of constraints $[x^{\vec{\alpha}} = 0]$ for all monomials $x^{\vec{\alpha}}$ of degree at most d that are witnessed to evaluate to 1 in at most a $\frac{\gamma\mu^d}{2(2n+1)^d}$-fraction of a sample of m_0 partial examples from $M_\mu(D)$ (for m_0 as given in Algorithm 1). Then with probability at least $1 - \delta/2$, ψ is $1-\gamma$-valid and contains all $1 - \frac{\gamma}{4(2n+1)^d}$-valid monomial constraints of degree at most d, including specifically all $\neg\ell x^{\vec{\alpha}}$ such that $x^{\vec{\alpha}} \rightsquigarrow \ell$.*

Proof: Since each monomial in ψ has degree at most d, every literal in each such monomial is simultaneously not set to $*$ by M_μ with probability at least μ^d; if a constraint is not $1 - \frac{\gamma}{(2n+1)^d}$ valid, then the monomial is witnessed to evaluate to 1 on $M_\mu(D)$ with probability at least $\frac{\mu^d\gamma}{(2n+1)^d}$. Therefore, Hoeffding's inequality implies that it is only not witnessed to evaluate to 1 sufficiently often to eliminate it from ψ with probability at most $\frac{\delta}{4(2n+1)^d}$. Since there are fewer than $(2n+1)^d$ (multilinear) monomials of degree at most d, by a union bound over these monomials, we find that with probability at least $1 - \delta/4$, the constraints corresponding to these monomials are all $1 - \frac{\gamma}{(2n+1)^d}$-valid. Again, by a union bound over the constraints, this means that their conjunction, ψ, is $1-\gamma$-valid.

function: `D-refute` (φ, d) decides whether or not there is a degree-d PCR refutation of φ.
input : System of polynomials φ, bound $p(n)$, $\epsilon, \delta, \gamma, \beta \in (0,1)$, partial examples $\rho^{(1)}, \ldots, \rho^{(m_0+m_1)}$ from $M(D)$ for $m_0 = \frac{2d(2n+1)^{2d}}{\mu^{2d}\gamma^2} \ln \frac{4(2n+1)}{\delta}$ and $m_1 = \frac{1}{2\gamma^2} \ln \frac{2}{\delta}$ where $d = \frac{1}{\mu\beta} \ln \frac{2m_1 \cdot p(n)}{\delta}$.
output : *Accept* or *Reject* (cf. Theorem 28)
begin
  Initialize ψ to an empty conjunction.
  foreach *Monomial* $x^{\vec{\alpha}}$ *of degree at most* d **do**
    FALSIFIED $\leftarrow 0$.
    for $i = 1, \ldots, m_0$ **do**
      if $x^{\vec{\alpha}} = 1$ *on* $\rho^{(i)}$ **then**
        Increment *FALSIFIED*.
      end
    end
    if *FALSIFIED* $\leq \frac{\gamma\mu^d}{2(2n+1)^d} m_0$ **then**
      $\psi \leftarrow \psi \wedge [x^{\vec{\alpha}} = 0]$.
    end
  end
  Initialize φ' to an empty conjunction.
  foreach *Constraint* $[Q(x) = 0]$ *from* φ **do**
    foreach *Monomial* $x^{\vec{\alpha}}$ *from* ψ *in lex. order* **do**
      if $x^{\vec{\alpha}} = \ell x^{\vec{\alpha}'}$ *where* $\neg\ell x^{\vec{\alpha}'}$ *appears as a submonomial in* Q **then**
        $Q \leftarrow Q$ with $\neg\ell$ deleted from every monomial containing $\neg\ell x^{\vec{\alpha}'}$.
      end
    end
    $\varphi' \leftarrow \varphi' \wedge [Q(x) = 0]$.
  end
  FAILED $\leftarrow 0$.
  for $i = m_0 + 1, \ldots, m_0 + m_1$ **do**
    if `D-refute`$((\varphi' \wedge \psi)|_{\rho^{(i)}}, 3d)$ *rejects* **then**
      Increment *FAILED*.
      if *FAILED* $> \lfloor \epsilon \cdot m_1 \rfloor$ **then**
        return Reject
      end
    end
  end
  return Accept
end

Algorithm 1: Learn+PCR

Now, when a monomial $x^{\vec{\alpha}}$ of degree at most d is $1 - \frac{\gamma}{4(2n+1)^d}$-valid (including monomials $\neg\ell x^{\vec{\alpha}}$ such that $x^{\vec{\alpha}} \rightsquigarrow \ell$), Hoeffding's inequality similarly guarantees that $x^{\vec{\alpha}}$ will be witnessed to evaluate to 1 on $M_\mu(D)$ in a $\frac{\gamma\mu^d}{2(2n+1)^d}$-fraction of m_0 partial examples with probability at most $\frac{\delta}{4(2n+1)^d}$.

Therefore, all of these monomial constraints appear in ψ except with probability $\delta/4$, and a final union bound gives that both conditions hold with probability $1 - \delta/2$. ■

The analysis of the algorithm is a generalization of the analysis of the degree-based algorithm discussed in Section 3.1. The main twist is that we may need to modify the PCR derivation by introducing learned monomials in order to obtain a low-degree derivation. We need one more lemma, establishing that we can maintain a canonical low-degree version of each monomial (and thus obtain cancellations from linear combinations of different constraints).

Lemma 27 *Let ψ be a conjunction of constraints of the form $x^{\vec{\alpha}} = 0$ of degree at most d. Let any two submonomials of degree at most d of a common monomial of a constraint be given that have been derived by successively eliminating one variable at a time by simulating resolution steps with monomials from ψ. Then there is a derivation of one from the other of degree at most $3d$.*

Proof: Let us write the constraint as $[C(x) + x^{\vec{\alpha}} = 0]$, where $x^{\vec{\alpha}}$ is the common monomial referred to in the theorem statement; let $x^{\vec{\beta}}$ and $x^{\vec{\gamma}}$ respectively be the two reduced monomials, obtained by simulating resolution steps with monomials from ψ in different orders. Recalling that we assume that the monomials are multilinear, we will show how to obtain $[C(x) + x^{\vec{\beta}\vee\vec{\gamma}} = 0]$ (where, note, $x^{\vec{\beta}\vee\vec{\gamma}}$ is another submonomial of $x^{\vec{\alpha}}$) in degree $3d$. We will do this by showing how to obtain $[C(x) + x^{\vec{\beta}\vee\vec{\gamma}} = 0]$ from either $[C(x) + x^{\vec{\beta}} = 0]$ or $[C(x) + x^{\vec{\gamma}} = 0]$. (WLOG, we will only show this for $[C(x) + x^{\vec{\beta}} = 0]$.) We can then derive $[C(x) + x^{\vec{\gamma}} = 0]$ from $[C(x) + x^{\vec{\beta}} = 0]$ and ψ in degree-$3d$ by "reversing" the derivation of $[C(x) + x^{\vec{\beta}\vee\vec{\gamma}} = 0]$ from $[C(x) + x^{\vec{\gamma}}]$ and ψ.[8]

Suppose there are k literals, $y_1, \ldots, y_k$ of $x^{\vec{\alpha}}$ that do not appear in *either* $x^{\vec{\beta}}$ or $x^{\vec{\gamma}}$; our objective will be to avoid introducing these y_i literals into our derivation of $[C(x)+x^{\vec{\beta}\vee\vec{\gamma}} = 0]$ from $[C(x)+x^{\vec{\beta}}]$, thereby maintaining low degree in the overall derivation. We will do this by constructing versions of the monomials of ψ that similarly do not contain $y_1, \ldots, y_k$ (which then facilitate the low-degree derivation). We start by considering the original derivation of $[C(x)+x^{\vec{\beta}} = 0]$ from $[C(x)+x^{\vec{\alpha}} = 0]$ using the monomial constraints from ψ. We divide the steps of the original derivation into $k+1$ *"epochs"* (consecutive subsequences), where the ith epoch (for $i = 0, \ldots, k-1$) ends with the elimination of y_{i+1} from the constraint $[C(x) + y_{i+1}x^{\vec{\alpha}_i} = 0]$, (or with $[C(x) + x^{\vec{\beta}} = 0]$ in the case of the final epoch). We note that since we only eliminate literals from these constraints,

$$\vec{\beta} \leq \vec{\alpha}_k \leq \vec{\alpha}_{k-1} \leq \cdots \leq \vec{\alpha}_1 \leq \vec{\alpha}.$$

We will show, by induction on the epochs $k-i$ ($i = 0, 1, \ldots, k$ here, i.e., considering the epochs in reverse order from the derivation of $x^{\vec{\beta}}$), that for each epoch there is a degree-$3d$ derivation of a set of monomials ψ_{k-i} such that for each monomial $x^{\vec{\eta}}$ in ψ, there is a monomial $x^{\vec{\eta}_{k-i}}$ in ψ_{k-i} such that none of $y_{(k-i)+1}, \ldots, y_k$ appear in $x^{\vec{\eta}_{k-i}}$, but containing (in addition to the rest of the literals of $x^{\vec{\eta}}$) all of the literals that appear in both $x^{\vec{\alpha}_{k-i}}$ and $x^{\vec{\beta}\vee\vec{\gamma}}$; denote this monomial by $x^{\vec{\alpha}'_{k-i}}$. We note that since ψ consists of monomials of degree at most d, as are $x^{\vec{\beta}}$ and $x^{\vec{\gamma}}$, these monomials all have degree at most $3d$. (Henceforth, for brevity, we will talk about epoch i, ψ_i, etc., taking i in the order $k, k-1, \ldots, 0$.)

For the base case, consider the monomial derived at the beginning of the final epoch ($x^{\vec{\alpha}_k}$) in which $x^{\vec{\beta}}$ is derived. We can use the multiplication rule to add all of the literals of $x^{\vec{\alpha}_k}$ to the

[8] This "reversing" of cut steps with monomials is detailed in Section 2.3.1. Briefly, in the relevant linear combination derivations using ψ, we switch the signs of the constraints derived from complementarity and the monomials from ψ.

monomials in ψ (they are contained in both $x^{\vec{\beta}\vee\vec{\gamma}}$ and $x^{\vec{\alpha}_k}$) to construct ψ_k. As we noted above, these monomials have degree at most $3d$, so this is a degree-$3d$ derivation.

Now, for the inductive step, given ψ_i, consider the monomial $\bar{y}_i x^{\vec{\eta}'_i}$ of ψ_i corresponding to the monomial $\bar{y}_i x^{\vec{\eta}_i}$ in ψ used to eliminate y_i. Since we know $\vec{\eta}_i \leq \vec{\alpha}_i$ (as we don't use the multiplication rule to weaken $y_i x^{\vec{\alpha}_i}$ in the constraint $[C(x) + y_i x^{\vec{\alpha}_i} = 0]$), all of the other monomials in ψ_i contain $x^{\vec{\eta}'_i}$ as a submonomial—it contains none of $y_{i+1}, \ldots, y_k$, and the rest of the literals from $x^{\vec{\alpha}'_i}$ (from $x^{\vec{\beta}\vee\vec{\gamma}}$) are contained in the other monomials by hypothesis. Thus, we can use $\bar{y}_i x^{\vec{\eta}'_i}$ to eliminate y_i from these monomials; to construct ψ_{i-1}, we eliminate y_i from the monomials of ψ_i in this way and then use the multiplication rule to weaken them with the rest of the literals from $x^{\vec{\gamma}\vee\vec{\beta}}$ eliminated during the ith epoch. Since the monomials of ψ_i did not contain $y_{i+1}, \ldots, y_k$ and did contain the rest of the literals of $x^{\alpha'_i}$, we note that the monomials of ψ_{i-1} (that we just constructed) further do not contain y_i and contain the rest of the literals of $x^{\alpha'_{i-1}}$, as needed. We need to check that the elimination of y_i (using the monomial $\bar{y}_i x^{\vec{\eta}'_i}$) only involved degree-$3d$ polynomials. This follows since the monomials of ψ_i had degree at most $3d$, and when they contained y_i, they were of the form $y_i x^{\vec{\xi}}$ where $\vec{\eta}'_i \leq \vec{\xi}$, where $\vec{\xi}$ had degree at most $3d-1$. We can therefore use multiplication to weaken $\vec{\eta}'_i$ to $\vec{\xi}$ while remaining within degree $3d$.

Now that we have constructed the sets $\psi_0, \psi_1, \ldots, \psi_k$ (taking $\psi_{k+1} = \psi$, weakened by $x^{\vec{\beta}}$ if we desire), we can use them to derive $[C(x) + x^{\vec{\gamma}\vee\vec{\beta}} = 0]$ from $[C'(x) + x^{\vec{\beta}} = 0]$ as follows. Let us again follow the reduction order by which $[C(x) + x^{\vec{\beta}} = 0]$ was derived from $[C(x) + x^{\vec{\alpha}} = 0]$ in reverse. Inductively, suppose we have constructed $x^{\vec{\alpha}'_{i+1}}$ (for $i = k, k-1, \ldots, 0$) taking $x^{\vec{\alpha}'_{k+1}} = x^{\vec{\beta}}$ (our base case). Now, for epoch i, we can use the monomials of ψ_{i+1} to reintroduce each of the literals $z_{i,j}$ that were eliminated during epoch i (except for y_i, the final literal): each monomial $x^{\vec{\xi}'}$ in ψ_{i+1} corresponding to the monomial $x^{\vec{\xi}}$ in ψ only contains extra literals from $x^{\vec{\alpha}'_{i+1}}$, which at the current stage is a submonomial of the constraint $[C(x) + z_{i,j} z_{i,j+1} \cdots x^{\vec{\alpha}'_{i+1}} = 0]$ we have currently constructed. Thus, we can reintroduce $z_{i,j-1}$ by reversing the cut derivation that eliminated it using the corresponding monomial in ψ_{i+1}. Much like before, we know that $z_{i,j-1} z_{i,j} \cdots x^{\vec{\alpha}'_{i+1}}$ is a submonomial of $x^{\vec{\beta}\vee\vec{\gamma}}$, and hence has degree at most $2d$; the relevant monomial of ψ_{i+1} is in turn of the form $\bar{z}_{i,j-1} x^{\vec{\xi}'}$ where $x^{\vec{\xi}'}$ is a submonomial of $z_{i,j} \cdots x^{\vec{\alpha}'_{i+1}}$, so this derivations also has degree at most $2d < 3d$. We thereby derive $[C(x) + x^{\vec{\alpha}'_i} = 0]$, completing this induction step. Since $x^{\vec{\alpha}'_0}$ is precisely $x^{\vec{\beta}\vee\vec{\gamma}}$, this completes the proof. ■

Theorem 28 *Given an input system of polynomials φ, a bound $p(n)$, $\mu, \epsilon, \gamma, \delta, \beta \in (0,1)$, and access to examples from $M_\mu(D)$ for a distribution D that has a width-$d = \frac{1}{\mu\beta} \ln \frac{2m_1 \cdot p(n)}{\delta}$ $\left(\beta, 1 - \frac{\gamma}{4n(2n+1)^d}\right)$ bias gap, (where $m_1 = \frac{1}{2\gamma^2} \ln \frac{2}{\delta}$) and given that either*

- *φ is satisfied by D with probability at least $\epsilon + 2\gamma$ or*
- *there exist systems of polynomials ψ_0 and ψ_1 such that*
  - *ψ_0 consists of $1 - \frac{\gamma}{4n(2n+1)^d}$-valid monomials and*
  - *ψ_1 is witnessed to evaluate to true with probability $1 - \epsilon + 2\gamma$ under $M_\mu(D)$*

  and there is a PCR refutation of size $p(n)$ of $\psi_0 \wedge \psi_1 \wedge \varphi$

Algorithm 1 decides which case holds with probability $1 - \delta$, and runs in time $n^{O(\frac{1}{\mu\beta} \log \frac{p(n)}{\gamma\delta})}$.

Proof: We note that the running time bound is essentially immediate from the description of the algorithm, as the degree-based algorithm, for degree d', runs in time $O(n^{3d'})$. We therefore turn to

considering the correctness of the algorithm. The first case is similarly simple: by Proposition 7, if φ is $\epsilon + 2\gamma$-valid under D, then since the learned constraint ψ is $1-\gamma$-valid with probability at least $1-\delta/2$ by Lemma 26, then $\varphi \wedge \psi$ is $\epsilon+\gamma$-valid—meaning that with probability at least $\epsilon+\gamma$, a masked example ρ is drawn from $M_\mu(D)$ for which $(\varphi \wedge \psi)|_\rho$ is satisfiable. It then follows by Hoeffding's inequality that with probability greater than $1-\delta/2$, for at least an ϵ-fraction of the actual m_1 examples, $(\varphi \wedge \psi)|_\rho$ will be satisfiable, and therefore by the soundness of PCR, at least an ϵ-fraction of the iterations must fail to find a refutation (of the consequence $(\varphi' \wedge \psi)|_\rho$), leading the algorithm to reject as needed with probability at least $1-\delta$.

It only remains to establish that the algorithm accepts with probability at least $1-\delta$ when there exists a size-$p(n)$ refutation from some almost perfectly valid ψ_0 and some ψ_1 that is witnessed satisfied with probability $1-\epsilon+2\gamma$ under $M_\mu(D)$. We assume (WLOG) that every constraint of ψ_1 appears in the refutation. We now begin by identifying two subsets of monomials, Π_1 and Π_0, of the size-$p(n)$ proof Π where Π_1 consists of monomials which either

1. contain a submonomial $x^{\vec{\alpha}}$ of degree $d/2$ (for, recall, $d = \frac{1}{\mu\beta}\ln\frac{2m_1p(n)}{\delta}$) for which every further submonomial $\ell x^{\vec{\alpha}'}$ of $x^{\vec{\alpha}}$ has ℓ balanced for $x^{\vec{\alpha}'}$ or
2. contain a submonomial of degree d for which every (sub-)submonomial $x^{\vec{\alpha}}$ of degree $d/2$ has a further submonomial $\ell x^{\vec{\alpha}'}$ for $x^{\vec{\alpha}'}$ and ℓ satisfying $x^{\vec{\alpha}'} \rightsquigarrow \neg\ell$,

and Π_0 consists of the rest of the monomials.

We first note that the monomials in Π_1 (and constraints in ψ_1) are simultaneously witnessed to simplify to 0 in most of the partial examples: we first note that by Hoeffding's inequality, with probability at least $1-\delta/2p(n)$, each constraint in ψ_1 is witnessed to evaluate to 0 in at least $(1-\epsilon+\gamma)m_1$ partial examples; likewise, for the monomials of Π_1 which have submonomials for which every degree-$d/2$ submonomial has an implied literal, by a union bound over all $2n$ literals, the probability that any of them is both a $(1-\frac{\gamma}{4n(2n+1)^d})$-implied literal indicated by Lemma 25 and 1 in any example is at most $\gamma/4$. (Note that with probability at least $1-\delta/(2p(n)m_1)$, every such monomial has some such unmasked literal in each example.) Therefore, by Hoeffding's inequality, these monomials are all witnessed to evaluate to 0 in at least $(1-\gamma)m_1$ of the examples with probability at least $1-\delta/(2p(n)m_1)$. Similarly, by Lemma 23, the monomials with β-balanced submonomials of degree at least d are witnessed to evaluate to 0 in each partial example with probability at least $1-\delta/(2p(n)m_1)$. Hence by a union bound over the monomials and examples, every monomial in Π_1 (and constraint in ψ_1) is ultimately (simultaneously) witnessed to evaluate to 0 in at least $(1-\epsilon)m_1$ of the examples considered in the main loop of the algorithm with probability $1-\delta/2$.

We now note that for every monomial in Π_0, every submonomial of degree greater than d must have a (sub)submonomial $x^{\vec{\alpha}}$ of degree $d/2$ with some literal ℓ such that for a further submonomial $\ell x^{\vec{\alpha}'}$, $x^{\vec{\alpha}'} \rightsquigarrow \ell$—every such $x^{\vec{\alpha}}$ must have a submonomial with a literal ℓ that is not balanced (or else the monomial would be in Π_1 by the first condition) and the further submonomial cannot have $x^{\vec{\alpha}'} \rightsquigarrow \neg\ell$ for every degree $d/2$ submonomial $x^{\vec{\alpha}}$ (or else it would be in Π_1 by the second condition). Therefore, by Lemma 26, the constraint $[\neg\ell x^{\vec{\alpha}'} = 0]$ is added to ψ by the algorithm. Moreover, since the constraints of ψ_0 are $1-\frac{\gamma}{4n(2n+1)^d}$-valid, and would reach degree d after eliminating fewer than n such literals, for the monomials of ψ_0 in Π_0 there is a conjunction of degree-$\leq d$ monomials ψ_0' consisting of submonomials of every monomial of ψ_0 appearing in Π_0 that (by Proposition 7) are $1-\frac{\gamma}{4(2n+1)^d}$-valid constraints, and hence also added to ψ by Lemma 26.

Similarly, for every monomial $x^{\vec{\alpha}}$ of an input constraint $[Q(x) = 0]$ (from φ) in Π_0, we can always derive a submonomial of degree at most d using these monomials—as long as our submonomial of $x^{\vec{\alpha}}$ has degree greater than d, some submonomial of degree $d/2$ must contain a submonomial $\ell x^{\vec{\alpha}'}$ for which a constraint $[\neg \ell x^{\vec{\alpha}'} = 0]$ is in ψ, which we can use to reduce its size further, so the monomials of φ' corresponding to monomials of φ in Π_0 have degree at most d.

To conclude, we observe that since all of the monomials of Π_1 (and constraints in ψ_1) are witnessed to evaluate to 0 in at least $(1-\epsilon)m_1$ of the partial examples considered by the algorithm in the main loop with probability $1 - \delta/2$, for each such partial example ρ, by Proposition 20 $\Pi|_\rho$ is a PCR refutation of (restrictions of) the input and constraints from ψ_0, consisting only of (restrictions of) constraints over monomials from Π_0. Now, we know that there is a set of degree-$\leq d$ monomial constraints ψ_0' consisting of submonomials of every monomial of ψ_0 in which literals have been eliminated in lex. order, that is included in ψ with probability at least $1 - \delta/2$, and for each input constraint in Π_0 there is a constraint supported on monomials of degree less than d that are submonomials of the original constaint that can be derived by the algorithm.

By induction on the steps of $\Pi|_\rho$, we will find that by carrying out the steps of the proof with these constraints over submonomials, we can always derive some constaint $[C_i'(x) = 0]$ over submonomials of the monomials from Π_0 of each ith constraint in $\Pi|_\rho$ (where the literals have been eliminated using ψ in lex. order), which hence have some further submonomials of degree at most d which can be derived in degree $3d$: If the next step of $\Pi|_\rho$ is obtained by applying the linear combination rule to $[C_i(x) = 0]$ and $[C_j(x) = 0]$, we can apply the same linear combination to the derived degree-d subconstraints $[C_i'(x) = 0]$ and $[C_j'(x) = 0]$ to obtain a subconstraint of the next step of degree at most d. Crucially, since we assume that the monomials have been pruned of literals using ψ in the same order, any cancellations that would occur among monomials in $C_i(x)$ and $C_j(x)$ occur among the corresponding submonomials in $C_i'(x)$ and $C_j'(x)$, thus indeed yielding a constraint of degree-d of the claimed form. If, on the other hand, the next step is obtained by applying the multiplication rule to $C_i(x)$, then we can derive the resulting constraint $C'(x)|_\rho$ (in which literals have been eliminated in lex. order) in degree at most $3d$ by Lemma 27, since these monomials of $x \cdot C_i(x)$ are in Π_0 and so the resulting monomials of $C'(x)|_\rho$ have degree at most d. This completes the induction step. Since, finally, the only submonomial of 1 (derived in the final step) is 1 itself, this ultimately yields a degree-$3d$ refutation of $\Pi|_\rho$. Since the algorithm therefore refutes φ at least $(1-\epsilon)m_1$ out of the m_1 examples with probability at least $1 - \delta$, it therefore accepts with probability at least $1 - \delta$ in this case, as needed. ■

4 Cryptographic non-automatizability

4.1 The generalized Diffie-Hellman assumption and propositional encoding

The Diffie-Hellman key exchange scheme was one of the contributions of the seminal work by Diffie and Hellman [19]. Roughly speaking, over the group $\mathbb{Z}_p^*$ (for a possibly composite p) and quadratic residue g, the parties come to share a secret key $\sigma = g^{ab} \bmod p$ by publicly exchanging $g^a \bmod p$ and $g^b \bmod p$ for privately chosen secrets a and b. It is known that computing $g^{ab} \bmod p$ from the public information (g, g^a, g^b, and p) when p is the product of integers equal to 3 modulo 4 ("Blum integers") is as hard as factoring such integers [10, 32, 37], which is commonly presumed to be hard.

Bonet, Pitassi, and Raz [13] exhibited TC_0 formulas essentially asserting that either any desired ith bit of the secret g^{ab} is equal to 0 or is equal to 1, such that moreover the (contradictory)

conjunction of these formulas for any i has a polynomial-size TC_0-Frege refutation. We will revisit their argument in more detail in the next section, but first we will give an overview of the formulas they use as these are somewhat nontrivial. We will review the variables used in particular, as the role played by these variables is crucial to our reductions.

Precisely, fixing an ith bit (we will follow them and only consider the least significant bit here) they give two formulas A_0 and A_1 respectively asserting that the given bit of g^{ab} is 0 or 1. A_0 and A_1 will actually use different sets of variables for the secret values a and b; in A_1 we will denote these values by variables c and d, respectively. The rest of the variables are shared between the two formulas.

These common variables include binary encodings of the n-bit integers p and g, as well as encodings of $g^{2^i} \bmod p$ for each $i \leq 2n$, and encodings of $p_i = i \cdot p$ for $i \leq n$. They also include binary encodings of $x = g^a \bmod p$, $y = g^b \bmod p$, and likewise $x^{2^i} \bmod p$ and $y^{2^i} \bmod p$ for $i \leq 2n$. In order to compute quotients and remainders, they further include values k_i and r_i for $i \leq n$ such that $0 \leq r_i \leq p$ and $2^i = p \cdot k_i + r_i$. We remark that the values of all of these variables can be computed from the public values in polynomial time.

The formulas perform the exponentiation of g by the secret values a and b (resp., c and d) by using an iterated product that is performed by an iterated sum in the exponent in the chinese remainder representation; conversion to and from the chinese remainder representation is performed in essence by iterated sums in modular arithmetic, with the isomorphisms between an additive representation (as $\mathbb{Z}_{q_i-1}$) and multiplicative representation of $\mathbb{Z}^*_{q_i}$ for the primes q_i used in the chinese remainder representation performed by table lookups. Naturally, these primes have size $O(\log n)$ so that these tables have polynomial size. Moreover, the primes (and generators) used for the representation only depend on the length of the integers n (not p or any of the other public or private values for the DH scheme). Thus, even a brute-force search for appropriate values can be performed in time polynomial in n, and these will be "hard-wired" into the formulas.

Given the iterated product, the formula A_0 is now given by the conjunction of formulas asserting for $i \leq n$, $2^i = p \cdot k_i + r_i$, $1 \leq r_i < p$, and $p_i = i \cdot p$ (to facilitate modular arithmetic); $\prod_i g^{2^i a_i} \bmod p = x$ and $\prod_i g^{2^i b_i} \bmod p = y$ (so a and b correspond to the private values reflected in the public values x and y); for every $j \leq n$, $\prod_i g^{2^{i+j} a_i} \bmod p = x^{2^j} \bmod p$ and $\prod_i g^{2^{i+j} b_i} \bmod p = y^{2^j} \bmod p$ (in support of the next step, where x^{2^j} and y^{2^j} are similarly publicly computable from x and y); and finally, $\prod_{i,j} g^{2^{i+j} a_i b_j} \bmod p$ ($= g^{ab} \bmod p$, the shared secret) is even. A_1 is similar, replacing a and b by c and d, and finally concluding that $\prod_{i,j} g^{2^{i+j} c_i d_i} \bmod p$ ($= g^{cd} \bmod p$) is odd.

4.2 Non-automatizability in PAC-Semantics

We begin by observing that essentially *all* of the existing cryptographic non-automatizability results actually establish non-automatizability under (general, "distribution-free") PAC-Semantics. Concretely, let us recall the argument used by Bonet, Pitassi, and Raz [13] to establish that TC_0-Frege is not automatizable given the security of the Diffie-Hellman protocol (and hence, given that factoring is intractable). The main step in their argument can be summarized as follows:

Theorem 29 (Bonet, Pitassi, Raz [13]) *The TC_0 formula $A^i_0 \wedge A^i_1$ described in Section 4.1 (where A^1_0 defines "$g^a \bmod p = x$, $g^b \bmod p = y$, and the ith bit of $g^{ab} \bmod p$ is 0" and A^i_1 defines "$g^c \bmod p = x$, $g^d \bmod p = y$ and the ith bit of $g^{cd} \bmod p$ is 1") has a polynomial-size TC_0-Frege refutation. There is also an algorithm that given n and $i \leq n$, runs in polynomial-time and produces A^i_0 and A^i_1 as output.*

In particular, by plugging in the known public values for the variables in these TC_0-formulas, the formula asserting e.g. that the ith bit is 0 when it is actually 1 has a polynomial-size TC_0-Frege refutation. Thus, the ability to decide whether it is A_0^i or A_1^i that has a polynomial-size TC_0-Frege refutation allows us to recover the ith bit of $g^{ab} \bmod p$, completely breaking the security of DH. Of course, a and b are (essentially) determined by g, p, $g^a \bmod p$ and $g^b \bmod p$. This trivially establishes that PAC-automatizability of TC_0-Frege for arbitrary distributions and masking processes is as hard as breaking DH:

Theorem 30 *Suppose TC_0-Frege is PAC-automatizable in time $T(N, 1/\epsilon, 1/\gamma, 1/\delta)$ for all distributions and masking processes on formulas of size N for $T(N, 2, 3, 1/\delta) \geq \Omega(N^k)$ for some sufficiently large constant k. Then for some polynomial P, there is an algorithm running in time $O(n \cdot T(P(n), 2, 3, n/\delta))$ that recovers $g^{ab} \bmod p$ with probability $1 - \delta$ from any n-bit p, generator g of $\mathbb{Z}_p^*$, $g^a \bmod p$, and $g^b \bmod p$ where a and b are arbitrary.*

Proof: Let such p, g, $g^a \bmod p$, and $g^b \bmod p$ be given, and let a and b be their respective natural representatives. Let D be a distribution that puts *all* of its mass on this single assignment, and let M be the masking process that hides only the values of the (vectors of) variables encoding the binary representations of a, b, and their copies c and d. Note that samples from $M(D)$ correspond to a vector encoding the public values and masking the secret values. Now, for any ith bit of the secret $g^{ab} \bmod p$, Theorem 29 establishes that the formulas A_0^i and A_1^i can be generated in time $O(n^k)$ for some k and have a TC_0-Frege refutation (without any additional premises) of size $N = P(n)$ for some polynomial P. In particular, plugging in the public values, it follows that the formula A_b^i (where the ith bit is actually b) is satisfied by the secret values—i.e., is 1-valid where since $1 - \gamma > 1/2$, the algorithm with $\epsilon = 1/2$ must accept such a query. Moreover, the other formula must have a refutation of size $P(n)$; that is, this other formula is actually 0-valid, and so, since $\gamma < 1/2$, the algorithm must reject it when $\epsilon = 1/2$. Thus, by running our algorithm on A_0^i for $i = 1, \ldots, n$, we recover $g^{ab} \bmod p$ in time $O(n \cdot T(P(n), 2, 3, n/\delta))$ as claimed. ■

We will strengthen this basic result in two stages. First, we will show that even if the distribution is restricted to an affine distribution and the masking process is M_μ for some constant μ, TC_0-Frege still cannot be PAC-automatized. We will then show that this result can be carried over to AC_0-Frege given a stronger assumption about the hardness of factoring, as in the work of Bonet et al. [11]. (An overview of the argument appeared previously in Section 1.3.2.)

For the first stage, we will simply substitute a parity of a vector of $k(n)$ new variables for each secret Boolean variable, and let $D_n^{\oplus k(n)}$ be the distribution in which these parities are constrained to take the same values as the underlying secrets; then for any constant μ, we can simulate access to the result of $M_\mu(D_n^{\oplus k(n)})$ with negligibly small failure probability. This then provides a "leakage resilient" encoding of these secret values that is strong enough to withstand the relatively benign leakage provided by M_μ. (This use of the parity encoding to obtain leakage-resilience goes back to work by Ishai, Sahai, and Wagner [23].)

Lemma 31 *Let D_n and M be point distributions over n variables and masks on n variables, respectively. Let θ be a substitution that takes each variable x such that $M(x) = *$ to a parity of $k(n)$ new variables, $x_1, \ldots, x_{k(n)}$, and leaves all other variables fixed. Let $D^{\oplus k(n)}$ be the distribution over this new set of variables such that the variables left fixed by θ take the same value as in D_n and the new variables are uniformly distributed over values satisfying $x_1 \oplus \cdots \oplus x_{k(n)} = b$ where x took value b in D_n. Then for any p-valid formula φ under D_n, $\theta\varphi$ is also p-valid under $D^{\oplus k(n)}$.*

Moreover, there is a distribution that can be sampled in linear time given an example from $M(D_n)$ that is $1 - n\mu^{k(n)}$-statistically close to $M_\mu(D^{\oplus k(n)})$.

Proof: Since every assignment in the support of $D^{\oplus k(n)}$ satisfies $x_1 \oplus \cdots \oplus x_{k(n)} = b$ where the original variable x takes value b with probability 1 in D_n, it is immediate that $\theta\varphi$ takes the same (fixed) value under every assignment drawn from $D^{\oplus k(n)}$ as φ took under the sole assignment in the support of D_n. We can sample from $M_\mu(D^{\oplus k(n)})$ as follows: we construct an assignment to the new set of variables by first taking the known values from our partial assignment from $M(D_n)$, and then filling in the unknown (new) variables by tossing an unbiased coin for each variable. We denote this distribution by $\tilde{D}$. We then sample M_μ in the natural way, and output the result.

To see that $M_\mu(\tilde{D})$ is statistically close to $M_\mu(D^{\oplus k(n)})$, we merely note that for each masked x under M, whenever at least one of the new variables x_i is masked by M_μ, then the distribution induced by $M_\mu(D^{\oplus k(n)})$ is uniform over the unmasked x_i, i.e., identical to $M_\mu(\tilde{D})$. Therefore, as long as every block of masked variables has at least one masked variable, which happens with probability $1 - \mu^{k(n)}$, the distributions are identical. A union bound over the (at most n) blocks gives the desired bound. ■

Repeating the argument of Theorem 30, the desired strengthening is now almost immediate:

Theorem 32 *Suppose TC_0-Frege is PAC-automatizable for affine distributions under M_μ for constant μ in time $T(N, 1/\epsilon, 1/\gamma, 1/\delta)$ for $T(N, 2, 3, 1/\delta) \geq \Omega(N^c)$ and $T(N, 2, 3, 1/\delta) < 2^{o((N \log \frac{1}{\delta})^{1/c})}$ for sufficiently large c. Then for some polynomial P, there is an algorithm running in time $\tilde{O}(nT(P(n) \cdot \log \frac{1}{\delta}, 2, 3, \frac{n}{\delta}))$ that recovers $g^{ab} \bmod p$ with probability $1 - \delta$ from any n-bit p, generator g of $\mathbb{Z}_p^*$, $g^a \bmod p$, and $g^b \bmod p$, where a and b are arbitrary.*

Proof: We first note that by Proposition 22, if φ has a TC_0-Frege refutation of size $P(n)$, then for any $k(n)$, the substitution instance $\theta\varphi$ taking some variables to parities of $k(n)$ new variables has a TC_0-Frege refutation of the same length in which each formula has size greater by at most a factor of $k(n)$. If we take $k(n) = n \log \frac{n}{\delta}$ and c such that $P(n) < O(n^{c-2})$, then $k(n) \geq \log \frac{nT(k(n)P(n),2,3,n/\delta)}{\delta} \log \frac{1}{\mu}$ for sufficiently large n since $\log T(N, 2, 3, n/\delta) < o((N \log \frac{n}{\delta})^{1/c}) = o(n \log \frac{n}{\delta})$ by assumption. We may then apply Lemma 31 to simulate samples from $M_\mu(D^{\oplus k(n)})$. Since the algorithm takes at most $T(N, 2, 3, n/\delta)$ examples, and each example is good with probability at least $1 - \frac{\delta}{2nT(N,2,3,n/\delta)}$, the overall probability that the algorithm fails is greater by at most $\delta/2n$, for an overall probability of δ/n. We use this algorithm to recover each bit of $g^{ab} \bmod p$ as before with a total failure probability of at most $1 - \delta$ as needed. ■

Theorem 32 gives a range of non-automatizability bounds for a range of assumptions on the hardness of factoring: for example, we can conclude that TC_0-Frege cannot be PAC-automatized in (quasi-)polynomial time unless integer factoring has a (quasi-)polynomial-time algorithm, even for the restricted case of affine distributions masked completely at random (by M_μ). This stands in contrast to PCR, which we saw is quasipolynomial time PAC-automatizable in this restricted case (Theorem 28). We can also obtain 2^{n^ϵ}-hardness of TC_0-Frege given 2^{n^ϵ}-hardness for integer factoring. Under this latter (strong) assumption, we can obtain a weaker conclusion for AC_0-Frege, following Bonet et al. [11, Theorem 7.1]:

Theorem 33 *Suppose AC_0-Frege is PAC-automatizable for affine distributions under M_μ for constant μ and ϵ in time $2^{\log^c(N/\delta)}$ for some constant c. Then for all $\eta > 0$, there is an algorithm for integer factoring running in time 2^{n^η}.*

Proof: Theorem 29 establishes that the DH formula on m bits generally has a refutation of size polynomial in m; suppose we take $m = \log^{(c+1)/\eta} n$. Then $A_0^i \wedge A_1^i$ has a refutation of size polynomial in m. We let θ be the substitution taking each of the $O(m)$ (secret) variables to a parity of size $k(n) = 2C \log \frac{1}{\mu} \log^c(n/\delta)$ for some $\delta = m^{-s}/2$ (C and s given below). Then Proposition 22 implies that $\theta(A_0^i \wedge A_1^i)$ (still) has a TC_0-Frege refutation of size polynomial in m; Theorem 21 now implies that this TC_0-Frege refutation has an efficiently computable AC_0-Frege translation of size $O(n^K)$ where K depends only on c and η.

Noting that the asymptotic size of these proofs is independent of C, we can fix $C = (K^c + 1)$ so that the assumed algorithm for PAC-automatizing AC_0-Frege under M_μ and affine distributions decides such instances in time $2^{C \log^c n}$. Lemma 31 enables us to simulate its examples from $M_\mu(D^{\oplus k(n)})$ with failure probability $O(m\mu^{k(n)}) = O(\frac{1}{2^{2C \log^c(n/\delta))}}) < \delta \frac{1}{m 2^{C \log^c n}}$ (for sufficiently large n), and hence we can simulate $m2^{C \log^c n}$ samples from $M_\mu(D^{\oplus k(n)})$ with overall failure probability δ. Thus, following the approach of Theorem 30, we can recover $g^{ab} \bmod p$ from g, p, $g^a \bmod p$, and $g^b \bmod p$ for m-bit values by making m queries. Following the reduction of Biham, Boneh, and Reingold [10], this enables us to factor (Blum) integers with constant success probability overall in polynomial in m repetitions. Suppose this reduction makes $O(m^r)$ queries; then we take $s = r$, and find overall that we solve instances of size $m = (\log^{c+1} n)^{1/\eta}$ in time $O(m^r 2^{C \log^c n}) < 2^{\log^{c+1} n} = 2^{m^\eta}$ (for m sufficiently large). ■

4.2.1 Hard polynomial queries under $M_\mu(D)$ for affine D

We close by observing that the same techniques that yielded Theorem 33 also enable us to easily construct queries in the same algebraic representation as considered by PCR that are NP-hard under $M_\mu(D)$ for affine D. Thus, the promise in PAC-automatizability that the yes-instances have small PCR proofs is essential unless NP has randomized quasipolynomial-time algorithms. In other words, the representational restriction of the queries alone seems to be insufficient to permit the design of good algorithms—the proof complexity of the query matters.

Theorem 34 *Unless* NP *has quasipolynomial time randomized algorithms, there is no* $2^{O(\log^c(n/\gamma)}$ *time algorithm that for all affine distributions* D *and constant* μ *and* δ*, given an input query polynomial* $Q(x)$*, and* $\gamma > 0$*, distinguishes* $\Pr_D[Q(x) = 0] < \gamma$ *from* $\Pr_D[Q(x) = 0] > 1 - \gamma$ *with probability* $1 - \delta$ *using access to examples from* $M_\mu(D)$*.*

Proof: We show how such an algorithm can solve satisfiability for CNFs. Suppose it runs in time $2^{O(\log^c(n/\gamma))}$. We first substitute size-$k(n) = \log \frac{1}{\mu}(\log \frac{n}{\delta} + C \log^{2c+1} \frac{n}{\gamma})$ parities for each of the variables of a given input CNF φ (C given below), and apply the transformation of Bonet et al. (Theorem 21) to obtain an AC_0 circuit encoding of size $O(n^K)$ where K depends only on c. In particular, if φ was satisfiable, we consider the affine distribution $D = D^{\oplus k(n)}$ obtained from the uniform parity encoding of the point distribution on the satisfying assignment given by Lemma 31; if φ was not satisfiable, we let D be the uniform distribution. In either case, note that Lemma 31 guarantees that we can sample from $M_\mu(D)$ $2^{C \log^{2c+1}(n/\gamma)}$ times with constant failure probability overall.

Now, we can in turn use a random polynomial encoding of the AC_0 circuit (due to Aspnes et al. [3] for characteristic zero) to obtain an expected additive-$\gamma\delta$ approximate encoding of the circuit under D of degree $O((\log \frac{1}{\gamma} + \log \frac{1}{\delta} + \log n) \log n)$ and hence size $2^{O((\log \frac{1}{\gamma} + \log n) \log^2 n)}$. (Recall δ is assumed constant.) By Markov's inequality, with probability $1 - \delta$ over the choice of polynomial

encodings, the polynomial agrees with the AC_0 circuit with probability $1-\gamma$ under D. Thus, the given algorithm for answering polynomial queries runs in time $2^{C\log^{2c+1}(n/\gamma)}$ (we fix C here) on such instances and distinguishes unsatisfiable φ (which are nonzero at most a γ-fraction of the time under D) from satisfiable φ which are nonzero at least a $1-\gamma$ fraction under their D with probability $1-3\delta$. ■

5 Directions for future research

Several problems present themselves as natural directions for future work. The most pressing of these is, *can the restriction to distributions with a bias gap be lifted?* That is, how can we efficiently reason about "medium-strength" biases? Although the ultimate objective of such work would be to strengthen these results to the distribution-free PAC setting, any work that handled a class of distributions that exhibited such biases would also be of interest. A similar direction would be to obtain results for a more general class of masking processes; although it seems that our results generalize to masking distributions that simultaneously reveal any width-w set of literals with non-negligible probability (for $w = \Omega(\log n)$) such as w-wise independent distributions (Wigderson and Yehudayoff [41] make a similar observation about their algorithm for population recovery, which uses the same partial-information setup), it would be desirable to find other, perhaps weaker properties that would also permit relatively efficient algorithms.

Of course, the results of this work beg the question so far as the classical (quasi)automatizability of PCR (and resolution) is concerned. Although there are families of counterexamples [12, 4] showing that a purely width (and/or treelike) based approach to finding small resolution proofs such as pursued by Ben-Sasson and Wigderson [9] cannot beat the current best-known bound of $n^{O(\sqrt{n\log n})}$ (for both resolution and PCR), it does not rule out other approaches. Since our algorithm and analysis essentially establish that every PCR proof over distributions with a bias gap has a low-degree approximate version using the learned monomials, it seems significant for our algorithm that the learned formula ψ may not have a small-degree derivation. Unfortunately, it is not clear how one might hope to exploit this in the absence of a distribution. Still, if *any* algorithm could find PCR (resp. resolution) derivations in quasipolynomial time, then using the results of our previous work [24], this would also immediately resolve both of the questions suggested in the previous paragraph.

Along these lines in the other direction, we note that the nonautomatizability results for resolution and PCR (first obtained from the work of Alekhnovich and Razborov [2] and Galesi and Lauria [22], respectively) merely show that such algorithms cannot be too sensitive to the length of the proof. This is too weak to obtain non-PAC-automatizability as our no-instances merely need to detect false queries, not queries requiring long proofs. It seems reasonable to conjecture that resolution and PCR are nonautomatizable in this stronger sense (and that therefore *some* restriction on the masking process is needed at a minimum) and it would be interesting if this could be shown.

The other natural direction in which one might hope to strengthen our results involves extending them to proof systems incomparable with PCR, such as cutting planes or k-DNF resolution. We already observed in previous work [24] that there are natural fragments of these proof systems (already well studied in the case of k-RES) that are PAC-automatizable. The question would be whether, as with degree-restricted PCR, we could use these algorithms as a starting point to obtain algorithms for the unrestricted proof system in the context of reasoning about a distribution.

Acknowledgements

The author would like to thank Paul Beame, Eli Ben-Sasson, and Leslie Valiant for comments and conversations that helped shape this work.

References

[1] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002.

[2] Michael Alekhnovich and Alexander A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM J. Comput.*, 38(4):1347–1363, 2008.

[3] James Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):135–148, 1994.

[4] Albert Atserias and María Luisa Bonet. On the automatizability of resolution and related propositional proof systems. *Inf. Comp.*, 189:182–201, 2004.

[5] Lucia Batman, Russell Impagliazzo, Cody Murray, and Ramamohan Paturi. Finding heavy hitters from lossy or noisy data. In P. Raghavendra et al., editor, *Proc. APPROX/RANDOM 2013*, number 8096 in LNCS, pages 347–362. Springer, 2013.

[6] Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *JAIR*, 22:319–351, 2004.

[7] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proc. 37th FOCS*, pages 274–282, 1996.

[8] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. *JCSS*, 56(3):277–298, 1998.

[9] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow – resolution made simple. *J. ACM*, 48(2):149–169, 2001.

[10] Eli Biham, Dan Boneh, and Omer Reingold. Breaking generalized Diffie-Hellman modulo a composite is no easier than factoring. *Inform. Process. Lett.*, 70:83–87, 1999.

[11] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldá, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth Frege proofs. *Comput. Complex.*, 13:47–68, 2004.

[12] María Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, 2001.

[13] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automization for Frege proof systems. *SIAM J. Comput.*, 29(6):1939–1967, 2000.

[14] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Gröbner basis algorithm to find proofs of unsatisfiability. In *Proc. 28th STOC*, pages 174–183, 1996.

[15] Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. To appear in 46th STOC, 2014.

[16] Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.

[17] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *JACM*, 7(3):201–215, 1960.

[18] Scott E. Decatur and Rosario Gennaro. On learning from noisy and incomplete examples. In *Proc. 8th COLT*, pages 353–360, 1995.

[19] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, 22:423–439, 1976.

[20] Zeev Dvir, Anup Rao, Avi Wigderson, and Amir Yehudayoff. Restriction access. In *Proc. 3rd ITCS*, 2012.

[21] Richard E. Fikes and Nils J. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3–4):189–208, 1971.

[22] Nicola Galesi and Massimo Lauria. On the automatizability of polynomial calculus. *Theory Comput. Sys.*, 47(2):491–506, 2010.

[23] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In D. Boneh, editor, *Proc. CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Berlin, 2003.

[24] Brendan Juba. Implicit learning of common sense for reasoning. In *Proc. 23rd IJCAI*, pages 939–946, 2013. Preliminary version: *Learning implicitly in reasoning in PAC-Semantics*, arXiv:1209.0056v1 [cs.AI].

[25] Brendan Juba. Integrated common sense learning and planning in POMDPs. Submitted, 2014.

[26] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Towards efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[27] Roni Khardon and Dan Roth. Learning to reason. *J. ACM*, 44(5):697–725, 1997.

[28] Roni Khardon and Dan Roth. Learning to reason with a restricted view. *Machine Learning*, 35:95–116, 1999.

[29] Jan Krajíček and Pavel Pudlák. Some consequences of cryptographical conjectures for S_2^1 and EF. In D. Leivant, editor, *Logic and Computational Complexity*, volume 960 of *LNCS*, pages 210–220. Springer, Berlin, 1995.

[30] Alexis Maciel and Toniann Pitassi. Towards lower bounds for bounded-depth Frege proofs with modular connectives. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetics*, number 39 in DIMACS Ser. Discrete Math. Theoret. Comput. Sci., pages 195–227. AMS, 1998.

[31] John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, 1969.

[32] Kevin McCurley. A key distribution system equivalent to factoring. *J. Cryptology*, 1:95–105, 1988.

[33] Loizos Michael. Partial observability and learnability. *Artificial Intelligence*, 174(11):639–669, 2010.

[34] Ankur Moitra and Michael Saks. A polynomial time algorithm for lossy population recovery. In *Proc. 54th FOCS*, pages 110–116, 2013.

[35] Christos Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *JCSS*, 61(2):217–235, 2000.

[36] Alexander A. Razborov. Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. *Math. Notes Acad. of Sci. USSR*, 41(4):333–338, 1987.

[37] Z. Shmuely. Composite Diffie-Hellman public-key generating systems are hard to break. Technical Report 356, Technion, Haifa, 1985.

[38] G. S. Tseitin. On the complexity of derivation in propositional calculus. In A. O. Slisenko, editor, *Studies in constructive mathematics and mathematical logic, part 2*, pages 115–125. Consultants Bureau, New York, 1970.

[39] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 18(11):1134–1142, 1984.

[40] Leslie G. Valiant. Robust logics. *Artificial Intelligence*, 117:231–253, 2000.

[41] Avi Wigderson and Amir Yehudayoff. Population recovery and partial identification. In *Proc. 53rd FOCS*, 2012.

A Axioms and rules of inference for bounded-depth Frege

The initial sequents (axioms) are the following

- $0 \to$ and $\to 1$
- The empty connective sequents $\to \wedge()$, $\vee() \to$, $\oplus_1() \to$, $\to \oplus_0()$, $\mathrm{Th}_k() \to$ for $k \geq 1$
- $A \to A$ for any formula A
- $\to \mathrm{Th}_0(A_1, \ldots, A_k)$ for any $A_1, \ldots, A_k$ and $k \geq 0$.

The rules of inference are the following

- (Weakening) From $\Gamma \to \Delta$, infer $\Gamma, A \to \Delta$ or $\Gamma \to \Delta, A$ for any formula A.
- (Contraction) From $\Gamma, A, A \to \Delta$ infer $\Gamma, A \to \Delta$; from $\Gamma \to \Delta, A, A$ infer $\Gamma \to \Delta, A$.
- (Permutation) From $A_1, \ldots, A_s \to B_1, \ldots, B_t$, infer $A_{\pi(1)}, \ldots, A_{\pi(s)} \to B_{\pi'(1)}, \ldots, B_{\pi'(t)}$ for any permutations π on $[s]$ and π' on $[t]$.
- (Cut) From $\Gamma, A \to \Delta$ and $\Gamma' \to A, \Delta'$ infer $\Gamma, \Gamma' \to \Delta, \Delta'$.
- (Negation-left) From $\Gamma \to A, \Delta$ infer $\neg A, \Gamma \to \Delta$.
- (Negation-right) From $A, \Gamma \to \Delta$ infer $\Gamma \to \neg A, \Delta$.

- (And-left) From $A_1, \wedge(A_2, \ldots, A_r), \Gamma \to \Delta$, infer $\wedge(A_1, \ldots, A_r), \Gamma \to \Delta$.
- (And-right) From $\Gamma \to A_1, \Delta$ and $\Gamma \to \wedge(A_2, \ldots, A_r), \Delta$, infer $\Gamma \to \wedge(A_1, \ldots, A_r), \Delta$.
- (Or-left) From $A_1, \Gamma \to \Delta$ and $\vee(A_2, \ldots, A_r), \Gamma \to \Delta$, infer $\vee(A_1, \ldots, A_r), \Gamma \to \Delta$.
- (Or-right) From $\Gamma \to A_1, \vee(A_2, \ldots, A_r), \Delta$, infer $\Gamma \to \vee(A_1, \ldots, A_r), \Delta$.
- (Mod-left) From $A_1, \oplus_{1-b}(A_2, \ldots, A_r), \Gamma \to \Delta$ and $\oplus_b(A_2, \ldots, A_r), \Gamma \to A_1, \Delta$ infer $\oplus_b(A_1, \ldots, A_r), \Gamma \to \Delta$.
- (Mod-right) From $A_1, \Gamma \to \oplus_{1-b}(A_2, \ldots, A_r), \Delta$ and $\Gamma \to A_1, \oplus_b(A_2, \ldots, A_r), \Delta$, infer $\Gamma \to \oplus_b(A_1, \ldots, A_r), \Delta$.
- (Threshold-left) From $\mathrm{Th}_k(A_2, \ldots, A_r), \Gamma \to \Delta$ and $A_1, \mathrm{Th}_{k-1}(A_2, \ldots, A_r), \Gamma \to \Delta$, infer $\mathrm{Th}_k(A_1, \ldots, A_r), \Gamma \to \Delta$.
- (Threshold-right) From $\Gamma \to A_1, \mathrm{Th}_k(A_2, \ldots, A_r), \Delta$ and $\Gamma \to \mathrm{Th}_{k-1}(A_2, \ldots, A_r), \Delta$ infer $\Gamma \to \mathrm{Th}_k(A_1, \ldots, A_r), \Delta$.