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 $\epsilon$-fraction of the time (for polynomially small $\epsilon$) 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 [31, 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 analysis.

Categories and Subject Descriptors
F.4.1 [Theory of Computation]: Mathematical Logic—Mechanical theorem proving, Proof theory; I.2.6 [Artificial Intelligence]: Learning—Knowledge acquisition

General Terms
Theory

Keywords
learning; proof complexity; automatizability; PAC-Semantics

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 [29, 30]. 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 $\varphi$, and our objective is to decide whether the data set provides empirical evidence that $\varphi$ 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, then there is a complete proof of the query $\varphi$, that we then certify $\varphi$ as satisfied with high probability under $D$. At the same time, we also wish to guarantee that if the query $\varphi$ 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 in Section 1.2: as we will review there, 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. The previous work [24] considered the advantages of this integrated formulation of learning and reasoning from

\[ \text{http://dx.doi.org/10.1145/2688073.2688108} \]
the perspective of learning theory: it enables reliable knowledge discovery and tolerance to adversarial noise. 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 1.1.1). 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 possessing fast algorithms: 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 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 assumptions 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 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. That is, whether or not it is \((1 - \epsilon)\)-valid:

**Definition 1** \((1 - \epsilon)\)-valid \([40]\). Given a distribution \( D \) over \([0, 1]^n\), we say that a Boolean formula \( \varphi \) is \((1 - \epsilon)\)-valid if \( \Pr_{x \in D}[\varphi(x) = 1] \geq 1 - \epsilon \).

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 \( i \)th attribute of each complete example drawn from the ground truth distribution \( D \), a \( \mu \)-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 Gemaro [18], and more recently was employed as the partial information model for Population Recovery [20, 41, 5, 35] (which we will discuss later). Elsewhere, this model is known as “masking completely at random (MCAR)”.²

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 \( x_i \) 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” \( C \). “Parity learning” then means that \( C \) is the class of parity functions: \( x_i = \bigoplus_{j \in S_i} x_j \) for some \( S_i \subseteq [n] \). 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 \( x_i \) is then determined by the unknown parity function. Equivalently, we could say that there is a rule (or constraint) \( \bigoplus_{j \in S_i} x_j = b_j \) and our examples \( (x_1, \ldots, x_n, x_i) \) 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 \( \bigoplus_{j \in S_j} x_j = b_j \) 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”.

**Definition 2** (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.

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.

#### 1.1.1 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 discuss this choice in Section 1.2.) We first note that this is \( \text{NP} \)-hard in general, even for affine distributions masked completely at random:

**Theorem 3.** Let \( M(D) \) be an affine distribution masked completely at random (with constant \( \mu \in (0, 1) \)). Unless \( \text{NP} \) 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 \( \epsilon \).

By contrast, if we only seek to test 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. Precisely, we will say that the proof system is PAC-automatizable (in time \( T \)) if there is a sufficiently fast algorithm to test whether or not a proof exists using rules that simplify to true on the masked examples²:

²A full definition of this operation is deferred to Section 2.
Definition 4 (PAC-automatizability). When we say that a proof system $T(N, 1/\epsilon, 1/\gamma, 1/\delta)$, we mean that there is an algorithm that is given $\varphi$, $\epsilon, \gamma, \delta > 0$, and $N$ as input and obtains samples from $M(D)$ for a given distribution $D$. This algorithm runs in time $T(N, 1/\epsilon, 1/\gamma, 1/\delta)$ and with probability $1 - \delta$ distinguishes $\varphi$ that are $(\epsilon + \gamma)$-valid from $\varphi$ that have a refutation of size $N$ in the system from additional premises $\psi_1, \ldots, \psi_k$ such that $\psi_1 \land \cdots \land \psi_k$ simplifies to true under partial examples drawn from $M(D)$ with probability at least $1 - \epsilon + \gamma$.\footnote{It turns out to be more convenient to state this definition in terms of refutations of $\varphi$ since this is how we will use the algorithms in most circumstances. We could equivalently have said that we were distinguishing when $\neg \varphi$ has a proof from when $\neg \varphi$ is not $(1 - \epsilon - \gamma)$-valid (where $\varphi$ is the query actually provided to the algorithm).}

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 5. Polynomial calculus with resolution (PCR) is PAC-automatizable in quasipolynomial time for affine distributions masked completely at random.

Comparing Theorems 3 and 5, 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\cite{29} 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 in Section 1.2 we will recall the advantages of such an integrated formulation from the standpoint of learning theory (shown by Juba\cite{24}), here we note that it provides a means to avoid difficulties in proof complexity. In particular, Galesi and Lauria\cite{22} (building on earlier work by Alekhnovich and Razborov\cite{2}) gave evidence that the proof search problem for PCR is intractable.\footnote{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.} 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\cite{31}, can be carried over to negative results for our PAC-automatizability problem. Specifically, Bonet et al.\cite{11}, building on earlier work by Bonet, Pitassi, and Raz\cite{13}, showed that the proof search problem for “bounded depth Frege systems” (recalled in Section 2.2) 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.

Theorem 6. If integer factoring requires subexponential time, then bounded-depth Frege is not PAC-automatizable in quasipolynomial time for affine distributions masked completely at random.

Contrasting Theorem 6 with Theorem 5, 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 $\text{AC}^0$-circuit\footnote{Where, recall $\text{AC}^0$ 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.} 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.\cite{3} for characteristic zero fields, building on an earlier, classic work by Razborov\cite{37} that used such a transformation of $\text{AC}^0$ 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.\cite{20} and essentially solved in a sequence of later works\cite{41, 5, 35}. 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.\cite{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\cite{41} or Moitra and Saks\cite{35}) 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.\cite{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.

1.2 Motivation and context

In knowledge discovery and data analytics, the extraction of knowledge from a data set is not an end goal, only an
termediate 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 [27]. 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).

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., \( \psi(x, y) = x = \langle \beta, y \rangle + \beta_0 \pm \epsilon \) (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 [28] 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 [33]. This is, nearly verbatim, the setup assumed in agnostic learning [28], 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 [29], 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 no harder than the proof search alone, 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 \( \varphi \) 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 [27] for example.

Khardon and Roth [29] 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 \( \text{NP} \)-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 [30] 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 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 \( \text{NP} \)-hard without any proof complexity promise (Theorem 3), as hard as breaking Diffie-Hellman key exchange\(^6\) under a weak proof complexity promise (Theorem 22), and solvable in quasipolynomial time under the stronger proof complexity promise of having a (quasi)polynomial size PCR proof (Theorem 5).

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

\(^6\)Therefore, as hard as integer factoring [10].
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 tree-like 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 tree-like 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.

2. BACKGROUND AND PRELIMINARIES

In this work, we consider a variety of proof systems under a weaker semantics than usual. It is not hard to show (by a union bound) that any classical logical inference can be applied to formulas possessing \((1 - \epsilon)\)-validity, as long as we allow for further loss in the approximation.\(^7\)

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

We are therefore safe in considering arbitrary kinds of propositional proof systems in this new setting.

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

- For any Boolean constant \( b, b|_\rho = b \).
- For any variable \( x_i \), if \( \rho_i = \ast \), 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 \( \psi_i \) (indexed by \( 1, \ldots, \ell \)) do not simplify to Boolean values under \( \rho \), then (indexing the rest by \( j_1, j_2, \ldots, j_{k-\ell} \))
  \[
  \oplus(\psi_1, \ldots, \psi_k)|_\rho = \oplus_{i=1}^{\ell} \psi_i|_\rho \oplus_{r=1}^{k-\ell} \psi_{j_r}|_\rho
  \]
  and otherwise it simplifies to a Boolean constant,

- A linear threshold \( \sum_{i=1}^{k} c_i \psi_i \geq b \) simplifies to 1 if \( \sum_{i: \psi_i|_\rho = 1} c_i + \sum_{i: \psi_i|_\rho = 0} c_i \leq \min \{0, c_i\} \geq b \), 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

That is, \( \varphi|_\rho \) is a formula over the variables \( x_i \) such that \( \rho_i = \ast \). 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, \( |P(x) \geq 0 \land [-P(x)] \geq 0 \).

2.1 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 \( \mathbb{F} = \mathbb{Q} \), 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 \( \sum_{\alpha \in \mathbb{N}^n} c_{\alpha} x_1^{\alpha_1} \cdots x_n^{\alpha_n} = 0 \) for coefficients \( c_{\alpha} \in \mathbb{F} \), where the products \( \prod_{i, \alpha \neq \alpha_i} x_i \) are the monomials corresponding to the degree vector \( \alpha \). 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 \( \bar{x}^{\alpha} \delta \) from a larger polynomial constraint \( P(x) = 0 \) and a second monomial constraint of the form \( [-\bar{x}^{\beta}] = 0 \) where \( \beta \leq \alpha \) by adding

\[\sum_{i: \psi_i|_\rho = 1} c_i + \sum_{i: \psi_i|_\rho \notin \{0, 1\}} \max \{0, c_i\} < b, \; and \; \text{otherwise \; is \; given \; by} \; \left( \sum_{i: \psi_i|_\rho = 1} c_i \right) \psi_i|_\rho \geq (b - \sum_{i: \psi_i|_\rho = 1} c_i) \cdot \psi_i|_\rho \].
an appropriate multiple of $-\ell x^d = 0$ to $P(x) = 0$, and then subtracting the same multiple of $[\ell x^d + -\ell x^d - x^d = 0]$ from the result: We then obtain the constraint $[P'(x) = 0]$ in which the monomial $x^d$ is substituted for the monomial $\ell x^d$ i.e., $\ell$ is eliminated. Notice, we can obtain $[P(x) = 0]$ from $[P'(x) = 0]$ and the monomial constraint $-\ell x^d$ again by repeating the derivation except subtracting the multiple of $-\ell x^d$ and adding the multiple of $[\ell x^d + -\ell x^d - x^d = 0]$, thus “reversing” the elimination of $\ell$. This will be important since we can’t freely use multiplication to simulate weakening of the monomials of $P(x)$ independently.

Following a 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 degree-$d$ fragment of polynomial calculus (and PCR) is automatizable in time $n^{O(d)}$.

We will be interested in the result of “plugging in” a partial assignment to each step of a PCR refutation:

**Definition 9 (Restricted proof).** Given a PCR refutation $\Pi$ and partial assignment $\rho$, the restriction of $\Pi$ under $\rho$, denoted $\Pi|_\rho$, is the proof obtained by substituting $\varphi|_\rho$ for each line $\varphi$ of $\Pi$.

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 10.** For any PCR refutation $\Pi$ and any partial assignment $\rho$, the restriction $\Pi|_\rho$ is also a PCR refutation.

### 2.2 Bounded-depth Frege

We will use the standard bounded-depth Frege sequent systems of propositional logic defined by Maciel and Pitassi [32]. In these systems, each line is of the form $A_1, \ldots, A_s \rightarrow B_1, \ldots, B_t$ (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, $\text{AC}_0$ and $\text{TC}_0$. These classes both use the connectives $\vee$, $\wedge$, and $\neg$, and $\text{TC}_0$ 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 $\Theta_b$ connective, a threshold connective that is true iff at least $b$ 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.

The main result of Bonet et al. [11] is essentially a translation from $\text{TC}_0$-Frege proofs to $\text{AC}_0$-Frege proofs:

**Theorem 11 (Theorem 6.1 of Bonet et al. [11]).** Suppose that $\Gamma \rightarrow \Delta$ has a $\text{TC}_0$-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 $\text{AC}_0$ formula equivalent to $\Gamma \rightarrow \Delta$ that is polynomial-time computable from $\Gamma \rightarrow \Delta$ and has an $\text{AC}_0$-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 $\Gamma \rightarrow \Delta$) but this is immediate. This translation enables non-automatizability for (sufficiently simple) specific $\text{TC}_0$-Frege formulas to be carried over to non-automatizability for $\text{AC}_0$-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 $\theta$ and propositional formula $\nu$, we typically denote the result of applying $\theta$ to $\nu$ by $\theta \nu$. 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 12.** Let $\theta$ be any substitution taking variables to depth-$d_1$ formulas, and suppose that there is a depth-$d_2$ Frege proof of $\varphi$ from $\{\psi_1, \ldots, \psi_s\}$. Then there is a depth-$(d_1 + d_2)$ Frege proof of $\theta \varphi$ from $\{\theta \psi_1, \ldots, \theta \psi_s\}$.

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 $\text{TC}_0$-Frege proofs to $\text{TC}_0$-Frege proofs.

### 3. Technical 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 $\mu$ for $M_\mu(D)$.

#### 3.1 Sketch of Theorem 5

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 closed under restrictions (natural in the sense of Beame, Kautz, and Sabharwal [6]). That is: it provides a simple technique for searching over conjunctions of premises $\psi$ such that $\psi|_\rho$ simplifies to the constant 1 for $\rho \in M_\mu(D)$ with probability $(1 - \epsilon + \gamma)$ (we say $\psi$ is $(1 - \epsilon + \gamma)$-testable) and such that there is a proof of the query $\varphi$ from $\psi$ in the given proof system. The observation is that for a restriction-closed proof system, when every step of the proof is hit by the restriction $\rho$, the desired premises $\psi$ will vanish from the proof with high probability. Thus, it suffices to check the possibility of $\varphi|_\rho$ on various partial examples $\rho$ 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 $\rho$ drawn from
bias gap
to be simulated in low degree. We observe that affine dis-
trictions from affine distributions, the surviving monomi-
variables is either strong or weak (and not of moderate
Latent Semantic Indexing generally features a nontrivial bias
model introduced by Papadimitriou et al. [36] to analyze
we note that the simple
Definition 13 (Bias gap).
Definition 18. Suppose D is a distribution with a width-d
distribution given by the linear system Ax = b if there is a lin-
eral 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^α.
Lemma 15. Let x^α be a monomial and D an affine dis-
then the marginal distribution over the variables appearing in x^α
is uniform.
On account of a bias gap, we show that a monomial of log-
arithmetic 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 16. Let x^α be a monomial of degree at least \(\frac{2}{\delta} \ln \frac{1}{\delta}\)
such that for any literal \(\ell\) of x^α and any submonomial x^α'
we observe that x^α' survives. Then x^α'|x = 0 on \(p \in M_p(D)\) with probability \(1 - \delta\).
The second lemma considers when the variables may be fixed
to take value 0:
Lemma 17. Let x^α be a monomial of degree \(2d\) for \(d \geq \frac{2}{\delta} \ln \frac{1}{\delta}\) such that for every submonomial of x^α of degree \(\delta\)
there is some further submonomial x^α'' such that condi-
tioned on x^α'' surviving, \(\ell\) is set to 0. Then with probability \(1 - \delta\), there is an unmasked variable of x^α that is fixed to zero under \(p \in M_p(D)\).
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
(\(\frac{2}{\delta} \ln \frac{1}{\delta}\)) submonomial of a surviving monomial in
our polynomial calculus proof must have some variable that
is fixed to 0, conditioned on the rest of the submonomial
surviving. Indeed, otherwise either there is a degree \(\frac{2}{\delta} \ln \frac{1}{\delta}\)
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 simul-
taneously unmasked (under \(M_p\)) 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 below:
Lemma 18. Suppose D is a distribution with a width-
(\(\beta, 1 - \frac{\gamma d}{2(2n+1)^2}\)) bias gap. Let \(\psi\) be the conjunction of con-
straints \(x^α = 0\) for all monomials x^α of degree at most d
that simplify to the value 1 in at most a \(\frac{\gamma d}{2(2n+1)^2}\)-fraction
of a sample of \(m_0\) partial examples from \(M_p(D)\) (for \(m_0 =
\frac{\delta d}{\psi} \ln \frac{1}{\delta}\)). Then with probability at least \(1 - \delta/2\),
\(\psi\) is 1 - \(\gamma\)-valid and contains all 1 - \(\gamma\)-valid mono-
monomial constraints of degree at most d, including specifically
all ¬x^α such that \(\ell\) is implied by x^α.
The heart of our analysis now is that we show that these
small monomial constraints allow us to simulate (the sur-
viving 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 condi-
tioned 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 elimi-
nate these variables that are fixed to 1 by resolution steps
using our learned monomial constraints. That is, if we quo-
tient 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

**Lemma 19.** Let $\psi$ be a conjunction of constraints of the form $x^d = 0$ 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 $\psi$. Then there is a derivation of one from the other of degree at most $3d$.

We prove Lemma 19 by showing 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 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 $d = O(1 + \ln \frac{n}{\delta})$ on the polynomials involved, it turns out that the overall algorithm runs in time $O(m \cdot n^{O(d)})$, where $m = \frac{\gamma}{\epsilon} \ln \frac{2}{\delta}$; 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.

In full generality, we obtain:

**Theorem 20.** (cf. Theorem 5). Given an input system of polynomial equations $\varphi$, 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-$d \left( = \frac{1}{\mu^2} \ln \frac{2^{2n+O(p)}}{\delta} \right)$ bias gap, (where $m_i = \frac{\gamma}{\epsilon \mu^2} \ln \frac{2}{\delta}$) and given that either

- $\varphi$ is satisfied by $D$ with probability at least $\epsilon + 2\gamma$ or
- there exist systems of polynomial equations $\psi_0$ and $\psi_1$ such that
  - $\psi_0$ consists of $1 - \frac{\gamma}{\epsilon \mu^2} \ln \frac{2^{2n+O(p)}}{\delta}$-valid monomials and
  - $\psi_1$ simplifies to 1 with probability $1 - \epsilon + 4\gamma$ under $M_\mu(D)$

and there is a PCR refutation of size $p(n)$ of $\psi_0 \land \psi_1 \land \varphi$ there is an algorithm that decides which case holds with probability $1 - \delta$, and runs in time $O\left( \frac{1}{\mu^3} \log \frac{2^{2n+O(p)}}{\delta} \right)$.

### 3.2 Sketch of Theorems 3 and 6

The main observation underlying both of the negative results is that affine distributions present hard examples because 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 Dichter’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 21) 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. Precisely, for any given setting of parity values for some basic variables $x_1, \ldots, x_n$, we substitute a parity of a vector of $k(n)$ new variables for each basic variable, and let $D^{\psi(k(n))}$ be the distribution in which these parities are constrained to take the same values as the underlying (secret) values of the basic variables.

**Lemma 21.** Let $x^* \in \{0, 1\}^n$ and $\rho$ consistent with $x^*$ be given. Let $\theta$ be a substitution that takes each variable $x_i$, such that $\mu_i = \rho$ to a parity of $k(n)$ new variables, $y_{1(i)} \ldots y_{k(n)}$, and leaves all other variables fixed. Let $D^{\psi(k(n))}$ be the distribution over this new set of variables such that the variables left fixed by $\theta$ take the same value as in $x^*$ and the new variables are uniformly distributed over values satisfying $y_{1(i)} \oplus \cdots \oplus y_{k(n)} = b$ where $x_i^* = b$. Then for any $p$-valid formula $\varphi$ under the point distribution for $x^*$, $\theta \varphi$ is also $p$-valid under $D^{\psi(k(n))}$. Moreover, there is a distribution that can be sampled in linear time given $\rho$ that is $1 - n \mu^{k(n)}$, statistically close to $M_\mu(D^{\psi(k(n))})$.

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 [37] and Aspnes et al. for characteristic zero [3]), at the cost of introducing some small error $\epsilon$ 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 $\epsilon$. This is Theorem 3.

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 TC0-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.\(^8\) Again, Lemma 21 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 22, showing a range of hardness results for PAC-automatizing TC0-Frege for a corresponding range of security assumptions for Diffie-Hellman key exchange.

**Theorem 22.** Suppose TC0-Frege is PAC-automatizable for affine distributions under \(M_\mu\) for constant \(\mu\) in time \(T(N, 1/\epsilon, 1/\gamma, 1/\delta)\) for \(T(N, 2, 3, 1/\delta) \geq \Omega(N^c)\) and \(T(N, 2, 3, 1/\delta) < 2^{\Omega(N \log 1/\delta)}\) for sufficiently large \(c\). Then for some polynomial \(P\), there is an algorithm running in time \(O(nT(P(n) \log 1/\epsilon, 2, 3, 1/\gamma))\) that recovers \(g^a \mod p\) with probability \(1 - \delta\) from any \(n\)-bit \(p\), generator \(g\) of \(\mathbb{Z}_p^*\), \(g^a\) mod \(p\), and \(g^b\) mod \(p\), where \(a\) and \(b\) are arbitrary.

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 TC0-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 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 21 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 \(2^{\Omega(n)}\)-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 lower bound for \(\eta > 0\), and Theorem 6 follows.

4. DIRECTIONS FOR FUTURE RESEARCH

Several problems present themselves as natural directions for future work. The most pressing of these, *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 [34]; 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 an 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 tree-like) 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 \(\psi\) 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 non-automatizable 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 thanks Paul Beame, Eli Ben-Sasson, and Leslie Valiant for conversations that helped shape this work. The author also thanks the anonymous reviewers for their constructive comments.

5. REFERENCES


