

Polynomial Interpolation for Sherali-Adams Lower Bounds

Nathan Acheampong
Université de Montréal

Ben Davis
McGill University

Robert Robere¹
McGill University

October 30, 2026

Abstract. We resolve two open problems regarding the Sherali-Adams proof system. First, we show that $\text{Res}(2)$, which manipulates 2-DNF formulas, is exponentially stronger than Sherali-Adams. This provides a sharp separation in power between Sherali-Adams and the bounded-depth Frege hierarchy, as Sherali-Adams polynomially simulates Resolution. Second, we prove that Sherali-Adams is not size-automatable unless $P = NP$. This resolves an open problem on the automatability of weak proof systems asked by many researchers. Both of these results are obtained by using a new technique for proving lower bounds on Sherali-Adams degree via polynomial interpolation.

¹R.R. supported by NSERC, and partially supported by a grant from the Simons Foundation. Part of this work was performed while R.R. was visiting the Logical Foundations of Complexity program, hosted at the Newton Institute at the University of Cambridge.

Contents

1	Introduction	1
1.1	Motivation and Background	1
1.2	Witnessing Formulas: Sink-of-DAG, Coloured PLS, and LOSSY(SoD)	3
1.3	Our Technique: Polynomial Interpolation.	7
1.4	Related Work	13
1.5	Organization	16
1.6	AI Disclosure	16
2	Algebraic Proof System Preliminaries	16
2.1	$\mathbb{R}$ -Nullstellensatz and Sherali-Adams.	16
2.2	Designs and Pseudoexpectations for Degree Lower Bounds	17
2.3	Indicator Algebras and Block-Degree Lower Bounds	19
I	Polynomial Interpolation for Pseudoexpectations	23
3	Warmup and Completeness of Interpolation	23
3.1	Polynomial Interpolation for the Bit Pigeonhole Principle	24
3.2	Polynomial Interpolation is Complete	26
4	$\mathbb{R}$ -Nullstellensatz Degree Bounds for SoD	28
4.1	The SoD Design	28
4.2	Proof of SoD Degree Lower Bounds	29
II	Separating Sherali-Adams and Res(2)	32
5	The CPLS Formula	32
6	The CPLS Pseudoexpectation	32
7	Proof of CPLS Degree Lower Bound	33
7.1	Setup and Technical Lemmas	34
7.2	The CPLS Pseudoexpectation is Non-Witnessing	35
7.3	The CPLS Pseudoexpectation is Non-Negative	36
7.4	Proofs of Technical Lemmas	37
8	Application: Separating Sherali-Adams and Res(2)	44
III	Non-automatability of Sherali-Adams	47
9	The Formulas and Reduction Preliminaries	47
9.1	The REF formula and Lossy(SoD)	47
9.2	Node-wise Reductions and Pseudoexpectations	49
10	Non-Automatability from Lossy(SoD) Pseudoexpectations	52
11	The Lossy(SoD) Pseudoexpectation	54
11.1	Intermediate Measures for SoD	54
11.2	The Final Construction for Lossy(SoD)	57
12	Proof of Lossy(SoD) Degree Lower Bounds	57
12.1	Setup and Technical Lemmas	58
12.2	The Lossy(SoD) Pseudoexpectation is Non-Witnessing	59
12.3	The Lossy(SoD) Pseudoexpectation is Non-Negative	60
	References	62

1 Introduction

The *Sherali-Adams proof system* [SA90] is the prototypical example of a *semi-algebraic* proof system manipulating polynomial inequalities over the reals, and one of the central proof systems studied in both proof complexity and optimization. In this paper, we present solutions to two significant open problems involving this proof system. Our first contribution is an exponential separation between Sherali-Adams and Res(2), which is a well-studied boolean proof system whose lines are 2-DNF formulas.

Theorem 1.1. *Sherali-Adams does not polynomially-simulate the Res(2) proof system. More specifically, there is an infinite family of unsatisfiable polynomial-size n -variate CNF formulas $\{F_n\}$ that have polynomial-size Res(2) refutations, but every Sherali-Adams refutation requires size $2^{\Omega(n^{1/3})}$.*

To state our second contribution, we recall that a proof system P is *automatable*² if there is an algorithm A which, given any tautology F , outputs a P -proof of F in time which is polynomial in the combined length of the shortest P -proof of F and the size of F . Automatability is a very well-studied concept in proof complexity, and represents a complexity-theoretic formulation of the *proof-search* problem implemented by many practical constraint solvers, such as those following the *CDCL paradigm* [MMZ⁺01, MS99, BS97].

Our second main result is that Sherali-Adams is not automatable unless $P = NP$. Indeed, we can even show that it is NP-Hard to approximate the length of the shortest Sherali-Adams refutation up to a 2^{n^ε} factor for some $\varepsilon > 0$.

Theorem 1.2. *Automating Sherali-Adams is NP-Hard. More precisely, there is a polynomial-time algorithm R which, given an n -variate 3-CNF formula F , outputs a CNF formula $R(F)$ such that:*

- *If F is satisfiable, then $R(F)$ has a polynomial-size Sherali-Adams refutation.*
- *If F is unsatisfiable, then every Sherali-Adams refutation of $R(F)$ requires size $2^{n^{\Omega(1)}}$.*

Such an algorithm immediately refutes automatability assuming $P \neq NP$: If A automated Sherali-Adams, then we could solve 3-SAT in polynomial time by running $A(R(F))$ and outputting “SAT” if it halted within a certain fixed polynomial time-bound, determined by the theorem. To the best of our knowledge, this is the first result regarding the non-automatability of the Sherali-Adams proof system in the literature, and the assumption that $P \neq NP$ is optimal, as if $P = NP$ then every proof system is automatable.

1.1 Motivation and Background

While both of these problems are naturally motivated in their own right, a major motivation for their study is due to their resistance to standard techniques. We begin by reviewing some background and related work regarding these two results, and also discuss some of the apparent roadblocks towards resolving them by known techniques.

Sherali-Adams vs Bounded-Depth Frege. We consider Sherali-Adams as a proof system for refuting polynomial encodings of unsatisfiable CNF formulas. Towards this, let $F = C_1 \wedge \dots \wedge C_m$ be a CNF formula on n variables $x_1, \dots, x_n$. We introduce n new *dual variables* $\bar{x}_1, \dots, \bar{x}_n$, and consider polynomials in the ring $\mathbb{R}[x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n]$. Each variable x_i is constrained to be boolean over $\mathbb{R}$ by adding the equations $x_i^2 = x_i$, and the variable $\bar{x}_i$ is constrained to be the boolean negation of x_i by adding the equation $x_i + \bar{x}_i = 1$. A conjunction of literals $\bigwedge_i x_i \wedge \bigwedge_j \neg x_j$ is encoded by the product term $\prod_i x_i \prod_j \bar{x}_j$; without loss of generality, we will also refer to these product terms as conjunctions. With these ingredients, we

²This notion was originally introduced in the literature as *automatizability* [BPR00], but recent papers have opted for the shorter name.

can encode the truth of each clause C_i as a real polynomial equation: if, e.g. $C_i = x \vee \neg y$, we define the conjunction $\overline{C}_i = \overline{x}y$, and C_i is satisfied if and only if $\overline{C}_i = 0$. A Sherali-Adams refutation of F is a polynomial identity of the form

$$\sum_{i=1}^m p_i \overline{C}_i + \sum_{j=1}^n q_j (x_j^2 - x_j) + r_j (1 - x_j - \overline{x}_j) = J + 1, \quad (1)$$

where $p_i, q_j, r_j \in \mathbb{R}[x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n]$ are real polynomials for each i, j , and J is a *conical junta* term: this is a non-negative linear combination $J = \sum_k \lambda_k T_k$, where each T_k is itself a conjunction. The *degree* of the refutation is the maximum degree of any of the polynomials $p_i \overline{C}_i$, $q_j (x_j^2 - x_j)$, $r_j (1 - x_j - \overline{x}_j)$, and T_k , and the *size* of the refutation is the number of monomials produced by expanding all polynomial expressions in the proof before performing any cancellations.

[Theorem 1.1](#) gives a sharp separation between Sherali-Adams and the *bounded-depth Frege* system, which is the propositional proof system whose lines are boolean formulas over the DeMorgan basis with bounded alternation depth between $\wedge$ and $\vee$. It is natural to stratify the bounded-depth Frege system into fragments based on the alternation depth of the formulas involved. For instance, in the alternation depth-0 fragment of this system, the lines are clauses, and we recover the well-studied *Resolution* proof system, which is known to be efficiently simulated by Sherali-Adams [\[FKP19\]](#). In the depth-1 fragment, the lines are now *DNF formulas*, and we obtain a proof system sometimes denoted by $\text{Res}(n)$, representing that we are allowed to reason about DNF formulas with terms of width $\leq n$ (where n is the number of underlying variables of the formula). More generally, the $\text{Res}(k)$ system is a fragment of $\text{Res}(n)$ whose lines are k -DNFs: $\text{Res}(1)$ is equivalent to Resolution, while $\text{Res}(2)$ is already known to be exponentially stronger than Resolution [\[AB04\]](#).

While Sherali-Adams and bounded-depth Frege have both received significant and sustained attention by researchers over the years, the question of their precise relationship has been left unanswered. In one direction, it was known that Sherali-Adams has polynomial-size refutations of the pigeonhole principle [\[DMR09\]](#), which, after a long line of work, is now known to require size $2^{\Omega(n^{1/d})}$ to refute in depth- d Frege systems [\[Ajt94, PBI93, KPW95, Hås25\]](#) for any constant d . The other direction has been apparently left completely open by the literature, but it is not hard to use known techniques to construct a family of formulas which require exponential-size Sherali-Adams refutations, but have *quasi*-polynomial size bounded-depth Frege refutations³. In fact, the quasi-polynomial upper bound on these formulas can even be carried out in the proof system $\text{Res}(\log)$, which is the fragment of bounded-depth Frege whose lines are DNF formulas with width $\log S$, where S is the length of the proof [\[Kra01\]](#).

However, this separation is unsatisfying for a few reasons. First, it is quantitatively weak, yielding only a quasi-polynomial upper bound in bounded-depth Frege. Second, it leaves open the precise relationship between Sherali-Adams and the fragments of bounded-depth Frege, as there is quite a significant gap between $\text{Res}(\log)$ and $\text{Res}(1) = \text{Resolution}$, which is the strongest fragment of bounded-depth Frege that Sherali-Adams is known to efficiently simulate. Thus, in both of these respects, our [Theorem 1.1](#) gives a sharp separation. Regarding the first point, it gives an exponential-versus-polynomial separation between Sherali-Adams and bounded-depth Frege. Regarding the second point, it shows that even extending Resolution to $\text{Res}(2)$ is enough to give exponential power over Sherali-Adams.

Non-Automatability of Sherali-Adams. It has been known since the 1990s that strong proof systems are not automatable assuming the security of various cryptographic primitives [\[KP98, BPR00, BDG⁺04\]](#). For

³The weak bit pigeonhole principle requires exponential-size Sherali-Adams refutations, but has quasi-polynomial-size $\text{Res}(\log)$ refutations. The first result follows from degree lower bounds and the size-degree relation for Sherali-Adams [\[AH20, Hak22\]](#), and the second result follows from an adaptation of the $\text{Res}(\log)$ proofs of the weak pigeonhole principle in [\[MPW02\]](#). As the result in [Theorem 1.1](#) is stronger, we leave the details as an exercise.

weaker proof systems, much less was known for many years, with perhaps the strongest result being due to Alekhovich and Razborov [AR08], who showed that Resolution is not automatable assuming a conjecture in parametrized complexity theory. A relatively recent breakthrough, due to Atserias and Müller [AM20], proves that automating Resolution is NP-Hard by showing, informally, that Resolution *cannot* prove lower bounds on its own length. An exciting line of follow-up work building on [AM20] has shown that automating a variety of weak proof systems is NP-Hard, including the Nullstellensatz and Polynomial Calculus proof systems [dRGN⁺21], the Res(k) proof system for all constant k [Gar24], the semi-algebraic Cutting Planes proof system [GKMP20], among others (cf. Section 1.4). Extending these results to the Sherali-Adams proof system has been asked by many researchers over recent years, although it has seemingly resisted all previous attacks⁴.

To discuss our approach, we must dive into the details of the Atserias-Müller proof [AM20]. If F is a 3-CNF formula on n variables, and s is a positive integer parameter, then we can define a CNF formula $\text{REF}_s(F)$ which is satisfiable if and only if F has a Resolution refutation of length at most s . The relationship between F and $\text{REF}_s(F)$ has three possibilities:

1. F is satisfiable. In this case, $\text{REF}_s(F)$ is unsatisfiable, and Pudlák [Pud03] proved that there is even a polynomial-size Resolution refutation of $\text{REF}_s(F)$.
2. F is unsatisfiable, and every Resolution refutation of F requires size $> s$. In this case, $\text{REF}_s(F)$ is unsatisfiable.
3. F is unsatisfiable, and there is a Resolution refutation of F with size $\leq s$. In this case, $\text{REF}_s(F)$ is satisfiable (and there is no refutation of $\text{REF}_s(F)$ at all).

Therefore, a refutation of $\text{REF}_s(F)$ proves that every Resolution refutation of F has length greater than s . Since $\text{REF}_s(F)$ is easy to refute in Resolution when F is satisfiable, if $\text{REF}_s(F)$ is hard to refute when F is unsatisfiable, then any polynomial-time automating algorithm could be used to distinguish satisfiable and unsatisfiable formulas, showing $\text{P} = \text{NP}$.

Following this template, Atserias and Müller showed that $\text{REF}_{n^{O(1)}}(F)$ did indeed require exponentially long Resolution refutations whenever F is unsatisfiable, showing that Resolution is conditionally non-automatable. The later work on non-automatability of weak proof systems followed this same template, proving lower bounds for $\text{REF}_s(F)$ in each of the various proof systems considered, sometimes with some local technical modifications on the formula $\text{REF}_s(F)$. This is also our approach: to prove Theorem 1.2, we follow the Atserias-Müller framework, and prove exponential lower bounds on the lengths of Sherali-Adams proofs for a (slightly modified) version of $\text{REF}_s(F)$. (In fact, we will actually prove a lower bound for a formula which is different from $\text{REF}_s(F)$, and combine it with a reduction to obtain to the final lower bound.) In the next section, we will discuss the difficulty underlying this construction, our technical contributions, and the relationship between these two problems.

1.2 Witnessing Formulas: Sink-of-DAG, Coloured PLS, and LOSSY(SOD)

At this point, the reader may be asking themselves: why are these two problems related to each other? It turns out that their common feature — and the key to their resolution — lies in a formula that is *easy* for Sherali-Adams (and even Resolution) to refute: the *Sink-of-DAG* formula.

For any positive integer n , the Sink-of-DAG formula, denoted SOD_n , is defined as follows. Imagine an $n \times n$ grid of nodes, where each node $u = (i, j) \in [n]^2$ has an associated *activity bit* $a_u \in \{0, 1\}$ and, if $i < n$, an associated binary-coded *successor pointer* $s_u \in [n]$. If $u = (i, j)$ is a node with $i < n$, then we call the node $(i+1, s_u)$ the *successor* of u . The SOD_n formula asserts the following inconsistent constraints.

⁴The first circulated version of [dRGN⁺21] claimed that Sherali-Adams was not automatable unless $\text{P} = \text{NP}$, but this result was later retracted.

First, the *distinguished source* node $(1, 1)$ is guaranteed to be active, meaning $a_{11} = 1$. Second, every active node must have an active successor. Finally, any node $u = (n, j)$ on the last row is inactive. We can refute this formula by proving, by induction on $i = n, n - 1, \dots, 1$, that every node on layer i is inactive, which contradicts the guaranteed activity of the distinguished source node. This inductive refutation can be implemented in polynomial-size and $\text{poly}(\log n)$ -width Resolution refutations, and hence by the known simulations, also has Sherali-Adams refutations with the same parameters. The two formulas that we use to prove [Theorem 1.1](#) and [Theorem 1.2](#) can be intuitively viewed as *obfuscated* variants of the Sink-of-DAG principle. We now introduce these two formulas, and discuss their relationship with SOD_n .

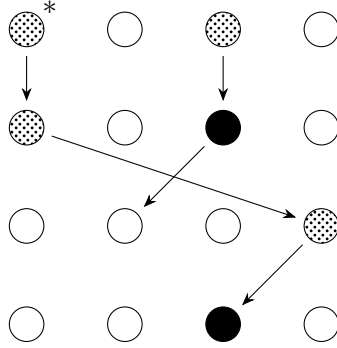


Figure 1: An instance of SOD_4 , where all shaded nodes are active and the distinguished source is indicated by *. Nodes which violate the formula are shaded solid. $(4, 3)$ is a violation since it is on the final row but is active, while $(2, 3)$ is a violation because it is active but its successor is not.

Coloured PLS. The formula we use for separating Sherali-Adams from $\text{Res}(2)$ is based on the *Coloured PLS formula* (denoted CPLS), which was introduced by Krajíček, Skelley, and Thapen [\[KST07\]](#). The formal definition of the formula is technical (cf. [Definition 5.1](#)), but we can sketch the basic idea as follows. For any positive integer n , the CPLS_n formula has a similar setup as SOD_n , and we will give a simplified description of the formula now. We once again have an $n \times n$ grid of nodes, where each node $u \in [n]^2$ lying above the last layer has an associated successor pointer $s_u \in [n]$. However, each node u now has an associated set of *colours* $S_u \subseteq [n]$, encoded by a set of n variables $c_{u,\alpha} \in \{0, 1\}$ for each $\alpha \in [n]$. The CPLS_n formula then asserts the following constraints. First, the distinguished source $(1, 1)$ has an empty colour set $S_{11} = \emptyset$. Second, if node u has v as its successor, then all colours at v also appear at u : $S_u \supseteq S_v$. Finally, every node $u = (n, j)$ on the bottom level has at least one colour.

Intuitively, it is helpful to think of the CPLS_n formula as the SOD_n formula where the activity of each node has been obfuscated by its colours: in CPLS_n , we regard a node as being active if it has no colours. By mimicking the refutation for SOD_n , we can give a polynomial-size Resolution refutation of this formula by induction: we prove, for each layer $i = n, n - 1, \dots, 1$, that every node on that layer has at least one colour. This refutation can still be implemented in Resolution (and also in Sherali-Adams) in polynomial size, but now the refutations have *large width*: in order to assert that any node u has at least one colour, we must write down the disjunction $\bigvee_{\alpha \in [n]} c_{u,\alpha}$. Indeed, it turns out that *any* Resolution refutation of CPLS_n requires large width:

Theorem 1.3 ([\[Tha16\]](#)). *For all sufficiently large n , any Resolution refutation of CPLS_n requires width at least $\Omega(n)$.*

Since it has short Resolution proofs, the CPLS formula will not be enough to separate Sherali-Adams from $\text{Res}(2)$. So, we will follow a standard tactic in proof complexity, and make the CPLS formula harder

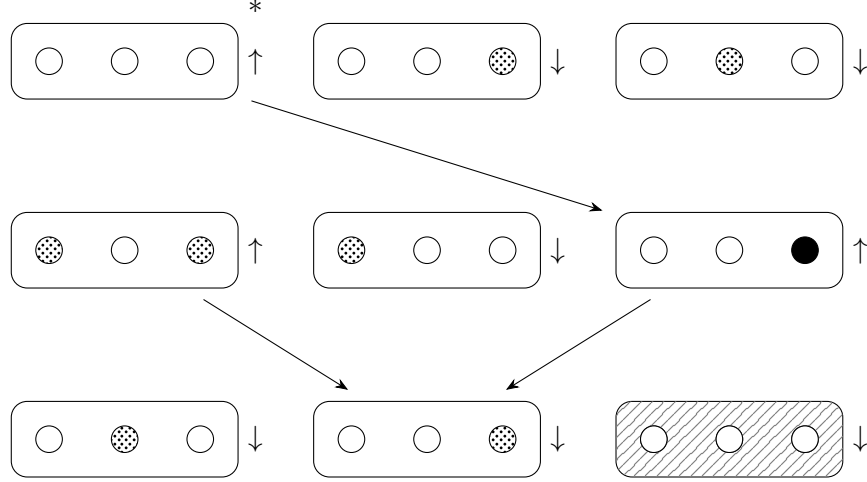


Figure 2: An instance of CPLS_3 . The colours present at each node are shaded, and activities are indicated by an arrow to the right of each node ($\uparrow$ indicates the node is active, $\downarrow$ inactive). The third colour at (2, 3), shaded solid above, violates the formula because it is not present at (1, 1). The striped node (3, 3) also violates the formula since it is inactive but contains no colours.

by *lifting* it. As we mentioned above, since any Resolution proof of CPLS requires large width, if we compose the CPLS formula with a simple gadget (like a two-bit XOR $\oplus$), a standard random restriction argument implies that $\text{CPLS} \circ \oplus$ needs exponential size Resolution refutations. Meanwhile, in $\text{Res}(2)$, a simple simulation argument (replicated in [Theorem 8.4](#)), shows that $\text{CPLS} \circ \oplus$ will still have polynomial-size refutations. For Sherali-Adams, the same random restriction argument would imply an exponential lower bound for $\text{CPLS} \circ \oplus$, provided that we can extend [Theorem 1.3](#) to Sherali-Adams. This is precisely what we do:

Theorem 1.4 (cf. [Theorem 6.2](#)). *For all sufficiently large n , any Sherali-Adams refutation of CPLS_n requires degree $\Omega(n)$.*

However, proving Sherali-Adams degree lower bounds for the CPLS_n formula turns out to be a surprisingly difficult and technical task. We will return to these technical difficulties in the next section, and turn now to discussing our lower bounds for $\text{REF}_s(F)$.

The LOSSY(SOD) formula and $\text{REF}_s(F)$. Surprisingly, the SOD formula also turns out to be closely related to $\text{REF}_s(F)$, thanks to the recent work of Li, Li, and Ren [[LLR26](#)]. Informally, Li, Li, and Ren showed that any refutation of $\text{REF}_s(F)$ could be efficiently transformed into a refutation of a formula denoted $\text{LOSSY}(\text{SOD})$, which we will discuss now⁵.

The $\text{LOSSY}(\text{SOD})$ formula can be viewed as a composition of SOD with another formula, denoted LOSSY , which we will discuss first. Let ℓ be a positive integer, and let $n = 2^{\ell+1}$. In simple terms, the LOSSY_n principle asserts the following: for any pair of functions

$$h : \{0, 1\}^{\ell+1} \rightarrow \{0, 1\}^{\ell}, \quad g : \{0, 1\}^{\ell} \rightarrow \{0, 1\}^{\ell+1},$$

there is a string $x \in \{0, 1\}^{\ell+1}$ such that $g(h(x)) \neq x$. To express this as a CNF contradiction, the functions h and g will both be encoded by strings $\{h_x \in \{0, 1\}^{\ell} \mid x \in \{0, 1\}^{\ell+1}\}, \{g_y \in \{0, 1\}^{\ell+1} \mid y \in \{0, 1\}^{\ell}\}$.

⁵Our notation is slightly different from the notation of [[LLR26](#)]: they denote LOSSY by RWPHP , and instead of SOD they use an *Iteration* principle ITER .

The LOSSY_n formula is then a $n^{O(1)}$ -size, width- $O(\ell)$ CNF formula asserting that $g(h(x)) = x$ for every $x \in \{0, 1\}^{\ell+1}$.

LOSSY has a natural interpretation in terms of codes: if we view the function h as a “compression” algorithm, converting plaintexts $x \in \{0, 1\}^{\ell+1}$ into codes $h(x) \in \{0, 1\}^\ell$, and g as a “decompression” algorithm mapping codes back to plaintexts, then the principle asserts that there is no lossless compression. It turns out that Sherali-Adams has polynomial-size and $O(\log n)$ -degree refutations of the LOSSY_n formula, which follows the natural proof by contradiction: if such a pair of functions h and g existed, then $n \geq 2n$, which is impossible.

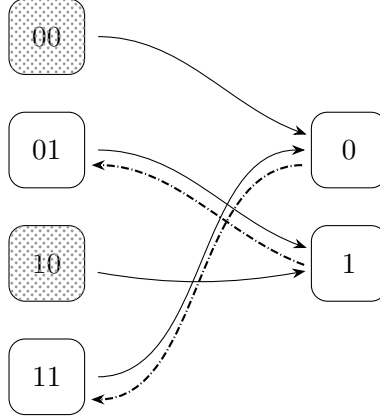


Figure 3: An instance of LOSSY_4 . h is shown by the solid lines, and g by the dashed lines. In this instance the formula is violated by plaintexts 00 and 10, shaded above.

The $\text{LOSSY}(\text{SOD})_{n,m}$ formula extends the LOSSY_n formula by replacing h with *embedded copies* of the SOD_m formula. We sketch the idea of the construction, and leave the full details to [Part III](#). In $\text{LOSSY}(\text{SOD})_{n,m}$, we retain the function $g : \{0, 1\}^\ell \rightarrow \{0, 1\}^{\ell+1}$ from LOSSY , which is still encoded by strings $g_y \in \{0, 1\}^{\ell+1}$ for each $y \in \{0, 1\}^\ell$. For each $x \in \{0, 1\}^{\ell+1}$, we introduce variables $a_{x,u}$ for each $u \in [m]^2$ and successor variables $s_{x,u}$ for each $u \in [m-1] \times [m]$, which encode an instance $\text{SOD}_m^{(x)}$ associated with each plaintext x . In addition, for each $x \in \{0, 1\}^{\ell+1}$ and node $u \in [m]^2$, we add new variables $f_{x,u} \in \{0, 1\}^\ell$, which together encode a function $f : \{0, 1\}^{\ell+1} \times [m]^2 \rightarrow \{0, 1\}^\ell$. The constraints of the $\text{LOSSY}(\text{SOD})_{n,m}$ formula assert the following: for each $x \in \{0, 1\}^{\ell+1}$, and each node $u \in [m]^2$, if the node u is associated with a violated constraint of $\text{SOD}_m^{(x)}$ on a given assignment to $a_{x,u}, s_{x,u}$ variables, then $g(f(x, u)) = x$.

Building on work by Göös et al [[dRGN⁺21](#)], Li, Li, and Ren constructed a kind of reduction from $\text{LOSSY}(\text{SOD})_{n,m}$ to $\text{REF}_s(F)$ when F is unsatisfiable, allowing a transfer of refutations from $\text{REF}_s(F)$ to $\text{LOSSY}(\text{SOD})_{n,m}$ [[LLR26](#)]. The overall approach is the same as our lower bound for CPLS_n : we prove a strong enough degree lower bound, and then use a random restriction argument to escalate this to a size lower bound.

Theorem 1.5. *For all sufficiently large integers n and m , any Sherali-Adams refutation of $\text{LOSSY}(\text{SOD})_{n,m}$ requires degree $\Omega(\min\{n, m^{1/3}\})$.*

The fact that such a degree lower bound is possible for the $\text{LOSSY}(\text{SOD})$ problem is particularly interesting in this regard, as $\text{LOSSY}(\text{SOD})$ is the composition of two problems, LOSSY and SOD , which both have low-degree refutations in Sherali-Adams.

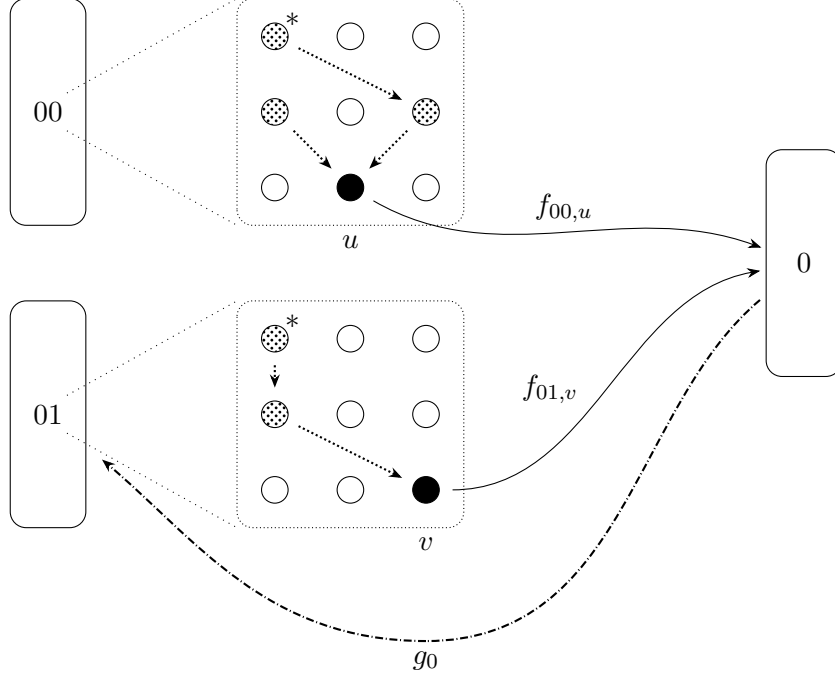


Figure 4: An instance of $\text{LOSSY}(\text{SOD})_{4,3}$. For simplicity, we show only the portions corresponding to the first two plaintexts, 00 and 01, and the first code, 0. The SOD instances associated with each plaintext are shown in the dotted boxes. Active SOD nodes are shaded, with solid shading used for nodes violating SOD. In this instance, the formula is violated by the node u associated with the plaintext 00, since u violates SOD and $f_{00,u} = 0$, but $g_0 \neq 00$. Notice that although v is also a violation in the SOD instance attached to the plaintext 01, this does not violate $\text{LOSSY}(\text{SOD})$ since in this case the compression and decompression pointers agree.

1.3 Our Technique: Polynomial Interpolation.

To summarize the prior discussion: in order to prove [Theorem 1.1](#) and [Theorem 1.2](#), it suffices to prove strong degree lower bounds on two principles, CPLS_n and $\text{LOSSY}(\text{SOD})_{n,m}$, which are “obfuscated” variants of the SOD_m problem. To prove these degree lower bounds, we develop a new technique for constructing *pseudoexpectations* using *polynomial interpolation*.

1.3.1 Pseudoexpectations.

The primary method to prove lower bounds on Sherali-Adams degree is to build a *pseudoexpectation*. Let $F(x) = C_1 \wedge \dots \wedge C_m$ be a CNF formula on n variables. If F was truly satisfiable, then a way to prove that there is no Sherali-Adams refutation of F is the following. Suppose we had a candidate Sherali-Adams refutation of F , given by

$$\sum_{i=1}^m p_i \overline{C}_i + \sum_{j=1}^n q_j (x_j^2 - x_j) + r_j (1 - x_j - \overline{x}_j) = J + 1,$$

where $p_i, q_j, r_j \in \mathbb{R}[x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n]$ are polynomials, and J is a conical junta term. Now, since F is satisfiable, we can let $\mathcal{D}$ denote the uniform distribution over satisfying assignments of F . Since any sample $x \sim \mathcal{D}$ will evaluate $\overline{C}_i(x) = x_j^2 - x_j = 1 - x_j - \overline{x}_j = 0$, and since $J(x)$ is always non-negative, we can

take the expectation of both sides of the above equation, obtaining:

$$0 = \mathbb{E}_{\mathcal{D}} \left[\sum_{i=1}^m p_i \overline{C}_i + \sum_{j=1}^n q_j (x_j^2 - x_j) + r_j (1 - x_j - \overline{x}_j) \right] = \mathbb{E}_{\mathcal{D}} [J + 1] \geq 1,$$

which is a contradiction.

When F is unsatisfiable, such an argument is impossible. However, a degree- d pseudoexpectation operator $\tilde{\mathbb{E}}$ mimics this argument, in the sense that it “looks like” an expectation over a distribution of satisfying assignments to degree- d polynomials. More formally, a *degree- d pseudoexpectation operator* $\tilde{\mathbb{E}}$ is a linear map from real polynomials to $\mathbb{R}$ satisfying the following properties:

1. $\tilde{\mathbb{E}}[1] = 1$,
2. $\tilde{\mathbb{E}}[C] = 0$ for any conjunction C with $\deg(C) \leq d$ and where $C \Rightarrow \overline{C}_i$, for some clause C_i in F .
3. $\tilde{\mathbb{E}}[C] \geq 0$, for every conjunction C with $\deg(C) \leq d$.
4. $\tilde{\mathbb{E}}[p(x_j^2 - x_j)] = 0$ and $\tilde{\mathbb{E}}[q(1 - x_j - \overline{x}_j)] = 0$, for every polynomial p, q with $\deg(p) \leq d - 2$ and $\deg(q) \leq d - 1$.

If we have such an operator, then applying it to both sides of a degree- d Sherali-Adams refutation will prove a contradiction by the same argument given above. Indeed, by convex programming duality, it turns out that the existence of such a map is *equivalent* to the degree lower bound [FKP19].

As we have already mentioned, there are two major methods for constructing pseudoexpectations in the literature: constructing a *pseudodistribution* [dV07], or by the *pseudo-calibration* method [BHK⁺19], and neither of these methods seem appropriate for constructing a pseudoexpectation for either the CPLS or LOSSY(SOD) problems. We review these two approaches next:

- *Pseudodistributions.* A degree- d pseudodistribution for F is a family of probability distributions $\{\mathcal{D}_S \mid S \subseteq [n], |S| \leq d\}$ such that every distribution $\mathcal{D}_S$ is supported on assignments to the variables $\{x_i : i \in S\}$. To be a pseudodistribution, two properties must be satisfied. First, for every set $|S| \leq d$, $\mathcal{D}_S$ must not be supported on any assignment that falsifies a clause of F . Second, the distributions must have *consistent marginals*: for any sets S, T with $|S|, |T| \leq d$, the two distributions $\mathcal{D}_S$ and $\mathcal{D}_T$ must have the same marginal distribution on the coordinates in $S \cap T$. Intuitively, these constraints enforce that a sample from a pseudo-distribution looks like the degree- d marginals of a true distribution over satisfying assignments to F , and we can recover a pseudoexpectation by, essentially, “taking an expectation” over the pseudodistribution. Constructing a pseudodistribution is a balancing act: we must patch together a collection of distributions on different variables of our target formula, ensuring that no constraint is ever falsified, and also that the marginals of all of the distributions in our family agree.

A useful property which is often exploited in the construction of pseudodistributions is *symmetry*: if our formula F is highly symmetric, meaning it is invariant under the action of a large symmetry group on its input variables, then this can be exploited to prove the consistent marginals property. Roughly speaking, this is because we can focus on constructing a distribution $\mathcal{D}_S$ for a single set of variables indexed by S , and then transport this distribution to other sets by considering the orbit of S under the symmetry — for instance, this can be used to give optimal degree lower bounds on Sherali-Adams refutations of the pigeonhole principle [DMR09]. A second useful property that is often exploited is *expansion*: when considering Sherali-Adams refutations of CSPs, pseudodistributions can be constructed for each subset of variables by taking a *closure* of the constraints touched by the variables, sampling a uniformly random (locally satisfying) assignment, and the expansion guarantees that the marginal property will be satisfied. This can be used to prove strong lower bounds for

CSPs, for example [Gri01, Sch08, GM08]. Neither CPLS_n nor $\text{LOSSY(SOD)}_{n,m}$ satisfy significant amounts of symmetry, and they are both far from being expanding, and so neither of these techniques are effective here. This leaves us with attempting to perform an *ad-hoc* construction, which seems extremely difficult to push through.

- *Pseudo-calibration.* The second major technique is the *pseudo-calibration method*, which has been used to prove lower bounds on a variety of average-case problems in the Sum-of-Squares proof system, an even stronger proof system than Sherali-Adams. However, this method is specially tailored for proving lower bounds for problems with a very particular structure: distinguishing uniformly random distributions of combinatorial configurations from random distributions with some *random planted subconfiguration*. The basic idea of the technique is to construct a pseudoexpectation iteratively, by enforcing that the low-degree moments $\tilde{\mathbb{E}}[M]$ of the distribution match (on average over the uniform distribution G) the true moments of the planted distribution. This is complicated by the fact that low-degree Sherali-Adams or Sums-of-Squares can often prove interesting local facts about samples from a distribution, and hence the moments must be “calibrated” to match what can be deduced in low-degree. This technique was introduced to prove strong lower bounds on the *planted clique* problem in the Sum-of-Squares system, and it has gone on to be used to prove lower bounds on a wide variety of planted problems [BHK⁺19, HKP⁺17, Pot19]. However, neither CPLS nor LOSSY(SOD) follow this average-case planted structure, and so this technique seems just inapplicable from the start.

1.3.2 Detour: $\mathbb{R}$ -Nullstellensatz Lower Bounds for SOD.

As the existing techniques do not seem appropriate for the task, we are left to develop new ones, and due to the close relationship between SOD and our target problems, understanding the complexity of SOD seems like a natural place to start. What is known about the SOD problem with regards to Sherali-Adams, or even other algebraic systems? An (auspicious) interesting result appears in the recent work of Göös et al [GHJ⁺24], who proved the following:

Theorem 1.6 ([GHJ⁺24]). *For all sufficiently large n , any Sherali-Adams refutation of SOD_n with degree $n^{o(1)}$ requires coefficients of magnitude $\exp(n^{\Omega(1)})$.*

Notably, the proof of Theorem 1.6 *does not explicitly construct a pseudo-expectation* that witnesses the lower bound. Rather, it starts with a *weak* lower bound for SOD in a restricted version of Sherali-Adams (the $\mathbb{R}$ -Nullstellensatz system) and then, by a recursive amplification technique, improves the $\mathbb{R}$ -Nullstellensatz lower bound to prove Theorem 1.6. As it was the direct inspiration of our own techniques, we briefly review this lower bound — particularly the lower bound for $\mathbb{R}$ -Nullstellensatz — next.

An $\mathbb{R}$ -Nullstellensatz refutation of a CNF formula $F = C_1 \wedge \dots \wedge C_m$ is a Sherali-Adams proof with no conical junta term. Formally, differentiating it from Equation (1), an $\mathbb{R}$ -Nullstellensatz refutation of F is a polynomial identity of the form

$$\sum_{i=1}^m p_i \overline{C}_i + \sum_{j=1}^n q_j (x_j^2 - x_j) + r_j (1 - x_j - \overline{x}_j) = 1,$$

where $p_i, q_j, r_j \in \mathbb{R}[x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n]$ are polynomials, as before. Just as degree lower bounds for Sherali-Adams are proved using pseudoexpectation operators, degree lower bounds for $\mathbb{R}$ -Nullstellensatz are typically proved by a weakening of pseudoexpectation operators called *d-designs*: a *d*-design D is just a pseudoexpectation which does not need to evaluate conjunctions to non-negative numbers.

The starting point of the proof of Theorem 1.6 is a new argument that $\mathbb{R}$ -Nullstellensatz requires large degree to refute SOD_n . However, they do *not* proceed by explicitly constructing a design for this problem. Rather, they prove this lower bound by a reduction to the *approximate degree lower bound for OR_n*,

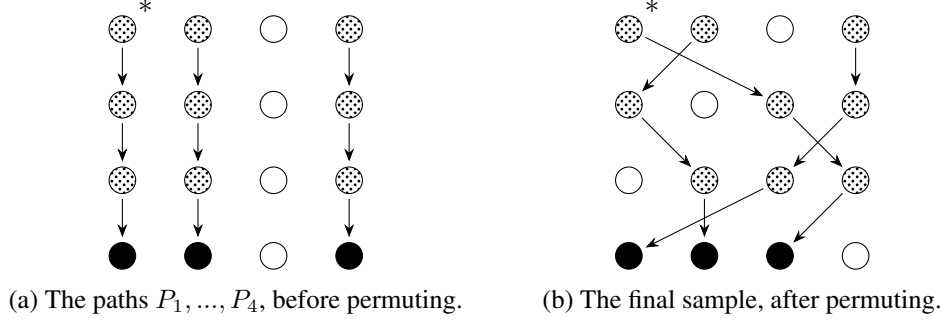


Figure 5: An example of the sampling procedure for $\mathcal{R}_x$, using $x = 101$.

showing that any low-degree $\mathbb{R}$ -Nullstellensatz proof for SOD_n would immediately imply a low-degree approximating polynomial for OR_n , violating the known lower bounds [NS94].

In more detail, this reduction worked as follows. Consider an SOD_n instance on an $n \times n$ grid, as before, and suppose that we have a low-degree $\mathbb{R}$ -Nullstellensatz refutation of SOD_n . We will use this refutation to construct a low-degree approximating polynomial for $\text{OR}_{n-1}(x)$: this is a polynomial $p \in \mathbb{R}[x_1, \dots, x_{n-1}]$ such that for all $x \in \{0, 1\}^{n-1}$, $|p(x) - \text{OR}_{n-1}(x)| \leq 1/3$.

Given any $x \in \{0, 1\}^{n-1}$, consider the following randomized assignment to the variables of SOD_n . For each node $u = (i, j) \in [n]^2$, the successor of u will be $(i + 1, j)$, so the successor pointers split the nodes into n paths $P_1, P_2, \dots, P_n$ walking directly down from the top layer to the bottom layer. The distinguished source $(1, 1)$ must be activated, so we activate all nodes in the path P_1 . Then, for each $i \in [n - 1]$, we activate all nodes in the path P_{i+1} if and only if $x_i = 1$. Finally, we *randomly permute* the pointers in each layer $i \in [n - 1]$, choosing a random permutation $\pi_i : [n] \rightarrow [n]$ and re-routing the pointers accordingly. Following [GHJ⁺24], let $\mathcal{R}_x$ denote the distribution generated from an assignment x in this way. By definition, if $\text{OR}_{n-1}(x) = 0$, then any sample $y \sim \mathcal{R}_x$ has a unique falsified constraint connected to the distinguished start node, while if $\text{OR}_{n-1}(x) = 1$, then any sample $y \sim \mathcal{R}_x$ has at least two falsified constraints. The paper then shows that a low-degree $\mathbb{R}$ -Nullstellensatz proof can be used to actually estimate the number of falsified constraints with low error over the distributions $\mathcal{R}_x$, and this approximation can be converted into a low-degree approximating polynomial for OR_{n-1} .

1.3.3 Polynomial Interpolation for Pseudoexpectations.

It turns out that there is a more direct way to prove this $\mathbb{R}$ -Nullstellensatz degree lower bound for SOD_n . This approach still uses the underlying distributions $\mathcal{R}_x$, but completely avoids the use of a reduction to OR_{n-1} . The main new insight is the use of *polynomial interpolation* to construct a design (cf. Section 4). This new approach is very closely related to the explicit constructions of dual polynomials witnessing large approximate degree for the OR function, as initially performed by Špalek [Spa08] and then refined by Bun and Thaler [BT22].

We use the following version of Lagrange interpolation: if $t_1, \dots, t_{d+1} \in \mathbb{R}$ is any sequence of $d + 1$ distinct *interpolation nodes*, and

$$\ell_i(x) = \prod_{\substack{j \in [d+1] \\ j \neq i}} \frac{x - t_j}{t_i - t_j},$$

then for any degree- d univariate polynomial $p(z)$, $p(x) = \sum_{i=1}^{d+1} \ell_i(x) p(t_i)$. Let's again consider the SOD_n problem, and for each $1 \leq k \leq n$, consider the distribution $\mathcal{D}_k$ defined as follows: sample a uniformly random assignment $x \in \{0, 1\}^{n-1}$ with $|x| = k - 1$, and output a sample from $\mathcal{R}_x$. Hence, any sample from

$\mathcal{D}_k$ looks like k uniformly random active paths walking from the top row to the bottom row. We ask: how does an arbitrary low-degree monomial m over the variables of SOD_n evaluate under $\mathcal{D}_k$ on average?

Well, by the symmetry in the distribution $\mathcal{D}_k$, it turns out that $\mathbb{E}_{\mathcal{D}_k}[m]$ actually has quite a simple form: there is a *univariate polynomial* $Q_m(z)$ with $\deg(Q_m) \leq \deg(m)$ such that for all k , $Q_m(k) = \mathbb{E}_{\mathcal{D}_k}[m]$. This claim may seem surprising at first, but it is not so, as can be seen by evaluating a few simple examples. For example, suppose that $m = \prod_{j=2}^{t+1} a_{1,j}$ is a product of t activity bits on the top row (note we have left out $a_{1,1}$, since it is always active), with $t \leq k$. In order for all of these bits to be active, they must have been chosen to be active under $\mathcal{D}_k$, and so

$$\mathbb{E}_{\mathcal{D}_k}[m] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j=2}^{t+1} a_{1,j} \right] = \frac{\binom{n-t-1}{k-t-1}}{\binom{n-1}{k-1}} = \frac{(k-1)_t}{(n-1)_t},$$

where we have used the falling factorial notation $(a)_b = a(a-1)\cdots(a-b+1)$; defining $Q_m(z) = (z-1)_t / (n-1)_t$ yields our univariate polynomial representation. In turn, this implies that every low-degree polynomial q over the variables of SOD_n can have its expectations encoded by a low-degree univariate polynomial $Q_q(z)$, just by taking an appropriate linear combination of the Q_m polynomials.

Now, in any sample $y \sim \mathcal{D}_k$, the only falsified constraints of SOD_n are activity constraints on the bottom row. By adapting the techniques of [GHJ⁺24], we can show that if C is a conjunction witnessing one of the falsified constraints, then the corresponding polynomial $Q_C(z)$ satisfies $Q_C(0) = 0$. Lagrange interpolation now suggests a candidate for a d -design. Choose the interpolation nodes $t_i = i$ for $i = 1, \dots, n$, and define the signed measure

$$\mu(x) := \sum_{k=1}^n \ell_k(0) \mathcal{D}_k(x),$$

where $\mathcal{D}_k(x)$ is the measure of x under the distribution $\mathcal{D}_k$. For any polynomial q over the variables of SOD_n , define the candidate design

$$D[q] := \sum_x \mu(x) q(x) = \sum_{k=1}^n \ell_k(0) \mathbb{E}_{\mathcal{D}_k}[q(x)].$$

This candidate is linear by definition, and $D[1] = 1$ since we can just interpolate the constant-1 polynomial. This means we just need to show it evaluates to 0 on conjunctions that witness a violated axiom of SOD_n under the distributions. But, by the claim above, for any polynomial q of degree $\leq n-1$ we can find a univariate polynomial Q_q such that $Q_q(z) = \mathbb{E}_{\mathcal{D}_k}[q]$, which implies (by interpolation) that

$$D[q] = Q_q(0).$$

Hence if a conjunction C witnesses a falsified constraint (and so is a multiple of an axiom), we have $D[C] = Q_C(0) = 0$!

1.3.4 Polynomial Interpolation for CPLS and LOSSY(SOD).

Our constructions for CPLS and LOSSY(SOD) directly build off of the above construction by “embedding” the Lagrange-interpolated measure μ for SOD_n in specific ways. However, since we are aiming for Sherali-Adams degree lower bounds, and not just $\mathbb{R}$ -Nullstellensatz lower bounds, special care must be taken to show that the result is a proper pseudoexpectation, which entails showing that the candidate pseudoexpectations evaluate all low-degree conjunctions to non-negative numbers. It is important to note that even though the distributions $\mathcal{D}_k(x)$ are always non-negative, the weights $\ell_k(0)$ will be negative for certain values of k , and so $\mu(x)$ can evaluate to a negative number. Indeed, considering the polynomial $Q_m(z)$ defined

above, we already see that $Q_m(0) < 0$ when t is odd, and so the design D we described above is not a pseudoexpectation for SOD_n , as we should expect.

To prove positivity, we take special advantage of the fact that CPLS and LOSSY(SOD) are “obfuscated” versions of SOD in different ways, either through the use of colours or the use of an outer LOSSY instance. We briefly review what is needed for these extensions, and refer the reader to [Part II](#) and [Part III](#) for the complete constructions.

CPLS. For CPLS, interpolating a natural generalization of the SOD distributions is sufficient. In $\mathcal{D}_k$, we first sample the layer matchings as before, and choose k of the resulting paths to be colourless, always including the path from the distinguished source. Each of the remaining $n - k$ paths receives a single independently and uniformly random colour, which is present at every node along that path. The only violated constraints are then colour creation constraints at the bottoms of the colourless paths. As in the SOD construction, the expectations of low-degree conjunctions can be represented by low-degree polynomials in k , and the polynomials corresponding to witnessing conjunctions vanish at zero. Thus, interpolation again gives a design.

The additional ingredient is to show that this design is non-negative on low-degree conjunctions, and here the random colours play a crucial role. After fixing the paths, any positive colour literal forces a path to carry the specified colour. On a path where the conjunction only asserts that certain colours are absent, each excluded colour rules out just one of the n possible choices for the path’s colour. If there are r such colours, then averaging over the random colour contributes a factor of $1 - (r/n)X$ to the pseudoexpectation value, where X is a random variable indicating that the path is coloured. Since r is bounded by the degree of the conjunction, if the degree is $\ll n$, then this will turn out to be a non-negative contribution.

LOSSY(SOD). Our construction for $\text{LOSSY(SOD)}_{n,m}$ is more involved. We start again with the construction of the measure μ used to prove a degree lower bound for SOD_m described above. Recall that

$$\mu(x) := \sum_{k=1}^m \ell_k(0) \mathcal{D}_k(x),$$

where we have interpolated over the nodes $t_k = k$ for $k = 1, \dots, m$. In our construction for LOSSY(SOD) we will do the following. We split the plaintexts into two groups: for the first group, we choose a random matching of $n/2$ plaintexts to codes, and assign the code pointers to recover exactly these $n/2$ plaintexts. The second group of $n/2$ plaintexts are assigned to uniformly random codes. Since each of the $n/2$ plaintexts in the second group will contribute a violated constraint, on each of these plaintexts, we embed a copy of μ on the corresponding SOD instances. Thanks to our earlier interpolation argument, embedding μ here will successfully “hide” the violated constraints of SOD from the design.

However, $\mu(x)$ can be negative, and so there is no guarantee that the pseudoexpectation values on low-degree conjunctions will be non-negative. This leads to the next idea: we *split* $\mu(x)$ into its positive and negative contributions:

$$\mu^+(x) := \sum_{k: \ell_k(0) > 0} \ell_k(0) \mathcal{D}_k(x), \quad \mu^-(x) := \sum_{k: \ell_k(0) < 0} |\ell_k(0)| \mathcal{D}_k(x),$$

so that $\mu(x) = \mu^+(x) - \mu^-(x)$. If the pseudoexpectation value of $\mu(x)$ is negative on one of the embedded SOD instances, this is entirely due to its contribution from $\mu^-(x)$. To make up for this, we will renormalize the values of $\mu^-(x)$ so that it sums to 1, and then embed the renormalized $\mu^-(x)$ on each of the SOD instances corresponding to plaintexts from the first matched group. Since the plaintexts in this group are already matched, there will be no solutions witnessed here, and so the positive contribution of $\mu^-(x)$ will cover the negative contribution introduced by $\mu(x)$ on the second group of plaintexts.

Unfortunately, this construction is *also* not entirely sufficient. A significant issue is that the interpolation weights $\ell_k(0)$ can be very large in magnitude on the interpolation nodes $k = 1, 2, \dots, m$:

$$\ell_k(0) = (-1)^{k-1} \binom{m}{k}.$$

This means that the positive weight contributed from the renormalized copies of $\mu^-(x)$ on the first group of plaintexts will not make up for the negative weight contributed from $\mu(x)$ on the second group.

To fix this issue, we make a better choice of interpolation nodes, and this choice is directly inspired by the construction of the dual polynomial witnessing that OR has large approximate degree [Spa08,BT22]. Instead of choosing the interpolation nodes $1, 2, \dots, m$ in the construction of μ , we will choose nodes so that $t_0 = 1$ and $t_k \approx O(dk^2)$, where d is a parameter to be chosen later. The quadratic spacing between nodes imitates the extremal points of *Chebyshev polynomials*, as explained by Bun and Thaler [BT22, Section 6.1.1]. For our purposes, choosing these nodes allows us to bound $|\ell_k(0)| \leq O(1/dk^2)$, and so the total contribution from the negative coefficients of $\mu(x)$ will be at most $O(1/d)$. This negative mass is now small enough to be made up for by the renormalized $\mu^-(x)$ on each of the first group of plaintexts, and setting $d = \Theta(\min\{n, m^{1/3}\})$ yields a maximal degree lower bound for SA.

1.4 Related Work

1.4.1 Algebraic Proof Systems and Bounded-Depth Frege.

There has been a significant amount of work invested by researchers to understand the precise relationship between bounded-depth Frege and other proof systems. A strong exponential separation between Resolution and $\text{Res}(2)$ was first presented by Atserias and Bonet [AB04], building on a weaker separation due to Atserias, Bonet, and Esteban [ABE02]. Atserias and Bonet used the reflection principle for Resolution to exhibit the separation, which was later shown to be equivalently captured by the CPLS principles by Krajíček, Skelley, and Thapen [KST07]. Later work by Segerlind, Buss, and Impagliazzo showed that $\text{Res}(k)$ is exponentially weaker than $\text{Res}(k+1)$ for all constant k [SBI04].

For algebraic and semi-algebraic proof systems, it follows from work of Galesi and Lauria [GL10b] that the *Polynomial Calculus over Resolution* (PCR) proof system is exponentially weaker than $\text{Res}(2)$. Their separation uses an $\oplus$ -composed version of the *graph ordering principle*. Such an approach would also work, in principle, for Sherali-Adams, but to our knowledge, there do not exist strong Sherali-Adams degree lower bounds for the graph ordering principle (see [dRLNS21] for a discussion). Another well-studied semi-algebraic system for which we have strong lower bounds is the *Cutting Planes* proof system. By the work of [GGKS18], one can give exponential separations between Cutting Planes and the $\text{Res}(\log)$ proof system by lifting an equivalent separation for Resolution.

A work spiritually related to our own is Bonacina and Bonet [BB25], which explicitly sets out to compare $\mathbb{Z}$ -Nullstellensatz, Sherali-Adams, and bounded-depth Frege. The explicit goal of their article was to determine the minimal axioms that one could add to bounded-depth Frege in order to efficiently simulate $\mathbb{Z}$ -Nullstellensatz and Sherali-Adams, respectively. Towards this goal, they introduced a new *levelled* form of the pigeonhole principle, and showed that (in our language) $\text{Res}(n)$ equipped with a levelled pigeonhole principle can simulate Sherali-Adams, and with a *onto* version of the levelled pigeonhole principle it can simulate $\mathbb{Z}$ -Nullstellensatz. However, they do not prove any separations between Sherali-Adams and $\text{Res}(2)$.

The most well-studied strengthening of Sherali-Adams is the *Sum-of-Squares* proof system, and we note that understanding the precise relationship between Sum-of-Squares and bounded-depth Frege is completely open. As far as we are aware, it is consistent with current knowledge that Sum-of-Squares can polynomially simulate all of bounded-depth Frege.

1.4.2 Automatability of Proof Systems.

It is important to note that when we refer to the non-automatability of weak proof systems, we are always referring to automatability with respect to the *size* of the proofs. When measured with respect to other measures — most notably their width (when considering Resolution) or their degree (when considering systems like Nullstellensatz, Polynomial Calculus, or Sherali-Adams) — some weak proof systems actually *do* admit efficient automating algorithms in some regimes.

As we have discussed above, perhaps the first strong result on the non-automatability of weak proof systems was due to Alekhnovich and Razborov [AR08], who showed that Resolution is not automatable assuming a conjecture in parametrized complexity. Their work was extended to Polynomial Calculus and Polynomial Calculus over Resolution by Galesi and Lauria [GL10a]. Razborov and Alekhnovich’s proof was later given a streamlined argument by Mertz, Pitassi, and Wei [MPW19].

After Atserias and Müller’s breakthrough result on the NP-Hardness of automating Resolution discussed above [AM20], there was an avalanche of non-automatability results proved for weak proof systems in subsequent years. A paper by de Rezende et. al. proved that Nullstellensatz and Polynomial Calculus are NP-Hard to automate [dRGN⁺21], and the original version of [dRGN⁺21] claimed non-automatability of Sherali-Adams proofs as well, but this result was later retracted. This is because the arguments of [dRGN⁺21] were based on lower bounds for the LOSSY principle in Nullstellensatz and Polynomial Calculus, and, as we remarked above, this principle is known to be *easy* for Sherali-Adams. Göös et al showed that Cutting Planes was NP-Hard to automate [GKMP20], and then Garlík [Gar24] showed that the $\text{Res}(k)$ proof systems are NP-Hard to automate. All of these arguments proved the separation by using the $\text{REF}_s(F)$ formula, sometimes with an additional “lifting” argument to fortify the principle against stronger proof systems. Recent work of Arteche, Atserias, de Rezende, and Khaniki formalized the Atserias-Müller argument in the bounded arithmetic theory PV, which gave a number of interesting consequences [AAdRK25]. A notable consequence is a *search version* of the Atserias-Müller reduction: there is a polynomial-time algorithm which, given a polynomial-length Resolution proof of $\text{REF}_{n^{O(1)}}(F)$, will output a satisfying assignment to the underlying formula F .

Papamakarios showed that depth- d Frege is hard to automate for all depths d [Pap24], however his argument does not apply to automating on CNF formulas — this extension seems closely related to proving a hierarchy theorem for bounded-depth Frege, which is a notorious open problem in proof complexity. Unlike the case of Resolution, *tree-like* Resolution has long been known quasi-polynomial time automatable by a simple dynamic programming argument [BP96]. de Rezende [dR21] showed that this quasi-polynomial was necessary (under rETH ⁶) and this result was strengthened to Tree-like $\text{Res}(k)$ by Carenini and de Rezende [CdR25].

Atserias and Bonnet generalized automatability to the notion of *weak automatability* [AB04], where a proof system P is *weakly automatable* if there is a proof system Q and an algorithm which produces a Q -proof of a given tautology in time polynomial in its smallest P -proof. If P is weakly automatable, and closed under partial restrictions, then P is known to have feasible interpolation [BPR00]; this can be used to show that many strong proof systems are *not* weakly automatable. For example, under cryptographic assumptions, it is known that Extended Frege [KP98], TC^0 -Frege [BPR00], and even bounded-depth Frege (for sufficiently large depths) [BDG⁺04], are not weakly automatable. However, non-weak-automatability results are not known for any weak proof system: even proving that Resolution is not weakly automatable is an outstanding open problem.

⁶The published version of [dR21] states that ETH is sufficient, but an erratum on the [author’s website](#) indicates that this is an error.

1.4.3 Sherali-Adams Lower Bound Techniques.

Pseudodistributions. The interpretation of Sherali-Adams by pseudodistributions goes back at least to the work of Fernandez de la Vega and Kenyon-Mathieu [dIVK07]. The equivalent formulation using pseudoexpectations appears, for example, in the work of Chan, Lee, Raghavendra, and Steurer [CLRS13], and is closely related to the earlier use of designs for Nullstellensatz lower bounds [Bus97]. We refer to [FKP19] for a detailed exposition of these viewpoints and their relationship to semi-algebraic proof systems.

As we have discussed above, a central difficulty in constructing pseudodistributions is ensuring that the local distributions have consistent marginals. Symmetry can help to establish this consistency, as in lower bounds for the pigeonhole and least number principles [DMR09]. For constraint satisfaction problems, another useful approach is to extend each small set of variables to a suitable *closure*, sample an assignment satisfying the constraints in this closure, and then restrict back to the original variables. Expansion properties can then be used to show that these distributions agree on their common marginals. This approach is developed in work of Georgiou and Magen [GM08] and Georgiou, Magen, and Tulsiani [GMT09]; related constructions also underlie strong Sum-of-Squares lower bounds for random CSPs [Sch08, KMOW17].

Pseudocalibration. A different approach is the *pseudo-calibration* method introduced by Barak et al. in their work on the planted clique problem [BHK⁺19]. Here, a planted distribution gives candidate moments, which are then “calibrated” when tested against low-degree functions of the input instance. This approach has been particularly successful for average-case Sum-of-Squares lower bounds [HKP⁺17]. In the Sherali-Adams setting, de Rezende, Potechin, and Risse [dRPR23] use a construction inspired by pseudo-calibration to prove lower bounds for refuting large cliques in random graphs. Their argument uses a *pseudo-measure* satisfying approximate versions of the pseudoexpectation conditions, and yields lower bounds on the sum of the absolute values of the coefficients in a refutation. Consequently, their size lower bounds apply to Sherali-Adams with polynomially bounded coefficients, including the unary-coefficient version of the system.

Good Stories. The work most closely related to our use of interpolation is Potechin’s *Good Story* framework for Sum-of-Squares lower bounds [Pot19]. This framework applies to symmetric problems whose unsatisfiability arises from integrality arguments, such as the Knapsack formula, or the existence of a perfect matching on an odd number of vertices. Potechin shows that for certain symmetric problems, the candidate pseudocalibration moments are rational functions of the problem parameters and, most importantly, they *agree* with the moments on nearby satisfiable instances with integral problem parameters. From here, he uses polynomial interpolation to extend these moments to rational problem parameters, where no satisfying assignment exists. Thus, polynomial interpolation already plays an explicit role in constructing pseudoexpectations in this earlier work.

Our construction has a different starting point than Potechin’s. We fix an unsatisfiable instance, and begin with genuine distributions indexed by an integer parameter measuring the number of violated constraints, such as the number of active paths in SOD. We show that the expectations of low-degree witnessing conjunctions agree, at the interpolation nodes, with polynomials in this parameter that vanish at zero. By interpolation, when we evaluate a witnessing conjunction on this interpolated measure, we recover the polynomial’s value at 0, and hence the design vanishes. This first gives a design for SOD, and obtaining pseudoexpectations for CPLS and LOSSY(SOD) requires additional arguments exploiting how these formulas obfuscate the underlying SOD instance. In particular, interpolation alone does not guarantee non-negativity, and the choice of interpolation nodes is an important part of our construction for LOSSY(SOD) (discussed above). We also show in Section 3 that every bounded-degree Sherali-Adams pseudoexpectation can be obtained by interpolating genuine probability distributions, giving a general connection between this approach and constructions using pseudodistributions.

1.5 Organization

We begin by recalling the necessary preliminaries on algebraic proof systems in [Section 2](#). After this, the paper is divided into three parts. In [Part I](#), we introduce the Lagrange interpolation technique, and prove that it is a universal technique for constructing pseudodistributions and pseudoexpectations. We then use interpolation to give an explicit construction of an $\mathbb{R}$ -Nullstellensatz degree lower bound for SOD_n . In [Part II](#), we prove the CPLS_n degree lower bound and the subsequent separation of Sherali-Adams from $\text{Res}(2)$. Finally, in [Part III](#), we prove the degree lower bound for $\text{LOSSY}(\text{SOD})$, and the non-automatability of Sherali-Adams. The reader should first read [Part I](#), as the ideas and techniques introduced are used in both [Part II](#) and [Part III](#). [Part II](#) and [Part III](#) are independent, and can be read in any order.

1.6 AI Disclosure

The results in [Part I](#) and [Part II](#) — the alternate proof of $\text{NS}_{\mathbb{R}}$ degree lower bounds for SOD_n from [\[GHJ⁺24\]](#), the degree lower bounds for CPLS_n , and the resulting separation between SA and $\text{Res}(2)$ — were discovered by the authors without AI assistance. The results in [Part III](#) were discovered by the authors through an interaction with GPT-6 Astra. The authors provided GPT with the interpolation construction for SOD and CPLS , and outlined the proof approach for extending the SOD lower bound to a degree lower bound for $\text{LOSSY}(\text{SOD})_{n,m}$ based on “splitting” the positive and negative contributions of $\mu(x)$. The authors then refined and simplified the produced construction. The writing of the paper was performed completely by the authors, with AI used to find mistakes, copyedit, and for literature search.

2 Algebraic Proof System Preliminaries

Throughout this paper, we will be considering the real algebraic proof systems $\mathbb{R}$ -Nullstellensatz and Sherali-Adams. In this section, we formally introduce these systems and various complexity measures for them.

2.1 $\mathbb{R}$ -Nullstellensatz and Sherali-Adams.

Let $x_1, \dots, x_n$ be a set of variables, intended to be thought of as being boolean, and let $\bar{x}_1, \dots, \bar{x}_n$ be a set of *dual variables*, intended to represent the boolean negations of the x_i variables. We work in the ring of polynomials $\mathbb{R}[x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n]$. To enforce booleanity and complementarity on the variables, we work modulo the ideal generated by

$$\{x_i^2 - x_i, x_i + \bar{x}_i - 1\}_{i=1, \dots, n}$$

A *conjunction* C is a conjunction of boolean literals over $x_1, \dots, x_n$. It will be particularly convenient to think of a conjunction C as a multilinear monomial over the variables $x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n$. In general, if $C = \bigwedge_{i \in P} x_i \wedge \bigwedge_{j \in N} \neg x_j$ is a conjunction, then we encode C as a multilinear polynomial by $\prod_{i \in P} x_i \prod_{j \in N} \bar{x}_j$. We will abuse notation and use “conjunction” to mean both the boolean conjunction and its multilinear representation. For instance, we can encode the conjunction $x \wedge \bar{y} \wedge z$ as the polynomial $x\bar{y}z$, which takes the same values as the conjunction over $\{0, 1\}$. The *degree* of a conjunction C , denoted $\deg(C)$, is the number of literals occurring in it, or equivalently its degree as a polynomial. If $C = \bigvee_i x_i \vee \bigvee_j \neg x_j$ is a clause, we let $\bar{C} = \bigwedge_i \neg x_i \wedge \bigwedge_j x_j$ denote the conjunction obtained by negating C and applying De Morgan’s law.

The first proof system we consider in this paper is the $\mathbb{R}$ -Nullstellensatz system. In proof complexity, the Nullstellensatz system was first considered by Beame et al [\[BIK⁺96\]](#).

Definition 2.1. Let $F(x) = C_1 \wedge \dots \wedge C_m$ be a CNF formula over variables $x_1, \dots, x_n$. An $\mathbb{R}$ -Nullstellensatz refutation of F is a sequence of multilinear polynomials $(p_i)_{i \in [m]}, (q_j)_{j \in [n]}, (r_j)_{j \in [n]}$, where $p_i, q_j, r_j \in \mathbb{R}[x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n]$, satisfying the identity

$$\sum_{i=1}^m p_i \bar{C}_i + \sum_{j=1}^n q_j (x_j^2 - x_j) + r_j (1 - x_j - \bar{x}_j) = 1.$$

The *degree* of the refutation is the maximum degree of any monomial occurring in the proof after expansion, but before cancellations; equivalently the degree is

$$\max \{ \deg(p_i \bar{C}_i) \}_{i \in [m]} \cup \{ \deg(q_j (x_j^2 - x_j)) \}_{j \in [n]} \cup \{ \deg(r_j (1 - x_j - \bar{x}_j)) \}_{j \in [n]}.$$

We let $\text{NS}_{\mathbb{R}}(F)$ denote the minimum degree of any Nullstellensatz refutation of F .

The Sherali-Adams system generalizes the $\mathbb{R}$ -Nullstellensatz system by introducing a *conical junta* term, and was originally introduced by Sherali and Adams [SA90] in the context of linear programming hierarchies. It was first noted as a propositional proof system by Dantchev [Dan07]. The algebraic presentation of Sherali-Adams below seemingly first appeared in work of Grigoriev, Hirsch, and Pasechnik [GHP01] under the name “static LS $^\infty$ ”.

Definition 2.2. A *conical junta* $J = \sum_i \lambda_i C_i$ is a non-negative combination of conjunctions. The *degree* of J , denoted $\deg(J)$, is the maximum degree of any conjunction C_i in the sum.

Definition 2.3. Let $F(x) = C_1 \wedge \dots \wedge C_m$ be a CNF formula on n variables. A *Sherali-Adams refutation*, or SA refutation, is given by polynomials

$$p_1, \dots, p_m, q_1, \dots, q_n, r_1, \dots, r_n \in \mathbb{R}[x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n],$$

and a conical junta J , such that

$$\sum_{i=1}^m p_i \bar{C}_i + \sum_{i=1}^n q_i (x_i^2 - x_i) + \sum_{i=1}^n r_i (x_i + \bar{x}_i - 1) + J = -1.$$

Just as for Nullstellensatz, the *degree* of the refutation is

$$\max \{ \deg(p_i \bar{C}_i) \}_{i \in [m]} \cup \{ \deg(q_i (x_i^2 - x_i)) \}_{i \in [n]} \cup \{ \deg(r_i (x_i + \bar{x}_i - 1)) \}_{i \in [n]} \cup \{ \deg(J) \}.$$

The *size* of the refutation is the maximum number of monomials produced by expanding all polynomials in the expression above before cancellations. We let $\text{SA}(F)$ denote the minimum degree of any Sherali-Adams refutation of F .

2.2 Designs and Pseudoexpectations for Degree Lower Bounds

To prove degree lower bounds for $\mathbb{R}$ -Nullstellensatz and Sherali-Adams, we use *designs* [Bus97] and *pseudoexpectations* [dIVK07, CLRS13], respectively. If C, D are consistent conjunctions, then we say D *witnesses* C if $D \Rightarrow C$. If C is a clause, then D *witnesses the falsity of* C if D witnesses the conjunction $\bar{C}$.

Definition 2.4. Let $\mathcal{P}_d(n) := \mathbb{R}[x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n]_{\leq d}$ denote the collection of all $\mathbb{R}$ -coefficient polynomials over the variables $x_1, x_2, \dots, x_n$ with degree at most d . A *degree- d $\mathbb{R}$ -design* is a function $D : \mathcal{P}_d(n) \rightarrow \mathbb{R}$ satisfying the following properties:

- *Normalization.* $D[1] = 1$.
- *Linearity.* $D[\alpha p + \beta q] = \alpha D[p] + \beta D[q]$ for all $\alpha, \beta \in \mathbb{R}$ and all $p, q \in \mathcal{P}_d(n)$.
- *Booleanity.* $D[p(x_j^2 - x_j)] = 0$ for all $p \in \mathcal{P}_{d-2}(n)$ and all $j \in [n]$.
- *Complementarity.* $D[p(1 - x_j - \bar{x}_j)] = 0$ for all $p \in \mathcal{P}_{d-1}(n)$ and all $j \in [n]$.

Let $F(x) = C_1 \wedge \dots \wedge C_m$ be any CNF formula. We say that D is a *design for F* if, in addition to the above properties, it satisfies the following additional property:

- *F -Nonwitnessing.* $D[C] = 0$, for every $i \in [m]$ and every conjunction $C \Rightarrow \bar{C}_i$ with $\deg(C) \leq d$.

A generalization of a d -design is given by a *pseudoexpectation*, which characterizes Sherali-Adams degree lower bounds. The only difference is that the function is required to be non-negative on all conjunctions.

Definition 2.5. A *degree- d pseudoexpectation* $\tilde{\mathbb{E}}$ is a degree- d $\mathbb{R}$ -design which also satisfies

- *Nonnegativity.* $\tilde{\mathbb{E}}[C] \geq 0$ for every conjunction $C \in \mathcal{P}_d(n)$ with $\deg(C) \leq d$.

Lemma 2.6. If F admits a degree- d , non-witnessing design, then $\text{NS}_{\mathbb{R}}(F) > d$. If F admits a degree- d non-witnessing pseudoexpectation, then $\text{SA}(F) > d$.

Proof. Let $F = C_1 \wedge \dots \wedge C_m$ be a CNF formula, and let D be a degree- d design for F . Suppose we have a degree- d $\mathbb{R}$ -Nullstellensatz refutation of F , given by

$$\sum_{i=1}^m p_i \bar{C}_i + \sum_{j=1}^n q_j (x_j^2 - x_j) + r_j (1 - x_j - \bar{x}_j) = 1.$$

Applying D to both sides and applying linearity yields

$$0 = \sum_{i=1}^m D[p_i \bar{C}_i] + \sum_{j=1}^n D[q_j (x_j^2 - x_j)] + D[r_j (1 - x_j - \bar{x}_j)] = D[1] = 1,$$

a contradiction. If D is additionally a pseudoexpectation, then a similar argument implies that there is no degree- d Sherali-Adams refutation. $\square$

The standard method to construct pseudoexpectation operators is through the use of *pseudodistributions*, the definition of which we recall next.

Definition 2.7. Let $x_1, \dots, x_n$ be a set of boolean variables, and let d be a positive integer. A *degree- d pseudodistribution* over these variables is a family of probability distributions

$$\mathcal{D} = \{\mathcal{D}_S : S \subseteq [n], |S| \leq d\},$$

such that the following properties hold:

- For each set $S \subseteq [n]$, $|S| \leq d$, $\mathcal{D}_S$ is supported on $\{0, 1\}^S$, interpreted as boolean assignments to variables in $\{x_i : i \in S\}$.
- For each $S, T \subseteq [n]$, $|S|, |T| \leq d$, we have $\mathcal{D}_S^{S \cap T} = \mathcal{D}_T^{S \cap T} = \mathcal{D}_{S \cap T}$, where $\mathcal{D}_A^B$ for $B \subseteq A$ is the marginal distribution of $\mathcal{D}_A$ to variables indexed by B .

Let F be any unsatisfiable CNF formula. Then $\mathcal{D}$ is a *pseudodistribution for F* if for every degree $\leq d$ conjunction C witnessing the falsity of a clause of F , $\mathcal{D}_{\text{vars}(C)}$ assigns probability 0 to any assignment satisfying C .

In other words, a pseudodistribution is an object that “looks like” a probability distribution to an external adversary that can only investigate marginals of up to d bits, but, the distribution is not supported on any assignment that witnesses a false clause of F . The following lemma is standard.

Lemma 2.8 ([FKP19]). *Let F be a CNF formula. Suppose $\mathcal{D}$ is a degree- d pseudodistribution over $x_1, \dots, x_n$ for F . Consider the operator $\tilde{\mathbb{E}} : \mathcal{P}_d(n) \rightarrow \mathbb{R}$ defined by*

$$\tilde{\mathbb{E}} \left[\prod_{i \in S} x_i \prod_{j \in T} \bar{x}_j \right] = \Pr_{x \sim \mathcal{D}_{S \cup T}} [\forall i \in S : x_i = 1 \text{ and } \forall j \in T : x_j = 0].$$

We can extend $\tilde{\mathbb{E}}$ to multilinear polynomials by linearity, and then to all of $\mathcal{P}_d(n)$ by defining $\tilde{\mathbb{E}}[p] = \tilde{\mathbb{E}}[q]$ whenever $p \equiv q \pmod{\langle x_i^2 - x_i, 1 - x_i - \bar{x}_i \rangle_{i=1}^n}$. Then $\tilde{\mathbb{E}}$ is a degree- d pseudoexpectation operator.

Conversely, if $\tilde{\mathbb{E}}$ is a degree- d pseudoexpectation operator, then for any $S \subseteq [n]$ with $|S| \leq d$ define the distribution $\mathcal{D}_S$ as follows: for each $a \in \{0, 1\}^S$, define

$$\Pr_{x \sim \mathcal{D}_S} [\forall i \in S : x_i = a_i] = \tilde{\mathbb{E}} \left[\prod_{i: a_i=1} x_i \prod_{j: a_j=0} (1 - x_j) \right]$$

Then the collection $\mathcal{D} = (\mathcal{D}_S)_S$ is a degree- d pseudodistribution.

Since we can use [Lemma 2.8](#) to pass back and forth between pseudodistributions and pseudoexpectations, we will refer to the pseudoexpectation $\tilde{\mathbb{E}}$ constructed from the above lemma as the *pseudoexpectation corresponding to the pseudodistribution* and *vice-versa*.

2.3 Indicator Algebras and Block-Degree Lower Bounds

In later sections, we will often be dealing with formulas which have variables which naturally take values in $[n]$, rather than $\{0, 1\}$. We opt to encode such variables in binary: for a positive integer ℓ , if $x \in [2^\ell]$, then we can encode x by ℓ binary variables $x_1 \dots x_\ell$, where $x = j$ if and only if $1 + \sum_{i=1}^{\ell} 2^{i-1} x_i = j$. Unfortunately, this encoding adds an extra (and inessential) layer of complication to our proofs. Instead, we will opt to work in an abstraction, where statements like “ $x = j$ ” become atomic. Such an abstraction was introduced for Resolution and $\text{Res}(k)$ proof systems by Davis and Robere [DR23] under the name “Blockwise Calculus”. Here we will work with a further generalization to handle the algebra of real polynomials.

Definition 2.9. A *multivalued variable* is a pair (x, S) , where S is a finite set. We think of x as taking a value in the range S . For any $j \in S$ we write $\llbracket x = j \rrbracket : S \rightarrow \{0, 1\}$ for the indicator function which is 1 if and only if $x = j$. We call any expression of the form $\llbracket x = j \rrbracket$ for a multivalued variable x an *indicator*. If X is a set of multivalued variables, then $\mathcal{I}(X)$ denotes the set of indicators over variables from X .

We can easily generalize conjunctions to multivalued variables. A *multivalued conjunction* C is any conjunction of indicators

$$C = \prod_i \llbracket x_i = j_i \rrbracket,$$

and a *multivalued clause* is any disjunction of indicator negations $\bigvee_i \llbracket x_i \neq j_i \rrbracket$. If C and D are multivalued conjunctions, then we write $C \Rightarrow D$ if every assignment which satisfies C also satisfies D . The *degree* of a multivalued conjunction is the number of indicators for distinct variables appearing in it. To differentiate indicator degree from regular degree, we write $\text{bdeg}(C)$ to indicate the degree of a multivalued conjunction, and refer to it as *block degree*.

Indicators for multivalued variables admit a natural algebra. For example, if (x, S) is a multivalued variable, then we have

$$\sum_{i \in S} \llbracket x = i \rrbracket = 1,$$

since exactly one indicator must be set to true. Similarly, we have $\llbracket x = i \rrbracket \llbracket x = j \rrbracket = 0$ for any $i \neq j$, and also $\llbracket x = i \rrbracket \llbracket x = i \rrbracket = \llbracket x = i \rrbracket$. As we wish to create designs and pseudoexpectations for polynomials built out of indicators, it will be helpful to build a formal algebra of indicators that satisfies these relations. Formally speaking, in this algebra we will be treating indicators $\mathcal{I}(X)$ as atomic, and we need to quotient by the relations we described above to ensure that everything is correctly represented. First, we introduce the basis polynomials for this algebra, which come from partial assignments.

Definition 2.10. Let $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ be a collection of m multivalued variables. A *partial assignment* to the variables in X is a function σ assigning variables x_i in X to values in S_i or to $*$, meaning they are unset. We let $|\sigma|$ denote the number of variables that σ sets to non- $*$ values. If σ is a partial assignment, then define the indicator conjunction

$$\chi_\sigma = \prod_{\substack{i \in [m] \\ \sigma(x_i) \neq *}} \llbracket x_i = \sigma(x_i) \rrbracket,$$

where if $\sigma(x_i) = *$ for all i then $\chi_\sigma = 1$.

Now we define the algebra properly.

Definition 2.11. Let $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ be a collection of m multivalued variables, and let $\mathcal{I}(X)$ denote the set of indicators from X . Define $\mathcal{A}(X) := \mathbb{R}[\mathcal{I}(X)]/\mathcal{J}$, where $\mathcal{J}$ is the ideal generated by the polynomials

$$\left\{ 1 - \sum_{i \in S_j} \llbracket x_j = i \rrbracket \right\}_{j \in [m]} \cup \{ \llbracket x_j = i \rrbracket \llbracket x_j = i \rrbracket - \llbracket x_j = i \rrbracket \}_{j \in [m], i \in S_j} \cup \{ \llbracket x_j = i \rrbracket \llbracket x_j = k \rrbracket \}_{j \in [m], i \neq k \in S_j}.$$

We refer to $\mathcal{A}(X)$ as the *indicator algebra* of X .

We can now define what it means to have *low degree* in an indicator algebra. Informally, an expression has low-degree if it can be built up by taking linear combinations of low-degree indicator conjunctions. To distinguish polynomials in an indicator algebra from standard polynomials, we will refer to this notion of degree as *block degree*.

Definition 2.12. Let $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ be a collection of m multivalued variables, and let $\mathcal{A}(X)$ denote the indicator algebra of X . For any $d \geq 0$, define

$$\mathcal{A}(X)_{\leq d} := \text{span} \{ \chi_\sigma \mid |\sigma| \leq d \}.$$

If $p \in \mathcal{A}(X)$ is an indicator polynomial, then the *block degree* of p is defined to be

$$\text{bdeg}(p) := \min \{ d : p \in \mathcal{A}(X)_{\leq d} \}.$$

Now we are ready to introduce generalizations of designs and pseudoexpectations to indicator algebras.

Definition 2.13. Let $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ be a collection of m multivalued variables. A *degree- d indicator design* is any map $D : \mathcal{A}(X)_{\leq d} \rightarrow \mathbb{R}$ satisfying the following properties:

- *Normalization.* $D[1] = 1$
- *Linearity.* $D[\alpha p + \beta q] = \alpha D[p] + \beta D[q]$ for all $p, q \in \mathcal{A}_{\leq d}$.

Let $F = C_1 \wedge \dots \wedge C_m$ be a conjunction of clauses built from indicators in $\mathcal{I}(X)$. We say D is *F-nonwitnessing* if the following holds:

- *F-Nonwitnessing.* $D[C] = 0$ for every $i \in [m]$ and every conjunction $C \Rightarrow \overline{C}_i$ with $\text{bdeg}(C) \leq d$.

For Sherali-Adams, the natural generalization of designs must be positive on all conjunctions.

Definition 2.14. Let X be a collection of multivalued variables. A degree- d indicator design $\tilde{\mathbb{E}}$ on X is called an *indicator pseudoexpectation* if the following holds:

- *Nonnegativity.* $\tilde{\mathbb{E}}[C] \geq 0$ for every conjunction C with $\text{bdeg}(C) \leq d$.

The following transfer principle allows us to translate indicator designs and pseudoexpectations to true designs and pseudoexpectations, simply by encoding each of the underlying variables in binary. Formally, suppose that $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ is a collection of multivalued variables where $|S_i| = 2^{w_i}$ for each i , with w_i being a natural number, and write $S_i = \{a_{i,0}, \dots, a_{i,2^{w_i}-1}\}$. Encode each multivalued variable x_i in binary by introducing w_i boolean variables $x_{i,0} \dots x_{i,w_i-1} \in \{0,1\}^{w_i}$, where $x_i = a_{i,j}$ if and only if $\sum_{k=0}^{w_i-1} 2^k x_{i,k} = j$, and let X^{bool} denote the set of all $\sum_{i=1}^m w_i$ boolean variables produced this way. Let $\overline{X}^{bool}$ denote the set of *dual* variables to the variables in X^{bool} . If $\llbracket x_i = j \rrbracket$ is an indicator, then we can translate $\llbracket x_i = j \rrbracket$ to a conjunction on the underlying boolean variables, denoted $\llbracket x_i = j \rrbracket^{bool}$, in the natural way. Similarly, if $C = \prod_i I_i$ is a conjunction of indicators, we let $C^{bool} = \prod_i I_i^{bool}$. If C is a conjunction over the boolean translation of variables, we let $\text{bdeg}(C)$ denote the number of underlying multivalued variables mentioned by C .

Definition 2.15. Let $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ be a collection of multivalued variables where $|S_i| = 2^{w_i}$ for each i . Suppose $D : \mathcal{A}(X)_{\leq d} \rightarrow \mathbb{R}$ is a degree- d indicator design, and define the *booleanization* of D , denoted D^{bool} , as follows.

For each conjunction C over $X^{bool}, \overline{X}^{bool}$ with $\text{bdeg}(C) \leq d$, let $\mathcal{I}_C$ be the set of variables of distinct indicators mentioned by the boolean variables in C . Given an assignment σ to these to these multivalued variables, we write $\sigma(C) = 1$ if the boolean conjunction C , interpreted under the assignment given by σ , evaluates to 1. Then define

$$D^{bool}[C] = \sum_{\substack{\sigma: \text{dom}(\sigma) = \mathcal{I}_C \\ \sigma(C) = 1}} D[\chi_\sigma].$$

extended to all multilinear polynomials by linearity, and to higher-degree polynomials by equivalence modulo the ideal generated by the boolean and complementarity axioms.

Theorem 2.16. Let $X = \{(x_1, S_1), \dots, (x_m, S_m)\}$ is a collection of multivalued variables, where $|S_i| = 2^{w_i}$ for each i . If $D : \mathcal{A}(X)_{\leq d} \rightarrow \mathbb{R}$ is a degree- d indicator design, then D^{bool} is a degree- d design satisfying, for every indicator conjunction C with $\text{bdeg}(C) \leq d$,

$$D[C] = D^{bool}[C^{bool}].$$

Moreover, if $D = \tilde{\mathbb{E}}$ is a pseudoexpectation, then so is $\tilde{\mathbb{E}}^{bool}$.

Proof. We must check that D^{bool} satisfies all the properties of a design. First, we observe that the indicator polynomial 1 translates to the constant polynomial 1. Hence $D^{bool}[1] = D[1] = 1$. Second, for any indicator conjunction C with $\text{bdeg}(C) \leq d$, it is clear that $D[C] = D^{bool}[C^{bool}]$, since there is exactly one assignment σ with satisfying C which assigns only the indicators in C .

We now prove $D^{bool}[C]$ is well-defined for every conjunction C . We prove this by induction on the degree of C . If $\deg(C) = 0$, then $C = 1$, and we have already seen that $D(C) = 1$. By induction, consider a conjunction C with $\deg(C) = i$. If all literals in C are positive, there is nothing to prove, so suppose $C = (1 - x_{i,j})C'$ for some conjunction C' with $\deg(C') = i - 1$. We need to show that

$$D^{bool}[(1 - x_{i,j})C'] = D^{bool}[C'] - D^{bool}[x_{i,j}C'].$$

Let I_C denote the set of indicators mentioned by C . Then, substituting the definition of D^{bool} and rearranging, we must equivalently show

$$\sum_{\substack{\sigma: \text{dom}(\sigma) = I_C \\ \sigma((1-x_{i,j})C') = 1}} D[\chi_\sigma] + \sum_{\substack{\sigma: \text{dom}(\sigma) = I_C \\ \sigma(x_{i,j}C') = 1}} D[\chi_\sigma] = \sum_{\substack{\sigma: \text{dom}(\sigma) = I_{C'} \\ \sigma(C') = 1}} D[\chi_\sigma]$$

This identity is clear. Consider any σ with $\text{dom}(\sigma) = I_{C'}$ and $\sigma(C') = 1$. There are two cases. If the block containing $x_{i,j}$ is in $I_{C'}$, then in any σ either $\sigma(x_{i,j}) = 0$ or $\sigma(x_{i,j}) = 1$, but not both. Hence the terms on the LHS of the equality are the same as the terms on the RHS of the equality. In the other case, the block containing $x_{i,j}$ is not in $I_{C'}$. Then the RHS is the same as summing over all extensions to this block, but since the block is not queried by C' we can use the identity $\sum_a \llbracket x_i = a \rrbracket = 1$ and recover the RHS exactly. Hence the LHS is still equal to the RHS.

From this identity, we observe that if D is non-witnessing for some F , then D^{bool} is non-witnessing for F^{bool} . Moreover, if D is a pseudoexpectation, and so D is nonnegative on all indicator conjunctions, it follows that $D^{bool}[C]$ is non-negative for every conjunction C , since it is a sum of non-negative numbers. This completes the proof. $\square$

Part I

Polynomial Interpolation for Pseudoexpectations

In this part, we introduce our interpolation technique, and use it to prove a Nullstellensatz degree lower bound for the SoD_n formula. This degree lower bound will be important in both [Part II](#) and [Part III](#).

3 Warmup and Completeness of Interpolation

Our new technique for producing designs and pseudoexpectations is the use of *Lagrange Interpolation*. As we have discussed in the introduction, the main insight in using interpolation is that we can interpolate *true distributions* by taking a signed linear combination of them, and produce a *pseudoexpectation*. The particular form of interpolation we use is the following, along with a concrete corollary.

Theorem 3.1. *Let $t_1, \dots, t_{d+1} \in \mathbb{R}$ be any sequence of $d + 1$ distinct real numbers, and define*

$$\ell_i(x) = \prod_{\substack{j \in [d+1] \\ j \neq i}} \frac{x - t_j}{t_i - t_j}.$$

Then for any degree $\leq d$ univariate polynomial $p(z)$ over $\mathbb{R}$ we have

$$p(z) = \sum_{i=1}^{d+1} p(t_i) \ell_i(z).$$

In particular, if $t_k = k$ for $k = 1, 2, \dots, d + 1$, we define

$$\omega_k^{(d+1)} := \ell_k(0) = \prod_{\substack{j \in [d+1] \\ j \neq k}} \frac{-j}{k - j} = (-1)^{k-1} \binom{d+1}{k},$$

then for every degree $\leq d$ univariate polynomial $p(z)$ over $\mathbb{R}$, $p(0) = \sum_{k=1}^{d+1} \omega_k^{(d+1)} p(k)$. Often the value d will be clear from context, so we will suppress the $d + 1$ from $\omega_k^{(d+1)}$.

Proof. For any $i, j = 1, \dots, d + 1$,

$$\ell_i(t_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

This means that $q(z) = p(z) - \sum_i p(t_i) \ell_i(z)$ has degree at most d and is equal to 0 at each of the $d + 1$ points $t_1, \dots, t_{d+1}$. But a non-zero polynomial of degree at most d has at most d distinct roots, and so $q(z)$ must be identically 0. $\square$

Before getting into any details, let us first give a high-level overview of the technique. Suppose that $F(x) = C_1 \wedge \dots \wedge C_m$ is an unsatisfiable k -CNF formula on some set of variables $x \in \{0, 1\}^n$, and furthermore, suppose that we can define a family of distributions $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{d+1}$ supported on $\{0, 1\}^n$ such that for each $k \leq d$, any x in the support of $\mathcal{D}_k$ has at most k falsified clauses when we evaluate it on F . Abusing notation, we write $\mathcal{D}_k(x)$ to denote the mass of x in $\mathcal{D}_k$. Then we can interpolate these distributions by defining

$$\mu(x) := \sum_{k=1}^{d+1} \omega_k \mathcal{D}_k(x),$$

where ω_k is defined in [Theorem 3.1](#). We note that $\sum_{x \in \{0,1\}^n} \mu(x) = 1$, as can be seen by applying the interpolation theorem to the constant polynomial $p(z) = 1$. However, since some of the interpolation weights ω_k are negative, $\mu(x)$ is not necessarily positive for all x . Since μ interpolates several probability distributions, it still makes (formal) sense to evaluate an average over it. We will introduce some notation for it:

Definition 3.2. Suppose S is any finite set, and $\mu : S \rightarrow \mathbb{R}$ is any function satisfying $\sum_{x \in S} \mu(x) = 1$. If $f : S \rightarrow \mathbb{R}$ is any function, define

$$\mathbf{E}_\mu[f] = \sum_{x \in S} f(x) \mu(x).$$

Remark 3.3. We reserve the symbol $\mathbb{E}$ for a true expectation over a probability distribution, and use $\mathbf{E}$ to indicate an “expectation” over a signed measure μ . Of course, if μ is non-negative and $\sum_x \mu(x) = 1$, then $\mathbb{E}_\mu[f] = \mathbf{E}_\mu[f]$ for every f .

Given any μ defined by interpolation, our candidate designs and pseudoexpectations will be

$$\mathbf{E}_\mu[C] = \sum_x C(x) \mu(x) = \sum_{k=1}^{d+1} \omega_k \mathbb{E}_{\mathcal{D}_k}[C].$$

Using interpolation, we automatically obtain that $\mathbf{E}_\mu[1] = 1$, and the candidate is linear by definition. In order to ensure these designs are non-witnessing, we must choose the distributions $\mathcal{D}_k$ so that for every witnessing conjunction C , there is a univariate polynomial $Q_C(z)$ with $\deg(q) \leq d$ such that $Q_C(k) = \mathbb{E}_{\mathcal{D}_k}[C]$ for all k , and $Q_C(0) = 0$. If this is possible, then by interpolation, we immediately get

$$\mathbf{E}_\mu[C] = \sum_{k=1}^{d+1} \omega_k \mathbb{E}_{\mathcal{D}_k}[C] = \sum_{k=1}^{d+1} \omega_k Q_C(k) = Q_C(0) = 0.$$

Intuitively, if $\mathcal{D}_k$ represents a distribution that witnesses some number of false clauses of some CNF formula F , and $\mathbb{E}_{\mathcal{D}_k}[C]$ can be represented by a low-degree polynomial, then interpolation implies that $\mathbf{E}_\mu[C]$ is the evaluation of C on a “distribution” with *no* falsified constraints of F .

3.1 Polynomial Interpolation for the Bit Pigeonhole Principle

We first illustrate this technique using the *bit pigeonhole principle* BPHP. We will use this opportunity to introduce multivalued indicators in our definitions, as they will be heavily used later in the paper.

Definition 3.4. Let n be a positive integer. Let $(x_1, [n]), (x_2, [n]), \dots, (x_{n+1}, [n])$ be a collection of $n+1$ multivalued variables with domain $[n]$. The *bit pigeonhole principle*, BPHP_n , is the formula

$$\text{BPHP}_n = \bigwedge_{1 \leq i < j \leq n+1} \bigwedge_{\alpha \in [n]} \llbracket x_i \neq \alpha \rrbracket \vee \llbracket x_j \neq \alpha \rrbracket.$$

We refer to each index $i \in [n+1]$ as a *pigeon* and each $\alpha \in [n]$ as a *hole*.

Remark 3.5. This formula has been regularly considered in the literature in a non-multivalued encoding. However, if we let $\text{BPHP}_n^{\text{bool}}$ be the boolean translation, as described in [Section 2.3](#), then the result is the same formula as in [\[AMO15\]](#).

We will prove a degree- n Sherali-Adams lower bound using the interpolation method just introduced. For this, we need to introduce true probability distributions $\mathcal{D}_k$ on the entire space of assignments. Informally, $\mathcal{D}_k$ will be obtained by assigning a random set of $n+1-k$ pigeons in a uniformly random matching, and the remaining k pigeons independently and uniformly at random. Hence any sample from $\mathcal{D}_k$ will have at most k “exceptional” pigeons which can cause violated constraints of BPHP_n .

Definition 3.6. For any positive integer $1 \leq k \leq n+1$, define the distribution $\mathcal{D}_k$ on $[n]^{[n+1]}$ as follows. Sample a uniformly random set of k pigeons $S \subseteq [n+1]$, and let $\bar{S} = [n+1] \setminus S$ be the remaining pigeons. Sample a uniformly random matching of the pigeons in $\bar{S}$ to $|\bar{S}|$ holes, and assign each pigeon in S to uniformly random holes. Define the measure

$$\mu(x) := \sum_{k=1}^{n+1} \omega_k \mathcal{D}_k(x)$$

where $\omega_k = (-1)^{k-1} \binom{n+1}{k}$ are the interpolation weights defined in [Theorem 3.1](#).

We claim that the operator $\mathbb{E}_\mu$, where μ is chosen from the previous definition, is a degree- n pseudoexpectation operator for BPHP_n . In fact, we will prove something even stronger. There is a simple pseudoexpectation, known as the *matching pseudoexpectation*, which is already known to give degree- n lower bounds on any Sherali-Adams refutation of BPHP_n .

Definition 3.7. The *matching pseudodistribution* is defined as follows. Let $S \subseteq [n+1]$ be any set of variables of BPHP_n such that $|S| \leq n$. We define $\mathcal{M}_S$ as follows: sample a uniformly random set of holes $H \subseteq [n]$ of size $|S|$, sample a uniformly random matching from S to H , and assign the variables in S according to this matching. Let $\tilde{\mathbb{E}}_{\mathcal{M}}$ denote the corresponding pseudoexpectation operator.

It is easy to verify that for any two sets S and T , the distributions $\mathcal{M}_S$ and $\mathcal{M}_T$ agree on the marginal distribution $\mathcal{M}_{S \cap T}$. Hence, by applying [Lemma 2.8](#), we have a degree- n pseudoexpectation operator for BPHP_n and hence $\text{SA}(\text{BPHP}_n) > n$. We now show that the operator $\mathbb{E}_\mu$ defined by interpolation is *exactly the same* as the matching pseudoexpectation.

Theorem 3.8. Let μ be the signed measure from [Definition 3.6](#). For all conjunctions C with $\text{bdeg}(C) \leq n$, $\tilde{\mathbb{E}}_{\mathcal{M}}[C] = \mathbb{E}_\mu[C]$.

Proof. We show $\mathbb{E}_\mu$ has the same values as the matching pseudoexpectation. To do so, we will first simplify the expression $\mathbb{E}_\mu[C]$, and so we need to consider $\mathbb{E}_{\mathcal{D}_k}[C]$.

Recall a sample from $\mathcal{D}_k$ is obtained by choosing a uniformly random set S of k pigeons, assigning them independently and uniformly to holes, and then assigning all pigeons in $\bar{S}$ to be a uniformly random matching. Given a conjunction C , let $p(C)$ be the set of pigeons mentioned by C , and assume wlog that C mentions at most one indicator for each pigeon. If $T \subseteq p(C)$ let C_T denote the restriction of the conjunction to the pigeons indexed by T .

Let $\binom{[n+1]}{k}$ denote the collection of all size- k subsets of $[n+1]$, and we write $S \sim \binom{[n+1]}{k}$ to mean that S is sampled uniformly at random from this collection. Then

$$\mathbb{E}_{\mathcal{D}_k}[C] = \sum_{T \subseteq p(C)} \Pr_{S \sim \binom{[n+1]}{k}} [S \cap p(C) = T] \frac{1}{n^{|T|}} \mathbb{E}_{\mathcal{M}_{p(C) \setminus T}} [C_{p(C) \setminus T}].$$

This is because to sample from $\mathcal{D}_k$, we first choose a uniformly random set of k pigeons. The pigeons in S are assigned independently and uniformly at random to holes (implying the $1/n^{|T|}$ factor), and the remaining pigeons are assigned as a matching, and hence according to the distribution $\mathcal{M}_{p(C) \setminus T}$.

Let $t = |p(C)|$ and let $m = n + 1$. Then we have

$$\begin{aligned} \Pr_{S \sim \binom{[n+1]}{k}} [S \cap p(C) = T] &= \frac{\binom{m-t}{k-|T|}}{\binom{m}{k}} \\ &= \frac{\prod_{i=0}^{|T|-1} (k-i) \prod_{i=0}^{t-|T|-1} (m-k-i)}{\prod_{i=0}^{t-1} (m-i)} \\ &= \frac{(k)_{|T|} (m-k)_{t-|T|}}{(m)_t}, \end{aligned}$$

where we have used the falling factorial notation $(n)_k := n(n-1) \cdots (n-k+1)$. With this in mind, for $0 \leq a \leq b$, define the polynomials

$$Q_{a,b}(z) := \frac{(z)_a (m-z)_{b-a}}{(m)_b},$$

and note that $Q_{a,b}(z)$ is a degree- b univariate polynomial in z . Then we can define

$$P_C(z) := \sum_{T \subseteq p(C)} \frac{1}{n^{|T|}} \mathbb{E}_{\mathcal{M}_{p(C) \setminus T}} [C_{p(C) \setminus T}] \cdot Q_{|T|,t}(z),$$

which is a degree- t polynomial in z , since the expectation does not depend on z . This implies that for all k , we have $P_C(k) = \mathbb{E}_{\mathcal{D}_k}[C]$. Therefore

$$\mathbb{E}_{\mu}[C] = \sum_{k=1}^{n+1} \omega_k \mathbb{E}_{\mathcal{D}_k}[C] = \sum_{k=1}^{n+1} \omega_k P_C(k) = P_C(0),$$

by interpolation. But

$$P_C(0) = \sum_{T \subseteq p(C)} \frac{1}{n^{|T|}} \mathbb{E}_{\mathcal{M}_{p(C) \setminus T}} [C_{p(C) \setminus T}] \cdot Q_{|T|,t}(0)$$

and $Q_{|T|,t}(0) = 0$ unless $|T| = 0$, at which point

$$Q_{0,t}(0) = \prod_{i=0}^{t-1} \frac{n-i}{n-i} = 1.$$

This means that

$$\mathbb{E}_{\mu}[C] = P_C(0) = \mathbb{E}_{\mathcal{M}_{p(C)}}[C] = \tilde{\mathbb{E}}[C]. \quad \square$$

3.2 Polynomial Interpolation is Complete

In fact, the same argument works for *any* design or pseudoexpectation. In this sense, using interpolation to produce these objects is a complete method. First, we will prove the statement for pseudoexpectations, as there are natural probabilistic interpretations of the objects involved.

Let n be a positive integer, and let $0 \leq d \leq n$. It will be more natural to interpolate at n , rather than at 0, so for $k = 0, 1, \dots, d$, we define the interpolation weights

$$\lambda_k := \prod_{\substack{j=0 \\ j \neq k}}^d \frac{n-j}{k-j}.$$

Theorem 3.9. Let $\tilde{\mathbb{E}} : \mathcal{P}_d(n) \rightarrow \mathbb{R}$ be a degree- d pseudoexpectation, and let $(\mathcal{D}_S)_{|S| \leq d}$ be the corresponding pseudodistribution. For any $k = 0, 1, \dots, d$, define the distribution $\mathcal{D}_k$ over $\{0, 1\}^n$ as follows. First, sample a uniformly random subset $S \subseteq [n]$ with $|S| = k$, and assign the bits in S by sampling from $\mathcal{D}_S$. Sample all remaining bits independently and uniformly at random.

If $\mu(x) := \sum_{k=0}^d \lambda_k \mathcal{D}_k(x)$, then for all degree $\leq d$ conjunctions C ,

$$\mathbb{E}_{\mu}[C] = \tilde{\mathbb{E}}[C].$$

Proof. Let C be a conjunction, and assume that C is non-trivial in the sense that C does not have both a literal and its negation occurring in it. Let $V \subseteq [n]$ denote the variable indices in C , and let C_T denote the restriction of C to variables in T for any $T \subseteq V$. Define $v = |V|$, then

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_k}[C] &= \sum_{T \subseteq V} \Pr_{|S|=k}[S \cap V = T] \cdot \frac{1}{2^{v-|T|}} \tilde{\mathbb{E}}[C_T] \\ &= \sum_{T \subseteq V} \frac{\binom{k}{|T|} \binom{n-k}{v-|T|}}{\binom{n}{v}} \cdot \frac{1}{2^{v-|T|}} \tilde{\mathbb{E}}[C_T]. \end{aligned}$$

As in the previous proof, define $Q_{a,v}(z) = (z)_a (n-z)_{v-a} / (n)_v$, and we can define the polynomial

$$P_C(z) := \sum_{T \subseteq V} \frac{Q_{|T|,n}(z)}{2^{v-|T|}} \tilde{\mathbb{E}}[C_T],$$

so that $P_C(k) = \mathbb{E}_{\mathcal{D}_k}[C]$ for all $k = 0, \dots, d$. Now we have

$$\mathbb{E}_{\mu}[C] = \sum_{k=0}^d \lambda_k \mathbb{E}_{\mathcal{D}_k}[C] = \sum_{k=0}^d \lambda_k P_C(k) = P_C(n),$$

where we have used interpolation. But observe that $Q_{t,n}(n) = 0$ unless $t = v$, at which point $Q_{v,n}(n) = (n)_v / (n)_v = 1$, and hence $P_C(n) = \tilde{\mathbb{E}}[C]$. This completes the proof. $\square$

If we are considering designs instead, we can interpolate over signed measures using the exact same construction as we used with pseudoexpectations above. The proof is identical to the previous theorem, so we omit it:

Theorem 3.10. Let $D : \mathcal{P}_d(n) \rightarrow \mathbb{R}$ be a d -design, and for any subset $S \subseteq [n]$ with $|S| \leq d$ and any $a \in \{0, 1\}^n$, define

$$\mathcal{D}_S(a) := D \left[\prod_{i \in S: a_i=1} x_i \prod_{i \in S: a_i=0} (1-x_i) \right].$$

For each $k = 0, \dots, d$, define the (signed) measure $\mathcal{D}_k : \{0, 1\}^n \rightarrow \mathbb{R}$ by

$$\mathcal{D}_k(a) := \sum_{\substack{S \subseteq [n] \\ |S|=k}} \frac{\mathcal{D}_S(a)}{\binom{n}{k} 2^{n-k}}, \quad \text{for all } a \in \{0, 1\}^n.$$

If $\mu = \sum_{k=0}^d \lambda_k \mathcal{D}_k$, then for every conjunction C with $\deg(C) \leq d$,

$$\mathbb{E}_{\mu}[C] = D[C].$$

4 $\mathbb{R}$ -Nullstellensatz Degree Bounds for SOD

The previous section gave a method for producing pseudoexpectations by Lagrange interpolation using existing pseudodistributions. In the remainder of [Part I](#), we showcase this technique by generating a design for the Sink-of-DAG formula (SOD), defined next. While this cannot produce a lower bound for Sherali-Adams, as Sherali-Adams has low-degree refutations of SOD, the construction in this section will play an important role in [Part II](#) and [Part III](#).

Definition 4.1. Let $n \geq 2$ be any positive integer. The SOD_n formula is defined as follows. We have an $n \times n$ grid of nodes, where each node $u \in [n]^2$ has an associated *activity variable* $(a_u, \{0, 1\})$, and each node $u \in [n-1] \times [n]$ has an associated *successor variable* $(s_u, [n])$. If $u = (i, j) \in [n]^2$ we let $r(u) = i$ denote the row index of u . The clauses of the formula are the following:

- $\llbracket a_{11} \neq 0 \rrbracket$
- $\llbracket a_u \neq 1 \rrbracket \vee \llbracket s_u \neq i \rrbracket \vee \llbracket a_{(r(u)+1, i)} \neq 0 \rrbracket$ for all $u \in [n-1] \times [n]$ and all $i \in [n]$.
- $\llbracket a_u \neq 1 \rrbracket$ for all $u \in \{n\} \times [n]$.

4.1 The SOD Design

We use the interpolation technique introduced in the previous section to generate a design for SOD_n .

First, we describe a method to generate assignments to SOD_n . For each layer $i < n$, let $\pi_i : [n] \rightarrow [n]$ be a permutation mapping the successors on layer i to layer $i+1$, and define $\pi = (\pi_i)_{i \in [n-1]}$. For any $i \in [n]$, we let P_i^π denote the set of nodes on the path containing the node $(1, i)$ under π . Let $\{1\} \subseteq S \subseteq [n]$ be any set containing 1. Given (π, S) , we can generate an assignment for SOD_n by assigning all successors according to π , which creates n disjoint paths from the top row to the bottom row. Next, for every $i \in S$, we active all nodes in P_i^π , and leave all other nodes inactive. Hence every (π, S) defines an assignment $x(\pi, S)$ to the variables of SOD_n .

The distributions $\mathcal{D}_k$. Let n be a sufficiently large integer, and $1 \leq k \leq n$. The distribution $\mathcal{D}_k$ will be obtained by sampling $\pi = (\pi_i)_i$ to be independent and uniformly random permutations for each layer, and then sampling a uniformly random set S with $\{1\} \subseteq S \subseteq [n]$ of size k . We then output $x(\pi, S)$.

Clearly every assignment in the support of $\mathcal{D}_k$ has exactly k violated constraints (i.e. k solutions), corresponding to active nodes at level n . For any assignment x to the variables of SOD_n , we abuse notation and let $\mathcal{D}_k(x)$ denote the probability that $\mathcal{D}_k$ outputs x .

Definition 4.2. Define

$$\mu(x) := \sum_{k=1}^n \omega_k \mathcal{D}_k(x), \quad (2)$$

where $\omega_k = \omega_k^{(n)}$ are the interpolation weights defined in [Theorem 3.1](#).

We use μ to give our design as follows: for any conjunction C , define

$$\mathbb{E}_\mu[C] := \sum_{k=1}^n \omega_k \mathbb{E}_{x \sim \mathcal{D}_k}[C(x)]. \quad (3)$$

In the remainder of this section, we prove that D is an $(n-1)$ -design for SOD_n .

Theorem 4.3. If μ is the signed measure defined in [Equation \(2\)](#), then $\mathbb{E}_\mu$ is a $(n-1)$ -design for SOD_n , and so any $\text{NS}_{\mathbb{R}}$ refutation of SOD_n requires degree at least $n-1$.

The same proof will imply the following stronger corollary, which will be useful in [Part III](#).

Corollary 4.4. *Let $0 \leq r \leq n - 1$, let $t_0, \dots, t_r$ be distinct integers in $[n]$, and let γ_i be their Lagrange weights for evaluation at zero. If we define the measure*

$$\mu := \sum_{i=0}^r \gamma_i \mathcal{D}_{t_i},$$

then $\mathbf{E}_\mu$ defines a degree- r indicator design for SOD_n .

4.2 Proof of SOD Degree Lower Bounds

Since μ is defined by interpolating true probability distributions, $\mathbf{E}_\mu[1] = 1$ and $\mathbf{E}_\mu$ is linear by definition. We need to show that $\mathbf{E}_\mu[C] = 0$ for any conjunction C with $\text{bdeg}(C) \leq n - 1$ which witnesses a falsified clause of SOD_n .

Setup. Let's now consider any conjunction C which witnesses a falsified clause of SOD_n . First, we observe that by definition, the measure μ define in [Equation \(2\)](#) is supported on assignments x which satisfy all clauses of SOD_n other than the axioms corresponding to activated nodes on the bottom layer. This means that for any conjunction C witnessing a falsified clause of SOD_n other than an active node on the bottom layer, we have $\mathbf{E}_\mu[C] = 0$. To prove $\mathbf{E}_\mu[C] = 0$ for all conjunctions C witnessing a violated activity clause on a node $u \in \{n\} \times [n]$, we will use interpolation. The next lemma formalizes what we need from such conjunctions.

Lemma 4.5. *Let C be a conjunction with $\text{bdeg}(C) \leq n - 1$ witnessing a violated activity clause for a node $u \in \{n\} \times [n]$. There is a polynomial $Q_C(z)$ with $\deg(Q_C) \leq \text{bdeg}(C)$ such that $Q_C(0) = 0$, and*

$$Q_C(k) = \mathbb{E}_{\mathcal{D}_k}[C] \quad \text{for all } k = 1, 2, \dots, n.$$

Once we have this lemma, we can complete the proof of [Theorem 4.3](#) as follows. We have already shown that $\mathbf{E}_\mu[1] = 1$, so let C be any conjunction with $\text{bdeg}(C) \leq n - 1$. We have argued above that $\mathbf{E}_\mu[C] = 0$ for any C that witnesses a violated SOD_n clause anywhere other than the bottom row. By [Lemma 4.5](#), we have a degree at most $n - 1$ univariate polynomial $Q_C(z)$ which we can interpolate:

$$\mathbf{E}_\mu[C] = \sum_{k=1}^n \omega_k \mathbb{E}_{\mathcal{D}_k}[C] = \sum_{k=1}^n \omega_k Q_C(k) = Q_C(0) = 0.$$

Note that [Corollary 4.4](#) also follows by an analogous interpolation argument using the same lemma. So, all we need is to prove [Lemma 4.5](#).

Proof of Lemma 4.5. The proof of this lemma is itself broken into two steps. First, in [Lemma 4.6](#), we will show that for any conjunction C , there is a univariate polynomial $Q_C(z)$ that encodes the values of $\mathbb{E}_{\mathcal{D}_k}[C]$. (However, it will turn out to be more convenient to condition on the permutations π used to generate the instances of $\mathcal{D}_k$.) Then, in [Lemma 4.7](#), we will show that when C witnesses a falsified activity clause for a node on the final row, then the polynomial Q_C evaluates to 0.

Lemma 4.6. *For any conjunction C with $\text{bdeg}(C) \leq n$, and any set of permutations $\pi = (\pi_i)$ for each of the layers, there is a univariate polynomial $Q_{C,\pi}(z)$ with $\deg(Q_{C,\pi}) \leq \text{bdeg}(C)$ such that*

$$Q_{C,\pi}(k) = \mathbb{E}_{\mathcal{D}_k}[C \mid \pi] \quad \text{for all } k = 1, \dots, n.$$

Proof. Once we have conditioned on π , the nodes in the instance are partitioned into n paths $P_1, P_2, \dots, P_n$, where all nodes in P_1 are active. The indicators in the conjunction C are either successor indicators ($\llbracket s_u = i \rrbracket$) or activity indicators ($\llbracket a_u = b \rrbracket$). Conditioned on π there are two possibilities. If a successor indicator in C disagrees with π , or if C contains an activity indicator $\llbracket a_u = 0 \rrbracket$ for a node $u \in P_1$, then C will always evaluate to 0. Otherwise, if C does not evaluate to 0 under π , then all successor indicators will be satisfied and we can remove any activity indicator of the form $\llbracket a_u = 1 \rrbracket$ for $u \in P_1$.

Let $B_C(\pi)$ be an indicator random variable which evaluates to 0 when C vanishes for every S containing 1, independently of k . Let C' be the subconjunction of activity indicators occurring in C , where indicators for nodes on P_1 have all been removed, and write

$$C' = \prod_{u \in \mathcal{P}} \llbracket a_u = 1 \rrbracket \prod_{v \in \mathcal{N}} \llbracket a_v = 0 \rrbracket,$$

for index sets $\mathcal{P}, \mathcal{N} \subseteq ([n] \times [n]) \setminus \mathcal{P}_1$. Since π is sampled independently from S , we can write

$$\mathbb{E}_{\mathcal{D}_k}[C \mid \pi] = \mathbb{E}_{|S|=k}[B_C C' \mid \pi].$$

If $B_C(\pi) = 0$, then define $Q_{C,\pi}(z) = 0$. Otherwise, assume $B_C(\pi) = 1$. For each node u , let $p(u) \in [n]$ denote the index of the path containing u , so that $u \in P_{p(u)}$. Define the sets

$$A := \{i \in [n] \setminus \{1\} \mid \exists u \in \mathcal{P} : i = p(u)\}, \quad B := \{i \in [n] \setminus \{1\} \mid \exists v \in \mathcal{N} : i = p(v)\},$$

so that A gathers the paths that have positive activity values in C' , and B gathers the paths that have negative activity values in C' . These sets are disjoint when $B_C = 1$. For each $i \in [n]$, let Y_i be the indicator random variable for membership in S , so $Y_i = 1$ if and only if $i \in S$. This allows us to write

$$\mathbb{E}_{|S|=k}[B_C C' \mid \pi] = \mathbb{E}_{|S|=k} \left[\prod_{i \in A} Y_i \prod_{j \in B} (1 - Y_j) \right] = \sum_{T \subseteq B} (-1)^{|T|} \mathbb{E}_{|S|=k} \left[\prod_{i \in A \cup T} Y_i \right],$$

where we have dropped the conditioning on π since π and S are sampled independently. For any set $I \subseteq [n] \setminus \{1\}$, we have

$$\mathbb{E}_{|S|=k} \left[\prod_{i \in I} Y_i \right] = \frac{\binom{n-|I|-1}{k-|I|-1}}{\binom{n-1}{k-1}} = \frac{(k-1)_{|I|}}{(n-1)_{|I|}}, \quad (*)$$

where $(n)_k = \prod_{i=0}^{k-1} (n-i)$ is the falling-factorial. Therefore, define the polynomial $q_i(z) = \frac{(z-1)_i}{(n-1)_i}$, and we have

$$\mathbb{E}_{|S|=k}[C' \mid \pi] = \sum_{T \subseteq B} (-1)^{|T|} q_{|A|+|T|}(k),$$

which is a degree at most $|A| + |B| \leq \text{bdeg}(C)$ univariate polynomial. Defining

$$Q_{C,\pi}(z) := B_C(\pi) \sum_{T \subseteq B} (-1)^{|T|} q_{|A|+|T|}(z),$$

we have that $Q_{C,\pi}(k) = \mathbb{E}_{\mathcal{D}_k}[C \mid \pi]$ for $k = 1, \dots, n$ and $\deg(Q_{C,\pi}) \leq \text{bdeg}(C)$. $\square$

We make two remarks on the previous lemma. First, from the previous lemma we can derive a polynomial $Q_C(z)$ encoding the values of $\mathbb{E}_{\mathcal{D}_k}[C]$ for each k by defining $Q_C(z) = \mathbb{E}_{\pi}[Q_{C,\pi}(z)]$, and so we can use interpolation to evaluate $\mathbf{E}$ effectively. However, we should not expect this construction to yield a pseudoexpectation, since Sherali-Adams actually has low-degree refutation of SOD_n . Indeed, it turns out to be easy

to find conjunctions which evaluate to negative values. Let u be any node in the first row other than $(1, 1)$, and consider the conjunction $C = \llbracket a_u = 1 \rrbracket$. Then $B_C(\pi) = 1$ for all π , since $u \neq (1, 1)$ and it is in the first row, and so $Q_{C,\pi}(z) = q_1(z) = (z - 1)/(n - 1)$. This means $Q_C(z) = \mathbb{E}_\pi [Q_{C,\pi}(z)] = (z - 1)/(n - 1)$, and so by interpolation we have

$$\mathbb{E}_\mu [\llbracket a_u = 1 \rrbracket] = \sum_{k=1}^n \omega_k \mathbb{E}_{\mathcal{D}_k} [\llbracket a_u = 1 \rrbracket] = \sum_{k=1}^n \omega_k Q_{\llbracket a_u = 1 \rrbracket}(k) = Q_{\llbracket a_u = 1 \rrbracket}(0) = -\frac{1}{n-1}.$$

Second, while the previous argument proves representation by a low-degree polynomial, it is not clear that $Q_C(0) = 0$ when C is a conjunction that witnesses falsified activity clauses on the bottom row. We will handle this in the next lemma, which is directly inspired from Claim 1 in [GHJ⁺24], and follows the same proof. To state the next lemma, if (π, S) is a sample from $\mathcal{D}_k$, then for any node $u \in \{n\} \times [n]$, we let $X_u(\pi)$ be the indicator random variable which is 1 if u is connected to the node $(1, 1)$ in the sample.

Lemma 4.7. *For any $k \leq n$, any node $u \in \{n\} \times [n]$, and any conjunction C' with $\text{bdeg}(C') \leq n - 2$,*

$$\mathbb{E}_{\mathcal{D}_k} [X_u C'] = \frac{1}{k} \mathbb{E} [\llbracket a_u = 1 \rrbracket C'].$$

Proof. We assume n is sufficiently large. Condition on C' evaluating to 1, and if we can show

$$\mathbb{E}_{\mathcal{D}_k} [X_u \mid C' = 1] = \frac{1}{k} \mathbb{E}_{\mathcal{D}_k} [\llbracket a_u = 1 \rrbracket \mid C' = 1],$$

then we are finished by averaging over assignments where $C' = 1$. Since C' queries at most $n - 2$ successor variables, and there are $n - 1$ layers of successor pointers, there is a row $i \in [n - 1]$ for which C' queries no successor in row i . Consider the following alternate sampling method for $\mathcal{D}_k$: first sample π, S , and then resample the permutation π_i determining row i by randomly mapping active nodes in layer i to active nodes in $i + 1$, and inactive nodes in layer i to inactive nodes in layer $i + 1$. Let x be a sample from $\mathcal{D}_k$, and let x' be any sample obtained in this resampling procedure. Clearly $x' \sim \mathcal{D}_k$, and also $C'(x) = C'(x')$, since C' does not read any nodes in row i . However, after resampling the distinguished sink node connected to $(1, 1)$ is rewired uniformly at random to one of k active bottom nodes. Hence

$$\mathbb{E}_x [X_u \mid C'(x) = 1] = \mathbb{E}_{x'} [X_u \mid C'(x') = 1] = \frac{1}{k} \mathbb{E}_{x'} [\llbracket a_u = 1 \rrbracket \mid C'(x') = 1] = \frac{1}{k} \mathbb{E}_x [\llbracket a_u = 1 \rrbracket \mid C'(x) = 1],$$

which finishes the proof. $\square$

We can now complete the proof of Lemma 4.5. Consider any conjunction of the form $C = \llbracket a_u = 1 \rrbracket C'$, where $\text{bdeg}(C) \leq n - 1$. By Lemma 4.6, for any set of permutations π , there is a polynomial $Q_{C',\pi}(z)$ with $\deg(Q_{C',\pi}) \leq \text{bdeg}(C') \leq n - 2$ such that $Q_{C',\pi}(k) = \mathbb{E}_{\mathcal{D}_k} [C' \mid \pi]$. Combining this with Lemma 4.7 implies

$$\frac{1}{k} \mathbb{E}_{\mathcal{D}_k} [C] = \mathbb{E}_{\mathcal{D}_k} [X_u C'] = \mathbb{E}_\pi [X_u(\pi) Q_{C',\pi}(k)].$$

Rearranging, we have for all $k = 1, \dots, n$,

$$k \mathbb{E}_\pi [X_u(\pi) Q_{C',\pi}(k)] = \mathbb{E}_{\mathcal{D}_k} [C].$$

Therefore, the polynomial $Q_C(z) = z \mathbb{E}_\pi [X_u(\pi) Q_{C',\pi}(z)]$ has $\deg(Q_C) \leq \deg(Q_{C',\pi}) + 1 \leq n - 1$, satisfies $Q_C(z) = \mathbb{E}_{\mathcal{D}_k} [C]$ for all $k = 1, \dots, n$, and also satisfies $Q_C(0) = 0$. $\square$

Part II

Separating Sherali-Adams and Res(2)

5 The CPLS Formula

In this part, we extend the arguments from [Part I](#) and use interpolation to prove a degree lower bound in Sherali-Adams for the *Coloured PLS formulas*, introduced by [KST07]. Using this degree lower bound, we will obtain an exponential size separation between Sherali-Adams and Res(2). We begin by defining the formula.

Definition 5.1. For any positive integer n , the CPLS_n formula is defined as follows.

Consider an $n \times n$ grid of nodes $[n] \times [n]$. Each node $u \in [n]^2$ contains a set of n binary values $c_{u,\alpha} \in \{0, 1\}$ called *colours*. The node $(1, 1)$ is called the *distinguished source node*. The variables of the formula are defined as follows:

- $(a_u, \{0, 1\})$, representing whether the node u is active, for all $u \in [n]^2$.
- $(s_u, [n])$, representing the successor of node u for all $u \in [n-1] \times [n]$.
- $(c_{u,\alpha}, \{0, 1\})$, representing whether the colour α is active at u , for all $u \in [n]^2$ and all $\alpha \in [n]$.
- $(e_u, [n])$, selecting an active colour for u if u is inactive, for all $u \in [n]^2$.

The constraints of CPLS are the following families of clauses:

1. *Colourless Source.* $\llbracket c_{1,1,\alpha} \neq 1 \rrbracket$ for all $\alpha \in [n]$, indicating that the source has no colours.
2. *Inactive Sinks.* $\llbracket a_u \neq 1 \rrbracket$ for all $u \in \{n\} \times [n]$, encoding that each sink is inactive.
3. *Colour Creation.* $\llbracket a_u \neq 0 \rrbracket \vee \llbracket e_u \neq \alpha \rrbracket \vee \llbracket c_{u,\alpha} \neq 0 \rrbracket$ for $u \in [n]^2$ and $\alpha \in [n]$, indicating that if a_u is inactive and $e_u = \alpha$, then $c_{u,\alpha} \neq 0$.
4. *Colour Propagation.* $\llbracket a_{ij} \neq 1 \rrbracket \vee \llbracket s_{i,j} \neq k \rrbracket \vee \llbracket c_{i+1,k,\alpha} \neq 1 \rrbracket \vee \llbracket c_{i,j,\alpha} \neq 0 \rrbracket$, for all $i, j, k, \alpha \in [n]$ with $i < n$. Informally, if (i, j) is an active node with $(i+1, k)$ as its successor, then any active colour at $(i+1, k)$ must also be an active colour at (i, j) .

We can give an informal argument that the CPLS_n formula is unsatisfiable as follows. Start from the distinguished source node $(1, 1)$. If $(1, 1)$ is inactive, then the colour creation axiom implies it must have a colour, contradicting that it is colourless. Hence, $(1, 1)$ is active and must have a successor $(2, i)$. If $(2, i)$ is inactive, then it has a colour, but now colour propagation implies that $(1, 1)$ must have a colour as well, which is a contradiction. Hence $(2, i)$ is active. By induction, we can trace a path of active nodes from $(1, 1)$ to the bottom level. But every node (n, i) on the bottom level is inactive, and hence has a colour, which is a final contradiction.

In the remainder of this part we will construct a pseudoexpectation for CPLS_n using interpolation. Our construction directly builds off the construction for the Sink-of-DAG formula introduced in the previous section. As we have seen, the construction for SOD_n in the previous part is not a pseudoexpectation, in the sense that it gave negative values to certain conjunctions. For CPLS_n we will be able to exploit the colours to prove positivity for all conjunctions.

6 The CPLS Pseudoexpectation

The distributions $\mathcal{D}_k$. Let n be sufficiently large, and for any $1 \leq k \leq n$, define the distribution $\mathcal{D}_k$ as follows. First, we assign all activity bits to be constant: if $u \in [n-1] \times [n]$, then u is active, and if $u \in \{n\} \times [n]$, then u is inactive. We sample (π, S) exactly as we did in the corresponding distribution

in [Section 4.1](#): we sample uniformly random permutations π for each layer, and independently choose a random subset S with $\{1\} \subseteq S \subseteq [n]$ of size k . As before, the permutations π set the successor pointers, and partition the nodes $[n] \times [n]$ into n disjoint paths $P_1, P_2, \dots, P_n$. It remains to sample the colour bits and selector pointers. For each $i \in [n]$, independently and uniformly sample a random colour $\gamma_i \in [n]$, and for the unique node $u \in n \times [n]$ on the path P_i , assign the selector pointer $e_u = \gamma_i$. Then, for each path P_i , if $i \notin S$ then for every node $v \in P_i$, assign $c_{v, \gamma_i} = 1$. Assign all other colours to 0, and assign all other selector pointers independently and uniformly at random.

Summarizing the above construction: each sample from $\mathcal{D}_k$ is determined by a tuple (π, S, γ, e) , where π encodes the matchings at each layer, $\{1\} \subseteq S \subseteq [n]$, $|S| = k$, is the set of colourless paths, $\gamma = (\gamma_i)_{i \in [n]}$ indicates a potential colour in each path, and e determines the values of the *internal* selector variables (selector variables on the bottom layer are assigned according to γ). The instance $x = x(\pi, S, \gamma, e)$ produced is a set of k uniformly random colourless paths (containing the distinguished path) moving from the top layer to the bottom layer, with the remaining $n - k$ paths each having a single uniformly random colour. The only violated constraints of CPLS_n are the colour creation constraints for nodes on the bottom level at the k colourless paths.

Definition 6.1. Let $d = n/100$, and define

$$\mu(x) := \sum_{k=1}^{d+2} \omega_k \mathcal{D}_k(x), \quad (4)$$

where $\omega_k = \omega_k^{(d+2)}$ are the interpolation weights defined in [Theorem 3.1](#).

We prove that $\mathbf{E}_\mu$ is a block-degree- d pseudoexpectation operator for CPLS_n , meaning that $\mathbf{E}_\mu$ is a d -design for CPLS_n and also $\mathbf{E}_\mu[C] \geq 0$ for every conjunction C with $\text{bdeg}(C) \leq d$.

Theorem 6.2. For all sufficiently large n , if μ is the measure defined in [Equation \(4\)](#), then $\mathbf{E}_\mu$ is a block-degree- d pseudoexpectation operator for CPLS_n .

7 Proof of CPLS Degree Lower Bound

In the remainder of this section we prove [Theorem 6.2](#). From now on we will use degree to mean block-degree. As we have seen, $\mathbf{E}_\mu[1] = 1$ by interpolating the constant 1 polynomial, and it is linear by definition. This means we need to show that $\mathbf{E}_\mu[C] = 0$ for every witnessing conjunction C , and also that $\mathbf{E}_\mu[C] \geq 0$ for every conjunction C . For concreteness, we record these in the next two lemmas, and proving both of them will complete the proof of [Theorem 6.2](#).

Lemma 7.1. For any witnessing conjunction C with $\text{bdeg}(C) \leq d$, $\mathbf{E}_\mu[C] = 0$.

Lemma 7.2. For any conjunction C with $\text{bdeg}(C) \leq d$, $\mathbf{E}_\mu[C] \geq 0$.

At a high level, the proof in this section proceeds along the same path as the proof in [Section 4.1](#). For each conjunction C , we will describe a univariate polynomial $Q_C(z)$ which satisfies $Q_C(k) = \mathbb{E}_{\mathcal{D}_k}[C]$, and argue that $Q_C(0) = 0$ whenever C witnesses a violated constraint of CPLS_n . However, in order to prove that $\mathbf{E}_\mu$ is a pseudoexpectation, we must prove that $\mathbf{E}_\mu[C] \geq 0$ for all conjunctions C . To do this we will again use interpolation, but this means that we will need to know the precise syntactic form of the polynomials $Q_C(z)$. This will significantly complicate the statements and proofs of the lemmas.

7.1 Setup and Technical Lemmas

As this proof will be significantly more technical than the corresponding proof in the previous section, we will begin by setting down some notation and stating the main technical lemmas we will need, while deferring some of their proofs.

First, we observe that sampling from $\mathcal{D}_k$ automatically sets many conjunctions to 0. This is recorded in the next definition.

Definition 7.3. A conjunction C is $\mathcal{D}$ -inconsistent if it has any of the following subconjunctions:

1. $\llbracket c_{1,1,\alpha} = 1 \rrbracket$ for any $\alpha \in [n]$,
2. $\llbracket a_{i,j} = 0 \rrbracket$ for some $i < n, j \in [n]$,
3. $\llbracket a_{n,j} = 1 \rrbracket$ for some $j \in [n]$,

Furthermore, suppose $\pi = (\pi_i)_{i \in [n-1]}$ is any set of permutations mapping successors of layers i to $i + 1$, and let $P_1, \dots, P_n$ be the n paths created by π . We say C is π -inconsistent if it has any of the following sub-conjunctions:

1. A successor indicator $\llbracket s_{ij} = k \rrbracket$ with $\pi_i(j) \neq k$.
2. A colour indicator $\llbracket c_{u,\alpha} = 1 \rrbracket$ for any $u \in P_1^\pi$.
3. For any P_i , any nodes $u, v \in P_i$, and any colour $\alpha \in [n]$, the conjunction $\llbracket c_{u,\alpha} = 1 \rrbracket \llbracket c_{v,\alpha} = 0 \rrbracket$.
4. For any P_i , any $u, (n, j) \in P_i$, and any $\alpha \neq \beta \in [n]$, the conjunction $\llbracket c_{u,\alpha} = 1 \rrbracket \llbracket e_{n,j} = \beta \rrbracket$.
5. For any P_i , any $u, v \in P_i$ and any $\alpha \neq \beta \in [n]$, the conjunction $\llbracket c_{u,\alpha} = 1 \rrbracket \llbracket c_{v,\beta} = 1 \rrbracket$.

Proposition 7.4. If C is $\mathcal{D}$ -inconsistent and x is a sample from the support of $\mathcal{D}_k$ for any k , then $C(x) = 0$. Furthermore, if C is π -inconsistent, and x is a sample from the support of $\mathcal{D}_k$ whose underlying layer permutations are π , then $C(x) = 0$.

Proof. By inspection of each case. For π -inconsistency, we note that no sample from $\mathcal{D}_k$ ever has more than one colour on a path, the distinguished path P_1^π never contains any colours, and if P_i^π is asserted to contain a colour α , then the selector for the sink node of P_i^π is hard-wired to choose α . $\square$

Next, we introduce the main technical lemmas we will use in the proofs of [Lemma 7.1](#) and [Lemma 7.2](#). A close relative of the polynomial we introduce first has already appeared in the proof of [Lemma 4.6](#).

Definition 7.5. For any non-negative integer $a \leq n - 1$, define the degree- a univariate polynomial

$$Q_a(z) := \frac{(n - z)_a}{(n - 1)_a},$$

where we have used the falling factorial notation $(n)_k = \prod_{i=0}^{k-1} (n - i)$ (note that $(n)_0 = 1$).

The importance of these polynomials is highlighted in the next lemma. Formally, for any $i \in [n]$, let X_i denote the indicator random variable which is 1 whenever $i \notin S$. We emphasize that X_i is the indicator for *non-membership* in S , as S chooses the colourless paths in our sampling procedure. (For instance, this means $X_1 = 0$ always.) Letting (π, S, γ, e) denote a sample from $\mathcal{D}_k$, the Q_a polynomials encode joint moments of the indicators X_i .

Lemma 7.6. Let $\pi = (\pi_i)_{i \in [n-1]}$ be any set of permutations for each layer of pointers in CPLS_n . For any $k \in [n]$, and any set $A \subseteq [n] \setminus \{1\}$ with $|A| \leq n$, we have

$$Q_{|A|}(k) = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} X_i \mid \pi \right].$$

Proof. If $|A| > n - k$ then both sides of the equation are 0, so assume $|A| \leq n - k$. The permutation π fixes the paths, and hence if $X_i = 1$ for all $i \in A$, then all paths indexed by A must be chosen to be coloured. This means that

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} X_i \mid \pi \right] &= \frac{\binom{n-|A|-1}{k-1}}{\binom{n-1}{k-1}} \\ &= \frac{(n-k)(n-k-1) \cdots (n-k-|A|+1)}{(n-1)(n-2) \cdots (n-|A|)} \\ &= \frac{(n-k)_{|A|}}{(n-1)_{|A|}} \\ &= Q_{|A|}(k). \end{aligned} \quad \square$$

With this lemma in hand, we are now ready to state the central technical lemma of the proof, which is the analogue of [Lemma 4.6](#) in the construction of the SOD_n design.

Lemma 7.7. *Let $\pi = (\pi_i)_{i \in [n-1]}$ be any collection of matchings of successor pointers from layer i to layer $i + 1$. For any conjunction C with $\text{bdeg}(C) \leq d$, there is a set $B \subseteq [n] \setminus \{1\}$ and a non-negative integer a with $a + |B| \leq \text{bdeg}(C)$, as well as nonnegative numbers $\delta_i \geq 0$ for all $i \in B$ such that the following holds. Define the degree $\leq \text{bdeg}(C)$ univariate polynomial*

$$P_{\pi, C}(z) := \sum_{T \subseteq B} (-1)^{|T|} Q_{a+|T|}(z) \prod_{j \in T} \delta_j.$$

Then there is a non-negative function $h_{\pi, C}(n)$ such that for every $k \in [n]$,

$$h_{\pi, C}(n) P_{\pi, C}(k) = \mathbb{E}_{\mathcal{D}_k} [C \mid \pi].$$

Moreover, if C does not contain any selector indicators $\llbracket e_u = \sigma \rrbracket$, where u is a node on row n , then $\sum_{i \in B} \delta_i \leq \text{bdeg}(C)/n$.

The next technical lemma shows that for any node $u \in \{n\} \times [n]$ on the bottom row, and any colour $\sigma \in [n]$, we can “exchange” selector indicators $\llbracket e_u = \sigma \rrbracket$ with colour indicators $\llbracket c_{u, \sigma} = 1 \rrbracket$. Peeking ahead, this lemma is proved using a similar approach as the proof of [Lemma 4.7](#). It will play an important role in proving both positivity and that witnessing conjunctions vanish.

Lemma 7.8. *For any node $u \in \{n\} \times [n]$, any $\sigma \in [n]$, and any conjunction C with $\text{bdeg}(C) \leq d - 1$,*

$$\mathbf{E}_{\mu} [\llbracket e_u = \sigma \rrbracket C] = \mathbf{E}_{\mu} [\llbracket c_{u, \sigma} = 1 \rrbracket C].$$

7.2 The CPLS Pseudoexpectation is Non-Witnessing

With the technical lemmas in hand, we are ready to prove [Lemma 7.1](#) and [Lemma 7.2](#). We begin by proving [Lemma 7.1](#): namely, that every low-degree witnessing conjunction C vanishes under $\mathbf{E}_{\mu}$.

Proof of Lemma 7.1. Since C is witnessing, it must contain a subconjunction that witnesses the falsity of one of the clauses listed at the beginning of [Definition 5.1](#). By [Proposition 7.4](#), for every $k \leq d + 2$, every input x in the support of $\mathcal{D}_k$ falsifies any witnessing conjunction for CPLS_n , other than a conjunction C containing

$$A_{n, j, \alpha} := \llbracket a_{(n, j)} = 0 \rrbracket \llbracket e_{(n, j)} = \alpha \rrbracket \llbracket c_{n, j, \alpha} = 0 \rrbracket$$

as a subconjunction for some $j, \alpha \in [n]$. So, without loss of generality, consider any conjunction C of the form $A_{n,j,\alpha}R$ for some conjunction R of degree $\text{bdeg}(R) \leq \text{bdeg}(C) - 3$. By [Lemma 7.8](#), we can exchange the selector indicator for a colour indicator:

$$\begin{aligned}\mathbf{E}_\mu[A_{n,j,\alpha}R] &= \mathbf{E}_\mu[\llbracket a_{(n,j)} = 0 \rrbracket \llbracket e_{(n,j)} = \alpha \rrbracket \llbracket c_{n,j,\alpha} = 0 \rrbracket R] \\ &= \mathbf{E}_\mu[\llbracket a_{(n,j)} = 0 \rrbracket \llbracket c_{n,j,\alpha} = 1 \rrbracket \llbracket c_{n,j,\alpha} = 0 \rrbracket R] = 0.\end{aligned}$$

This final expression is always 0, since a colour cannot be both active and inactive. $\square$

7.3 The CPLS Pseudoexpectation is Non-Negative

Next, we prove non-negativity, which will use both of the technical lemmas [Lemma 7.7](#) and [Lemma 7.8](#) from [Section 7.1](#).

Proof of [Lemma 7.2](#). Let C be any conjunction with $\text{bdeg}(C) \leq d$. By repeatedly applying [Lemma 7.8](#), to prove the lemma we may assume without loss of generality that C contains no bottom-row selector variables. By definition of $\mathbf{E}$, we have

$$\mathbf{E}_\mu[C] = \sum_{k=1}^{d+2} \omega_k \mathbb{E}_{\mathcal{D}_k}[C].$$

Towards applying the previous lemmas, for any set of permutations π , define

$$\mathbf{E}_\mu[C|\pi] := \sum_{k=1}^{d+2} \omega_k \mathbb{E}_{\mathcal{D}_k}[C|\pi].$$

We will prove $\mathbf{E}_\mu[C|\pi] \geq 0$ for all choices of π , and the result will hold for $\mathbf{E}_\mu[C]$ by averaging over the choice of π .

By [Lemma 7.7](#), there is a non-negative function $h(n) := h_{\pi,C}(n)$ and a degree $\leq \text{bdeg}(C)$ polynomial $P(z) := P_{\pi,C}(z)$ such that

$$\mathbb{E}_{\mathcal{D}_k}[C|\pi] = h(n)P(k),$$

where

$$P(z) = \sum_{T \subseteq B} (-1)^{|T|} Q_{a+|T|}(z) \prod_{j \in T} \delta_j$$

is defined as in the statement of that lemma. Since C has no bottom-row selector variables, we have that $\sum_{i \in B} \delta_i \leq \text{bdeg}(C)/n$, which will be crucial for proving positivity. Applying interpolation, we have $\mathbf{E}_\mu[C] = h(n)P(0)$, so it suffices to show $P(0) \geq 0$. So, we turn our attention to proving this.

Recalling the definition of Q_i from [Definition 7.5](#), we have

$$\begin{aligned}P(0) &= \sum_{T \subseteq B} (-1)^{|T|} Q_{a+|T|}(0) \prod_{j \in T} \delta_j \\ &= \sum_{T \subseteq B} (-1)^{|T|} \frac{n}{n-a-|T|} \prod_{j \in T} \delta_j \\ &= \frac{n}{n-a} + \sum_{\emptyset \neq T \subseteq B} (-1)^{|T|} \frac{n}{n-a-|T|} \prod_{j \in T} \delta_j \\ &\geq \frac{n}{n-a} - \sum_{\emptyset \neq T \subseteq B} \frac{n}{n-a-|T|} \prod_{j \in T} \delta_j.\end{aligned}$$

Since $a \geq 0$, the leading term is at least 1, and since $a + |T| \leq d$, the inner $n/(n - a - |T|)$ term is at most $n/(n - d)$. Hence we have

$$\frac{n}{n - a} - \sum_{\emptyset \neq T \subseteq B} \frac{n}{n - a - |T|} \prod_{j \in T} \delta_j \geq 1 - \frac{n}{n - d} \sum_{\emptyset \neq T \subseteq B} \prod_{j \in T} \delta_j.$$

We can re-write the sum over non-empty subsets:

$$1 - \frac{n}{n - d} \sum_{\emptyset \neq T \subseteq B} \prod_{j \in T} \delta_j = 1 - \frac{n}{n - d} \left(-1 + \prod_{j \in B} (1 + \delta_j) \right).$$

Since C has no bottom-row selector variables, by [Lemma 7.7](#) we have $\Delta := \sum_{j \in B} \delta_j \leq d/n$. Using the fact that $\delta_j \geq 0$ for each j , we substitute Δ and get

$$\prod_{j \in B} (1 + \delta_j) \leq \sum_{i=0}^{|B|} \left(\sum_{j \in B} \delta_j \right)^i \leq \sum_{i=0}^{\infty} \Delta^i \leq \frac{1}{1 - \Delta} \leq \frac{n}{n - d}.$$

Returning to the main calculation, we obtain

$$\begin{aligned} 1 - \frac{n}{n - d} \left(-1 + \prod_{j \in B} (1 + \delta_j) \right) &\geq 1 - \frac{n}{n - d} \left(\frac{n}{n - d} - 1 \right) \\ &= 1 - \frac{nd}{(n - d)^2}. \end{aligned}$$

Let $\varepsilon = 1/100$. Since $d \leq \varepsilon n$, we have

$$1 - \frac{nd}{(n - d)^2} \geq 1 - \frac{\varepsilon n^2}{n^2(1 - \varepsilon)^2} \geq 1 - \frac{\varepsilon}{(1 - \varepsilon)^2}.$$

This last expression is easily verified to be larger than 0, and hence $P(0) \geq 0$. This implies $\mathbf{E}_\mu[C|\pi] \geq 0$, proving the lemma. $\square$

7.4 Proofs of Technical Lemmas

Now we finally prove the main technical lemmas. We begin by proving [Lemma 7.7](#), which will be needed to prove the second. We repeat the statement:

Lemma 7.7. *Let $\pi = (\pi_i)_{i \in [n-1]}$ be any collection of matchings of successor pointers from layer i to layer $i + 1$. For any conjunction C with $\text{bdeg}(C) \leq d$, there is a set $B \subseteq [n] \setminus \{1\}$ and a non-negative integer a with $a + |B| \leq \text{bdeg}(C)$, as well as nonnegative numbers $\delta_i \geq 0$ for all $i \in B$ such that the following holds. Define the degree $\leq \text{bdeg}(C)$ univariate polynomial*

$$P_{\pi, C}(z) := \sum_{T \subseteq B} (-1)^{|T|} Q_{a+|T|}(z) \prod_{j \in T} \delta_j.$$

Then there is a non-negative function $h_{\pi, C}(n)$ such that for every $k \in [n]$,

$$h_{\pi, C}(n) P_{\pi, C}(k) = \mathbb{E}_{D_k}[C | \pi].$$

Moreover, if C does not contain any selector indicators $\llbracket e_u = \sigma \rrbracket$, where u is a node on row n , then $\sum_{i \in B} \delta_i \leq \text{bdeg}(C)/n$.

Proof. As described above, a sample from $\mathcal{D}_k$ is described by a tuple (π, S, γ, e) . As per the statement of the lemma, since we consider a conditional distribution we will assume π is fixed. After conditioning on π we let $P_1, P_2, \dots, P_n$ denote the partition of $[n]^2$ induced by π into disjoint paths. For convenience, let us also condition on an arbitrary assignment $e = (e_{(i,j)})_{i,j}$ to the colour selector variables for each (i, j) with $i < n$ which satisfies the internal selector terms in C . Note that each of these selector variables is selected independently from all other variables, and hence by averaging over e we get that $\mathbb{E}_{\mathcal{D}_k}[C|\pi] = f(n) \mathbb{E}_{\mathcal{D}_k}[C|\pi, e]$ for some non-negative function $f(n) \geq 0$. We can absorb this function into $h_{\pi, C}(n)$, and assume without loss of generality that C has no more internal selector terms, so we proceed to analyze $\mathbb{E}_{\mathcal{D}_k}[C|\pi, e]$.

First, if C is $\mathcal{D}$ -inconsistent or π -inconsistent, then it evaluates to 0 under the conditional distribution, as per [Proposition 7.4](#) above. So, we proceed assuming C is consistent with $\mathcal{D}$ and π . Similarly, if C is inconsistent with the fixed assignment to the selector variables, then it will also evaluate to 0. So, for simplicity, under these assumptions and conditioning, we will omit the successor indicators, activity indicators, and selector indicators on rows above row n from the description of C , and write

$$C = \prod_{(u, \alpha) \in I} \llbracket c_{u, \alpha} = 1 \rrbracket \prod_{(v, \beta) \in J} \llbracket c_{v, \beta} = 0 \rrbracket \prod_{((n, j), \sigma) \in K} \llbracket e_{n, j} = \sigma \rrbracket, \quad (5)$$

where I, J, K are index sets, the total number of indicators in what remains in C is at most $\text{bdeg}(C)$, and where J is assumed without loss of generality to not contain any index (v, β) with $v \in P_1$. Hence, we may view C as asserting that certain colours are present, certain other colours are not present, and that certain selector values on the final row point to certain colours.

We now state the main technical claim of the lemma. Define the sets of path indices

$$\begin{aligned} A &:= \{i \in [n] \mid \exists (u, \alpha) \in I : u \in P_i\}, \\ B &:= \{i \in [n] \setminus A \mid \exists (v, \beta) \in J : v \in P_i\}. \end{aligned}$$

In other words, A contains (indices of) the paths P_i which contain a colour under C , and B contains (indices of) paths P_i which have some colours which are not present under C . Recall X_i is the indicator random variable which is 1 if and only if the path P_i is chosen to contain a colour, and hence $X_i = 1$ if and only if $i \notin S$.

Claim 7.9. There is a non-negative integer t and a non-negative number $\delta_j \geq 0$ for each $j \in B$ such that

$$\mathbb{E}_{\mathcal{D}_k}[C|\pi, e] = \frac{1}{n^t} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \prod_{j \in B} (1 - \delta_j X_j) \mid \pi \right]$$

Moreover, if there are no selector indicators in C , then $\sum_{j \in B} \delta_j \leq \text{bdeg}(C)/n$.

We postpone the proof of the claim, and finish the proof of the lemma. Linearity and [Lemma 7.6](#) imply

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \prod_{j \in B} (1 - \delta_j X_j) \mid \pi \right] &= \sum_{T \subseteq B} (-1)^{|T|} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \prod_{j \in T} \delta_j X_j \mid \pi \right] \\ &= \frac{1}{n^{|A|}} \sum_{T \subseteq B} (-1)^{|T|} \prod_{j \in T} \delta_j \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A \cup T} X_i \mid \pi \right] \\ &= \frac{1}{n^{|A|}} \sum_{T \subseteq B} (-1)^{|T|} Q_{|A|+|T|}(k) \prod_{j \in T} \delta_j. \end{aligned}$$

This is a degree at most $\text{bdeg}(C)$ polynomial expression in k since $\deg(Q_{|A|+|B|}) \leq |A| + |B| \leq \text{bdeg}(C)$. Substituting into [Claim 7.9](#), we get

$$\mathbb{E}_{\mathcal{D}_k}[C|\pi, e] = \frac{1}{n^{t+|A|}} \sum_{T \subseteq B} (-1)^{|T|} Q_{|A|+|T|}(k) \prod_{j \in T} \delta_j.$$

Averaging over e and defining $h_{\pi, C}(n)$ appropriately (in particular, by absorbing the $f(n)$ and $1/n^{t+|A|}$ terms), we obtain

$$\mathbb{E}_{\mathcal{D}_k}[C|\pi] = h_{\pi, C}(n) \sum_{T \subseteq B} (-1)^{|T|} Q_{|A|+|T|}(k) \prod_{j \in T} \delta_j = h_{\pi, C}(n) P_{C, \pi}(k),$$

which, setting $a = |A|$, finishes the proof of the lemma. So all that remains is to prove [Claim 7.9](#).

Proof of Claim. We begin by refining the expression of C given in [Equation \(5\)](#). For simplicity of notation, in the remainder of this proof, we will use the notation $c_{u, \alpha}$ to represent the indicator $\llbracket c_{u, \alpha} = 1 \rrbracket$ and $1 - c_{u, \alpha} = \llbracket c_{u, \alpha} = 0 \rrbracket$. Let $u_i \in P_i$ denote the final node on the path P_i . For any $i \in [n]$, define

$$I_i := \{(u, \alpha) \in I \mid u \in P_i\}, \quad J_i := \{(v, \beta) \in J \mid v \in P_i\}$$

and

$$e_i := \begin{cases} \llbracket e_{u_i} = \sigma_i \rrbracket & \exists (u_i, \sigma_i) \in K \\ 1 & \text{otherwise.} \end{cases}$$

We note that e_i is well-defined since C is consistent, and hence has at most one selector term for each node on the bottom row. In addition to the index sets A and B , define the index set

$$D := \{i \in [n] \setminus (A \cup B) \mid e_i \neq 1\}.$$

Then we can define the conjunctions

$$\begin{aligned} C_A &:= \prod_{i \in A} e_i \prod_{(u, \alpha) \in I_i} c_{u, \alpha} \prod_{(v, \beta) \in J_i} (1 - c_{v, \beta}) \\ C_B &:= \prod_{j \in B} e_j \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}), \\ C_D &= \prod_{i \in D} e_i, \end{aligned}$$

and observe that $C = C_A C_B C_D$ by definition.

First, for each $i \in A$, let $v_i \in P_i$ be some arbitrarily selected node on the path such that $(v_i, \alpha_i) \in I_i$. We show that $\mathbb{E}_{\mathcal{D}_k}[C_A C_B C_D | \pi, e] = \mathbb{E}_{\mathcal{D}_k}[C'_A C_B C_D | \pi, e]$, where

$$C'_A := \prod_{i \in A} c_{v_i, \alpha_i}.$$

Since $i \in A$, C asserts that P_i contains a colour. By consistency, C_A cannot assert that P_i does not have the colour α_i , and so all positive colour indicators must assert α_i , while all negative colour indicators $(1 - c_{v, \beta})$ for $v \in P_i$ are for colours $\beta \neq \alpha_i$. However, all such negative colour indicators will always be set to 1 if $c_{u_i, \alpha_i} = 1$, and we can remove redundant ones as well. Furthermore, if $c_{u_i, \alpha_i} = 1$ and $e_i \neq 1$, then $e_i = \llbracket e_{u_i} = \alpha_i \rrbracket$ by consistency, and will always be satisfied. Hence, under the conditional distribution, we

may select a single positive colour term for each path P_i with $i \in A$, and remove all other positive colour indicators and all selector indicators from C_A .

This means we can consider

$$\mathbb{E}_{\mathcal{D}_k} [C'_A C_B C_D | \pi, e] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} c_{v_i, \alpha_i} \prod_{j \in B} e_j \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \prod_{i \in D} e_i \mid \pi, e \right]$$

We further condition on the colourless path set $\{1\} \subseteq S \subseteq [n]$ with $|S| = k$, noting that

$$\mathbb{E}_{\mathcal{D}_k} [C'_A C_B C_D | \pi, e] = \mathbb{E}_{|S|=k} \left[\mathbb{E}_{\mathcal{D}_k} [C'_A C_B C_D \mid S, \pi, e] \right].$$

This means we just need to prove for some non-negative integer t and for some non-negative δ_j ,

$$\mathbb{E}_{\mathcal{D}_k} [C'_A C_B C_D \mid S, \pi, e] = \frac{1}{n^t} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \prod_{j \in B} (1 - \delta_j X_j) \mid S, \pi \right]. \quad (6)$$

Consider the value of $C'_A C_B C_D$ under this extra conditioning. Under S , we have conditional independence between C'_A , C_B , and C_D since A, B, D are mutually disjoint, and the paths that e_i , and $c_{(v, \beta)}$ all lie on are each fixed, leaving only the independent randomness used for choosing the colours and selector pointers on these paths. Hence we will write

$$\mathbb{E}_{\mathcal{D}_k} [C'_A C_B C_D | S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} [C'_A \mid S, \pi, e] \mathbb{E}_{\mathcal{D}_k} [C_B \mid S, \pi, e] \mathbb{E}_{\mathcal{D}_k} [C_D \mid S, \pi, e].$$

This allows us to consider these terms separately.

First, let's consider the C'_A term, which by conditional independence is equal to

$$\mathbb{E}_{\mathcal{D}_k} [C'_A | S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} c_{v_i, \alpha_i} \mid S, \pi, e \right] = \prod_{i \in A} \mathbb{E}_{\mathcal{D}_k} [c_{v_i, \alpha_i} \mid S, \pi, e].$$

For any sample in the conditional distribution, $c_{v_i, \alpha_i} = 1$ if and only if $i \notin S$ (and hence $X_i = 1$), and also α_i was chosen to be the active colour on the path P_i . Hence we have $\mathbb{E}_{\mathcal{D}_k} [c_{v_i, \alpha_i} | S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} [X_i / n | S, \pi, e]$, which we can reassemble to

$$\mathbb{E}_{\mathcal{D}_k} [C'_A \mid S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \mid S, \pi, e \right]. \quad (7)$$

Let's now consider

$$\mathbb{E}_{\mathcal{D}_k} [C_B \mid S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j \in B} e_j \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right].$$

Consider the partition $B = B_0 \cup B_1$, where $j \in B_0$ if $e_j = 1$, and $j \in B_1$ otherwise. We can write $C_B = C_{B_0} C_{B_1}$, where

$$C_{B_0} := \prod_{j \in B_0} \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta})$$

$$C_{B_1} := \prod_{j \in B_1} \llbracket e_{u_j} = \sigma_j \rrbracket \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}).$$

Using this partition and conditional independence under S , we can consider the terms $\mathbb{E}_{\mathcal{D}_k} [C_{B_i} \mid S, \pi, e]$ separately.

First, consider $\mathbb{E}_{\mathcal{D}_k} [C_{B_0} \mid S, \pi, e]$. By conditional independence across paths we have

$$\mathbb{E}_{\mathcal{D}_k} [C_{B_0} \mid S, \pi, e] = \prod_{j \in B_0} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right]$$

For each $i \in [n]$, let F_i denote the set of colours that are *forbidden* by C on path P_i . Formally,

$$F_i := \{\beta \in [n] \mid \exists (v, \beta) \in J : v \in P_i\}.$$

Define $r_i = |F_i|$ to be the number of colours forbidden by C on P_i .

For each $j \in B_0$, either $j \in S$ or $j \notin S$. If $j \in S$, meaning $X_j = 0$, then the path is colourless, and $c_{v, \beta} = 0$ for all $(v, \beta) \in J_j$, so the product $\prod_{(v, \beta) \in J_j} (1 - c_{v, \beta})$ evaluates to 1. If $j \notin S$, then $X_j = 1$ and the path has one colour, meaning the product evaluates to 1 only if none of the colours in F_j were selected, which happens with probability $1 - r_j/n$. Hence for each $j \in B_0$, we have

$$\mathbb{E}_{\mathcal{D}_k} \left[\prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right] = \mathbb{E}_{\mathcal{D}_k} \left[1 - \frac{r_j}{n} X_j \mid S, \pi, e \right],$$

which we can recombine to obtain

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_k} [C_{B_0} \mid S, \pi, e] &= \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j \in B_0} \left(1 - \frac{r_j}{n} X_j \right) \mid S, \pi, e \right] \\ &= \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j \in B_0} (1 - \delta_j X_j) \mid S, \pi, e \right], \end{aligned}$$

where we define $\delta_j = r_j/n$ in this case.

So, next we consider

$$\mathbb{E}_{\mathcal{D}_k} [C_{B_1} \mid S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j \in B_1} \mathbb{I}[e_{u_j} = \sigma_j] \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right].$$

By conditional independence, it is again enough to consider any individual term in the product over B_1 .

Fix any $j \in B_1$, and consider the term $\mathbb{I}[e_{u_j} = \sigma_j]$. For these j , define $\delta_j = 1$ if and only if $\sigma_j \in F_j$, i.e. if the colour chosen by the selector is forbidden. (Note that δ_j does not depend on the conditioning by S , and only on the definition of the conjunction C .) We claim for any $j \in B_1$,

$$\mathbb{E}_{\mathcal{D}_k} \left[\mathbb{I}[e_{u_j} = \sigma_j] \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right] = \mathbb{E}_{\mathcal{D}_k} \left[\frac{1}{n} (1 - \delta_j X_j) \mid S, \pi, e \right].$$

If $j \in S$, meaning $X_j = 0$ and P_j is colourless, then $e_{u_j} = \sigma_j$ with probability $1/n$ and $\prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) = 1$. In this case, we have

$$\mathbb{E}_{\mathcal{D}_k} \left[\mathbb{I}[e_{u_j} = \sigma_j] \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right] = \frac{1}{n} = \mathbb{E}_{\mathcal{D}_k} \left[\frac{1}{n} (1 - \delta_j X_j) \mid S, \pi, e \right],$$

since $X_j = 0$.

Otherwise, if $j \notin S$, meaning $X_j = 1$, then either $\sigma_j \in F_j$ ($\delta_j = 1$), or $\sigma_j \notin F_j$ ($\delta_j = 0$). If $j \notin S$ and $\sigma_j \in F_j$, then the selector e_{u_j} is hardwired to point to the colour present in the path P_j . Hence if $e_{u_j} = \sigma_j$, then $c_{u, \sigma_j} = 1$ for every $u \in P_j$. So if σ_j is forbidden, then the entire conjunction will evaluate to 0. This means that

$$\mathbb{E}_{\mathcal{D}_k} \left[\mathbb{I}[e_{u_j} = \sigma_j] \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right] = 0 = \mathbb{E}_{\mathcal{D}_k} \left[\frac{1}{n} (1 - \delta_j X_j) \mid S, \pi, e \right],$$

where the second equality follows since $\delta_j = X_j = 1$. On the other hand, if $j \notin S$ and $\sigma_j \notin F_j$, then $e_{u_j} = \sigma_j$ with probability $1/n$ and the product of the remaining colour indicators evaluates to 1, since they are never chosen to be 0. Hence

$$\mathbb{E}_{\mathcal{D}_k} \left[\mathbb{I}[e_{u_j} = \sigma_j] \prod_{(v, \beta) \in J_j} (1 - c_{v, \beta}) \mid S, \pi, e \right] = \frac{1}{n} = \mathbb{E}_{\mathcal{D}_k} \left[\frac{1}{n} (1 - \delta_j X_j) \mid S, \pi, e \right],$$

which follows since $\delta_j = 0$.

Finally, consider the expectation of C_D . Since all of these selectors are chosen independently and uniformly by γ , we have

$$\mathbb{E}_{\mathcal{D}_k} [C_D \mid S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j \in D} \mathbb{I}[e_{u_j} = \sigma_j] \mid S, \pi, e \right] = \frac{1}{n^{|D|}}.$$

Combining the factors from C_{B_0}, C_{B_1} , and C_D using conditional independence under S , we have

$$\mathbb{E}_{\mathcal{D}_k} [C_B C_D \mid S, \pi, e] = \mathbb{E}_{\mathcal{D}_k} [C_{B_0} C_{B_1} C_D \mid S, \pi, e] = \frac{1}{n^t} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in B} (1 - \delta_i X_i) \mid S, \pi, e \right] \quad (8)$$

for some non-negative integer t . Using conditional independence, [Equations \(7\) and \(8\)](#), and the independence of X_i from e , we have

$$\begin{aligned} \mathbb{E}_{\mathcal{D}_k} [C \mid S, \pi, e] &= \mathbb{E}_{\mathcal{D}_k} [C'_A C_B C_D \mid S, \pi, e] \\ &= \mathbb{E}_{\mathcal{D}_k} [C'_A \mid S, \pi, e] \mathbb{E}_{\mathcal{D}_k} [C_B C_D \mid S, \pi, e] \\ &= \frac{1}{n^t} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \mid S, \pi, e \right] \mathbb{E}_{\mathcal{D}_k} \left[\prod_{j \in B} (1 - \delta_j X_j) \mid S, \pi, e \right] \\ &= \frac{1}{n^t} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \prod_{j \in B} (1 - \delta_j X_j) \mid S, \pi, e \right] \\ &= \frac{1}{n^t} \mathbb{E}_{\mathcal{D}_k} \left[\prod_{i \in A} \frac{X_i}{n} \prod_{j \in B} (1 - \delta_j X_j) \mid S, \pi \right], \end{aligned}$$

which proves [Equation \(6\)](#). To prove the final note in the claim, observe that if $e_j = 1$ for all $j \in B$, then $B_1 = \emptyset$, and hence $\delta_j = r_j/n$ for all $j \in B$. But r_j is the number of forbidden colours on path j , and the total number of forbidden colours is no more than the degree of C . Hence, $\sum_{j \in B} \delta_j \leq \text{bdeg}(C)/n$. $\square$

□

Next, we prove [Lemma 7.8](#), which will use a re-randomization argument similar to the proof of [Lemma 4.7](#). We restate the lemma for convenience:

Lemma 7.8. *For any node $u \in \{n\} \times [n]$, any $\sigma \in [n]$, and any conjunction C with $\text{bdeg}(C) \leq d - 1$,*

$$\mathbf{E}_{\mu} [\llbracket e_u = \sigma \rrbracket C] = \mathbf{E}_{\mu} [\llbracket c_{u,\sigma} = 1 \rrbracket C].$$

Proof. We prove this lemma in three steps. First, for any node $u \in \{n\} \times [n]$ on the bottom row, let Z_u be an indicator random variable for the event that $u \in P_1$, and let Y_u be an indicator random variable for the event that u has no colours. Since Z_u is defined completely by π , by [Lemma 7.7](#) we can define

$$P_{u,C}(z) := \mathbb{E}_{\pi} [Z_u(\pi) h_{\pi,C}(n) P_{\pi,C}(z)].$$

This polynomial has degree at most $\text{bdeg}(C)$, and for all k , $P_{u,C}(k) = \mathbb{E}_{\mathcal{D}_k} [Z_u C]$.

Second, we use the same resampling trick that was used in [Lemma 4.7](#) to prove that

$$\mathbb{E}_{\mathcal{D}_k} [Z_u C] = \frac{1}{k} \mathbb{E}_{\mathcal{D}_k} [Y_u C], \quad (9)$$

and so for all k ,

$$k P_{u,C}(k) = \mathbb{E}_{\mathcal{D}_k} [Y_u C].$$

So, if we define the polynomial $P'(z) = z P_{u,C}(z)$, then $P'(0) = 0$ and $P'(k) = \mathbb{E}_{\mathcal{D}_k} [Y_u C]$. By an interpolation argument, we will finally use the polynomial P' to prove the lemma.

We now turn to proving the lemma, and begin by proving [Equation \(9\)](#). If C is identically 0 then the result holds, so suppose this is not the case. Since $d \leq n/100$, there is a row $i \in [n - 1]$ such that C does not query any variables from nodes in row i . Consider the conditional expectations

$$\mathbb{E}_{x \sim \mathcal{D}_k} [Z_u \mid C(x) = 1], \quad \mathbb{E}_{x \sim \mathcal{D}_k} [Y_u \mid C(x) = 1].$$

Let x' be obtained by first sampling $x \sim \mathcal{D}_k$, and then replacing the matching between colourless paths on level i with a uniformly random matching between these paths. Clearly $x' \sim \mathcal{D}_k$, and $C(x) = C(x')$ since C does not query any nodes in row i . However, after resampling, the distinguished sink node is rewired uniformly at random to one of the k colourless nodes. Hence

$$\mathbb{E}_{x \sim \mathcal{D}_k} [Z_u \mid C(x) = 1] = \mathbb{E}_{x' \sim \mathcal{D}_k} [Z_u \mid C(x') = 1] = \frac{1}{k} \mathbb{E}_{x' \sim \mathcal{D}_k} [Y_u \mid C(x') = 1] = \frac{1}{k} \mathbb{E}_{x \sim \mathcal{D}_k} [Y_u \mid C(x) = 1].$$

Averaging over assignments satisfying C completes the proof of [Equation \(9\)](#).

With this in hand, consider the conjunction $C' = \llbracket e_u = \sigma \rrbracket C$, and consider the polynomial $P'(z) = z P_{u,C'}(z)$. Clearly $\deg(P') \leq \text{bdeg}(C') + 1 \leq d + 1$, and so by interpolation we have

$$\mathbf{E}_{\mu} [Y_u C'] = P'(0) = 0.$$

To use this to finish the proof of the lemma, observe that for every k ,

$$\mathbb{E}_{\mathcal{D}_k} [Y_u \llbracket e_u = \sigma \rrbracket C] = \mathbb{E}_{\mathcal{D}_k} [(\llbracket e_u = \sigma \rrbracket - \llbracket c_{u,\sigma} = 1 \rrbracket) C].$$

To see this, consider any element x in the support of $\mathcal{D}_k$. Observe that if $Y_u = 1$ then u has no colours, and so $c_{u,\sigma} = 0$. Hence $Y_u \llbracket e_u = \sigma \rrbracket C$ has the same value as $\llbracket e_u = \sigma \rrbracket C$ in this case. On the other hand, if

$Y_u = 0$, then u has a colour, and so either $e_u = \sigma$, implying $c_{u,\sigma} = 1$, or $e_u \neq \sigma$, implying $c_{u,\sigma} = 0$. In either case, the RHS evaluates to 0.

Performing this substitution everywhere and using linearity, we have

$$\mathbf{E}_\mu [Y_u \llbracket e_u = \sigma \rrbracket C] = \mathbf{E}_\mu [\llbracket e_u = \sigma \rrbracket C] - \mathbf{E}_\mu [\llbracket c_{u,\sigma} = 1 \rrbracket C].$$

But we have just shown the LHS of this equation is 0, and so $\mathbf{E}_\mu [\llbracket e_u = \sigma \rrbracket C] = \mathbf{E}_\mu [\llbracket c_{u,\sigma} = 1 \rrbracket C]$. $\square$

8 Application: Separating Sherali-Adams and Res(2)

The following result follows immediately from the previous section, when combined with the translation theorem from [Theorem 2.16](#).

Theorem 8.1. *If ℓ is a positive integer, and $n = 2^\ell$, then $\text{SA}(\text{CPLS}_n^{\text{bool}}) = \Omega(n)$.*

Using this theorem, we can prove an exponential separation between Sherali-Adams and the Res(2) proof system, which is the extension of Resolution whose lines are 2-DNF formulas. The particular separating formula will be an XOR-composed version of CPLS_n .

Theorem 8.2. *Let ℓ be a sufficiently large positive integer and let $n = 2^\ell$. The formula $\text{CPLS}_n^{\text{bool}} \circ \oplus$ has polynomial size Res(2) refutations, but any Sherali-Adams refutation in the dual-variable encoding requires size $2^{\Omega(n)}$.*

In the remainder of this section we prove this theorem. We first recall the definition of the Res(k) proof system, and define the formula $F \circ \oplus$. Then we will prove the polynomial size upper bound in Res(2) and the exponential lower bound for Sherali-Adams.

First, we recall the definition of Res(k).

Definition 8.3 (*Res(k) and Resolution*). Let k be a positive integer, let $F = C_1 \wedge \dots \wedge C_m$ be a CNF formula, and let G be a k -DNF formula. A *Res(k) proof* of G from F is a sequence of k -DNFs

$$\Pi = G_1, G_2, \dots, G_s$$

such that $G_s = G$ and, for all $i \in [s]$, each G_i is either a clause of F or it is derived from earlier lines from one of the proof rules depicted in [Figure 6](#). A Res(k) proof is a *Resolution* proof if every line is a clause, and the only proof rule used is the cut rule. The *size* of the proof is the total number of symbols appearing in it. The proof is a *refutation* of F if $G = \perp$. In the rules below, A and B are DNF formulas and ℓ, ℓ_i are boolean literals.

$$\begin{array}{ccc} \frac{A \vee \bigwedge_i \ell_i \quad A \vee \ell}{A \vee (\ell \wedge \bigwedge_i \ell_i)} & \frac{A \vee (\bigwedge_i \ell_i) \quad B \vee (\bigvee_i \neg \ell_i)}{A \vee B} & \\ \text{\textbf{\(\wedge\)-Introduction}} & \text{\textbf{Cut}} & \\ \\ \frac{A}{A \vee B} & \frac{}{\ell \vee \neg \ell} & \\ \text{\textbf{Weakening}} & \text{\textbf{Axiom Introduction}} & \end{array}$$

Figure 6: Inference rules of Res(k). Each conjunction appearing in the proof must have at most k literals occurring.

Let $F(x) = C_1 \wedge C_2 \wedge \dots \wedge C_m$ be a CNF formula on n variables $x_1, \dots, x_n$ with width at most w . The formula $(F \circ \oplus)(y, z)$ is a CNF formula with $2n$ variables $y_1, y_2, \dots, y_n, z_1, \dots, z_n$, width $2w$, and with at most $m2^w$ clauses, which is defined as follows. Informally, for each variable x_i , we replace x_i with $y_i \oplus z_i$, where $\oplus$ denotes the two-bit XOR function, and then re-write the result in CNF. Syntactically, this is done as follows. For each variable x_i , and each occurrence of x_i in any clause, we replace positive occurrences with the formula $(y_i \vee z_i) \wedge (\neg y_i \vee \neg z_i)$, and negative occurrences with the formula $(y_i \vee \neg z_i) \wedge (\neg y_i \vee z_i)$. After this substitution, the formula is an $\wedge \vee \wedge \vee$ formula, and so we re-write the formula as a CNF by distributing the innermost $\vee$ s over the $\wedge$ s.

To show there are efficient Res(2) refutations of $\text{CPLS}_n^{\text{bool}} \circ \oplus$, we prove the following theorem.

Theorem 8.4. *Suppose F is a width- w CNF formula with n variables and a length- s Resolution refutation. Then $F \circ \oplus$ has a size $O(2^w ns)$ Res(2) refutation.*

Proof. Let $\Pi = (C_1, \dots, C_s = \perp)$ be a length- s Resolution refutation of F . We show how to simulate this proof in Res(2) to get a refutation of $F \circ \oplus$. The basic idea is to derive for each clause C_i in F the 2-DNF encoding $C_i \circ \oplus$ directly.

Assume without loss of generality that all clauses of F are used in the refutation, otherwise we can discard any clauses that are not used. If $C = \bigvee_{i \in P} x_i \vee \bigvee_{j \in N} \neg x_j$ is a clause over the variables $x_1, \dots, x_n$, define the 2-DNF

$$\mathcal{X}(C) = \bigvee_{i \in P} (y_i \wedge \neg z_i) \vee (\neg y_i \wedge z_i) \vee \bigvee_{j \in N} (y_j \wedge z_j) \vee (\neg y_j \wedge \neg z_j).$$

Then $\mathcal{X}(C)$ directly encodes $C \circ \oplus$ as a 2-DNF. We prove by induction that, whenever the Resolution proof Π derives C_i , we can derive $\mathcal{X}(C_i)$ in Res(2).

There are two cases, depending on how C_i was derived. If C_i is a clause from F , then we begin by introducing all clauses in the CNF formula $C_i \circ \oplus$ in the Res(2) proof we are constructing in size at most $O(w2^w)$. We now need to go from $C_i \circ \oplus$ to $\mathcal{X}(C_i)$. We derive $\mathcal{X}(C_i)$ by first deriving the tautological DNF

$$\bigvee_{j \in P \cup N} ((y_j \wedge z_j) \vee (\neg y_j \wedge \neg z_j)) \vee ((y_j \wedge \neg z_j) \vee (\neg y_j \wedge z_j)),$$

which can be done by repeated uses of Axiom- and $\wedge$ -Introduction. Afterwards, we can perform the appropriate cuts with the clauses in $C_i \circ \oplus$ to derive $\mathcal{X}(C_i)$.

The second case is when C_i was derived from earlier clauses, denoted $C_j = A_j \vee x$ and $C_k = A_k \vee \neg x$, by the Resolution rule

$$A_j \vee x, A_k \vee \neg x \vdash A_j \vee A_k.$$

By induction, we can suppose we have derived $\mathcal{X}(C_j), \mathcal{X}(C_k)$, and we show how to derive $\mathcal{X}(C_i)$. We can write

$$\begin{aligned} \mathcal{X}(C_j) &= D_j \vee (y \wedge \neg z) \vee (\neg y \wedge z), \\ \mathcal{X}(C_k) &= D_k \vee (\neg y \wedge \neg z) \vee (y \wedge z), \end{aligned}$$

where $D_j = \mathcal{X}(A_j)$ and $D_k = \mathcal{X}(A_k)$ are 2-DNFs. We first derive from $\mathcal{X}(C_k)$ the two DNFs

$$D_k \vee (y \vee \neg z), \quad D_k \vee (\neg y \vee z)$$

as follows. Use an Axiom Introduction and a Weakening to derive $y \vee \neg y \vee \neg z$, which we then cut with $\mathcal{X}(C_k)$ to get $D_k \vee y \vee (\neg y \wedge \neg z)$. Then, use a second Axiom Introduction and Weakening to derive $z \vee \neg z \vee y$, which we cut with the resulting DNF to get $D_k \vee y \vee \neg z$. Repeat the same steps *mutatis mutandis* to obtain

the second DNF. Once we have the two DNFs above, we can cut them both with $\mathcal{X}(C_j)$ to obtain $\mathcal{X}(C_i)$. Thus this one Resolution step is replaced with $O(1)$ steps. This completes the proof of the induction.

It remains to measure the size of the refutation, but this is straightforward. The initial clauses of the Resolution proof are replaced with 2^w clauses from $F \circ \oplus$. Then, each intermediate clause of the proof C_i is replaced with a 2-DNF $\mathcal{X}(C_i)$ of size at most $O(|C_i|) = O(n)$, where $|C_i|$ is the number of symbols in C_i . An application of the cut rule is simulated by $O(1)$ steps in Res(2), each contributing $O(n)$ size at most. Hence the total size of the resulting proof is $O(2^w ns)$, as claimed. $\square$

The $\text{CPLS}_n^{\text{bool}}$ formula has width $O(\log n)$ and $n^{O(1)}$ -size Resolution refutations — the upper bound was shown by [KST07] when all pointers are encoded in unary, and by [DR23] for the binary encoding we use here. Combining this fact with the previous theorem proves the following:

Theorem 8.5. *There are polynomial-size Res(2) refutations of $\text{CPLS}_n^{\text{bool}} \circ \oplus$.*

It remains to prove that any dual-variable Sherali-Adams refutation of $\text{CPLS}_n^{\text{bool}} \circ \oplus$ requires size $2^{\Omega(n)}$. This follows a classic random restriction argument.

Theorem 8.6. *For any sufficiently large n , every dual-variable Sherali-Adams refutation of $\text{CPLS}_n^{\text{bool}} \circ \oplus$ requires size $2^{\Omega(n)}$.*

Proof. As this is a standard random restriction argument, we only sketch the proof. Consider a dual-variable Sherali-Adams refutation of $\text{CPLS}_n^{\text{bool}} \circ \oplus$ with S monomials. By standard techniques, we may assume that all polynomials in the proof are multilinear without loss of generality [FKP19].

Let $x_1, x_2, \dots, x_N$ be an enumeration of the variables of the underlying $\text{CPLS}_n^{\text{bool}}$ formula, which we note has width $O(\log n)$. Consider the following random restriction ρ : for each $i \in [N]$, with probability $1/2$ we set y_i to a uniformly random bit, and with probability $1/2$ we set z_i to a uniformly random bit. Then

$$\Pr_{\rho}[\text{the degree of the proof restricted by } \rho \text{ is } > \log_{4/3} 2S] < 1.$$

This is because if m is any monomial in the proof with $\deg(m) > \log_{4/3} 2S$, then

$$\Pr[m \upharpoonright \rho \neq 0] \leq (3/4)^{\deg(m)} \leq (3/4)^{\log_{4/3}(2S)} \leq \frac{1}{2S},$$

and hence a monomial m with $\deg(m) > \log_{4/3}(2S)$ exists after the random restriction with probability $\leq 1/2$ by a union bound. By the probabilistic method, this implies that there exists a restriction ρ such that the resulting proof has degree at most $\log_{4/3} 2S$. However, after any such restriction, we are left with a copy of $\text{CPLS}_n^{\text{bool}}$ with some of the input variables negated, and hence any Sherali-Adams refutation of the resulting formula must have degree at least $\Omega(n)$ by Theorem 8.1. Therefore $\log_{4/3} 2S = \Omega(n)$, implying $S = 2^{\Omega(n)}$, completing the proof of the theorem. $\square$

This implies Theorem 1.1, since CPLS_n has $O(n^3)$ variables.

Part III

Non-automatability of Sherali-Adams

This part is dedicated to proving [Theorem 1.2](#).

Theorem 1.2. *Automating Sherali-Adams is NP-Hard. More precisely, there is a polynomial-time algorithm R which, given an n -variate 3-CNF formula F , outputs a CNF formula $R(F)$ such that:*

- If F is satisfiable, then $R(F)$ has a polynomial-size Sherali-Adams refutation.
- If F is unsatisfiable, then every Sherali-Adams refutation of $R(F)$ requires size $2^{n^{\Omega(1)}}$.

As discussed in the introduction, we do this by first proving a Sherali-Adams degree lower bound for the formula $\text{LOSSY}(\text{SOD})$, which we then transfer and lift to a size lower bound for a formula based on $\text{REF}(F)$, from which non-automatability then follows. We therefore begin this portion of the paper by formally defining the formulas $\text{REF}(F)$ and $\text{LOSSY}(\text{SOD})$ in [Section 9](#), before showing how we derive non-automatability from a degree bound for $\text{LOSSY}(\text{SOD})$ in [Section 10](#). We conclude by proving the degree bound itself in [Section 11](#), building on the techniques and results from [Part I](#).

9 The Formulas and Reduction Preliminaries

9.1 The REF formula and Lossy(SoD)

We begin by defining the Refuter principle $\text{REF}(F)$ for Resolution.

Definition 9.1. Let $s \in \mathbb{N}$ be a positive size parameter and let $F = C_1 \wedge \dots \wedge C_m$ be a CNF formula over variables $x_1, \dots, x_n$, where $s = 2^{w_s} \geq m$ and $n = 2^{w_n}$ for some $w_s, w_n \in \mathbb{N}$. The formula $\text{REF}_s(F)$ takes as input the proof DAG of a (purported) length- s Resolution refutation Π of F , where each node D_i in the DAG is encoded by the following.

- For each literal ℓ over the variables $x_1, \dots, x_n$, a variable $\text{LIT}_{i,\ell} \in \{0, 1\}$, indicating whether D_i contains ℓ
- A derivation rule $\text{RULE}_i \in \{\text{AXIOM}, \text{WEAKENING}, \text{RESOLUTION}\}$
- Two pointers $\text{DERIV}_i^0, \text{DERIV}_i^1 \in [s]$, indicating the clauses from which D_i was derived
- A pointer $\text{RES}_i \in [n]$ indicating the resolved variable, if D_i was derived by the Resolution rule.
- A pointer $\text{CLAUSE}_i \in [m]$, indicating a clause of F which D_i is a weakening of, if D_i is claimed to be a weakening of a clause of F .

The constraints of the formula are as follows.

1. $\llbracket \text{RULE}_i = \text{AXIOM} \rrbracket \wedge \llbracket \text{CLAUSE}_i = j \rrbracket \rightarrow \llbracket \text{LIT}_{i,\ell} = 1 \rrbracket$
for all $i \in [s], j \in [m]$, and all literals $\ell \in C_j$
2. $\llbracket \text{DERIV}_i^b \neq j \rrbracket$
for all $j \leq i \in [s]$
3. $\llbracket \text{RULE}_i = \text{WEAKENING} \rrbracket \wedge \llbracket \text{DERIV}_i^0 = j \rrbracket \wedge \llbracket \text{LIT}_{j,\ell} \rrbracket \rightarrow \llbracket \text{LIT}_{i,\ell} \rrbracket$
for all $i \in [s], j < i$, and all literals ℓ
4. $\llbracket \text{RULE}_i = \text{RESOLUTION} \rrbracket \wedge \llbracket \text{DERIV}_i^0 = j \rrbracket \wedge \llbracket \text{RES}_i = k \rrbracket \wedge \llbracket \text{LIT}_{j,\ell} = 1 \rrbracket \rightarrow \llbracket \text{LIT}_{i,\ell} = 1 \rrbracket$
for all $i \in [s], j < i, k \in [n]$, and each literal $\ell \neq x_k$
5. $\llbracket \text{RULE}_i = \text{RESOLUTION} \rrbracket \wedge \llbracket \text{DERIV}_i^1 = j \rrbracket \wedge \llbracket \text{RES}_i = k \rrbracket \wedge \llbracket \text{LIT}_{j,\ell} = 1 \rrbracket \rightarrow \llbracket \text{LIT}_{i,\ell} = 1 \rrbracket$
for all $i \in [s], j < i, k \in [n]$, and each literal $\ell \neq \neg x_k$

6. $\llbracket \text{RULE}_i = \text{RESOLUTION} \rrbracket \wedge \llbracket \text{DERIV}_i^0 = j \rrbracket \wedge \llbracket \text{RES}_i = k \rrbracket \rightarrow \llbracket \text{LIT}_{j,x_k} = 1 \rrbracket$
for all $i \in [s]$, $j < i$, and $k \in [n]$
7. $\llbracket \text{RULE}_i = \text{RESOLUTION} \rrbracket \wedge \llbracket \text{DERIV}_i^1 = j \rrbracket \wedge \llbracket \text{RES}_i = k \rrbracket \rightarrow \llbracket \text{LIT}_{j,\neg x_k} = 1 \rrbracket$
for all $i \in [s]$, $j < i$, and $k \in [n]$
8. $\llbracket \text{LIT}_{s,\ell} = 0 \rrbracket$ for each literal ℓ

Constraint (1) enforces that $C_j \implies D_i$ whenever D_i was claimed to be a weakening of clause C_j , constraints (2 - 7) enforce that derivation steps are sound, and constraint (8) asserts that $D_s = \perp$.

Note that for simplicity and since the exact definition will not be relevant for us, we opt here for a binary encoding of the *non-relativized* version of $\text{REF}(F)$, in the terminology of [AM20]. This aligns with the definition of [LLR26], and while it differs from the relativized version used in [dRGN⁺21], it is easy to see that Sherali-Adams refutations of the relativized version can be converted to refutations of the non-relativized version without worsening the size or degree, by simply substituting 0 for all disabling variables. Importantly, this continues to be true for the lifted version we will define in Section 10.

Rather than working directly with $\text{REF}(F)$, for the bulk of this part we will instead be working with the $\text{LOSSY}(\text{SOD})$ formula, as described in the introduction. We define this formula formally now.

Definition 9.2. Let n and m be positive size parameters such that $n = 2^{w_n}$ and $m = 2^{w_m}$ for some $w_n, w_m \in \mathbb{N}$. The formula $\text{LOSSY}(\text{SOD})_{n,m}$ is defined as follows. We are concerned with a set of n *plaintexts*, and a set of $n/2$ *codes*. Each plaintext has an associated $[m] \times [m]$ grid of nodes. The variables specify disjoint inputs I_p to SOD_m on the grid attached to each plaintext $p \in [n]$, as well as an *encoding* pointer $f_{p,u} \in [n/2]$ for each SOD node $u \in [m] \times [m]$ and a *decoding* pointer $g_\sigma \in [n]$ for each code $\sigma \in [n/2]$. Intuitively, this specifies that the plaintext p can be encoded as $f_{p,u}$ for every node u in its associated SOD grid witnessing a clause of SOD falsified by I_p . In particular, every plaintext maps to at least one code, and could map to many. The clauses of the formula then assert that this encoding is uniquely inverted by the decoding pointers, which is a contradiction.

More formally, in addition to the encoding and decoding pointers, for each $p \in [n]$ we have variables $a_{p,u} \in \{0, 1\}$ for each $u \in [m] \times [m]$ and $s_{p,u} \in [m]$ for each $u \in [m-1] \times [m]$, specifying I_p . The clauses are:

1. $\llbracket a_{p,(1,1)} = 0 \rrbracket \wedge \llbracket f_{p,(1,1)} = \sigma \rrbracket \rightarrow \llbracket g_\sigma \neq p' \rrbracket$ for all $p \neq p' \in [n]$ and $\sigma \in [n/2]$.
2. $\llbracket a_{p,u} = 1 \rrbracket \wedge \llbracket s_{p,u} = i \rrbracket \wedge \llbracket a_{p,(r(u)+1,i)} = 0 \rrbracket \wedge \llbracket f_{p,u} = \sigma \rrbracket \rightarrow \llbracket g_\sigma \neq p' \rrbracket$
for all $u \in [m-1] \times [m]$, $i \in [m]$, $p \neq p' \in [n]$ and $\sigma \in [n/2]$.
3. $\llbracket a_{p,(m,i)} = 1 \rrbracket \wedge \llbracket f_{p,(m,i)} = \sigma \rrbracket \rightarrow \llbracket g_\sigma \neq p' \rrbracket$ for all $i \in [m]$, $p \neq p' \in [n]$ and $\sigma \in [n/2]$.

We will often handle parts of the input associated with different plaintexts separately, so we will separate each input to this formula into a tuple $(g, x_1, \dots, x_n)$, where g is a string of pointers $g_\sigma \in [n]$, indexed by $[n/2]$, and for each $p \in [n]$, x_p specifies the input to SOD_m and encoding pointers associated with the plaintext p . Extending the notation from Section 4, we view x_p as a triple of the form (a, π, f) , where a is a string of activity bits indexed by $[m] \times [m]$, π is a choice of row permutations, and f is a string of encoding pointers indexed by $[m] \times [m]$. For the decoding pointers, it will usually be most convenient to deal directly with the mapping encoded by g , so define $\phi_g : [n/2] \rightarrow [n]$ by $\phi_g : \sigma \mapsto g_\sigma$. Conversely, for any mapping $\phi : [n/2] \rightarrow [n]$, denote by g^ϕ the assignment to the decoding pointers which encodes ϕ , so that $g_\sigma^\phi = \phi(\sigma)$ for all $\sigma \in [n/2]$. Since we will be concerned with which plaintexts have matching encodings, we say that $p \in [n]$ is *matched with σ by g* if $g_\sigma = p$, and simply *matched by g* if $p \in \text{im}(\phi_g)$.

For convenience we will denote by $\mathcal{I}_m = \{0, 1\}^{m^2} \times [m]^{m(m-1)}$ the range of the original variables of SOD_m , and we denote the range of each x_p (recall this includes both the SOD variables and encoding pointers) by $\mathcal{I}_m^{\text{enc}} = \mathcal{I}_m \times [n/2]^{m^2}$.

As mentioned in an earlier footnote, this differs slightly from the similar problem studied by [LLR26], which they denote by RWPHP(PLS). Where our definition uses the grid-based SOD principles as its source of indirection, theirs instead uses the classical Iteration principle, ITER. These formulas are easily seen to be irreducible by constant-depth decision trees (in the sense we will make formal in the next subsection), so this will not impact our bounds.

9.2 Node-wise Reductions and Pseudoexpectations

In order to translate our degree bounds for LOSSY(SOD) into size bounds for $\text{REF}(F)$, we will need to target a looser notion of degree than we have used so far, tailored specifically to the REF formula.

Definition 9.3. For any $s \in \mathbb{N}$ and any CNF formula F , let X_s^F denote the set of multivalued variables of $\text{REF}_s(F)$. We view this set as being implicitly divided into proof *nodes* $D_1, \dots, D_s$ in the natural way. Then for $k \in \mathbb{N}$, let $S_{\leq k}$ denote the set of partial assignments to X_s^F which assign non- $*$ values to variables associated with at most k distinct nodes D_i and define

$$\mathcal{A}_{\leq k}^{\text{node}}(X_s^F) := \text{span} \{ \chi_\sigma \mid \sigma \in S_{\leq k} \}$$

Similarly to block degree, for any indicator polynomial $p \in \mathcal{A}(X_s^F)$, the *node degree* of p is

$$\text{ndeg}(p) := \min \left\{ d : p \in \mathcal{A}_{\leq d}^{\text{node}}(X_s^F) \right\}$$

We may also define versions of indicator designs and pseudoexpectations for this notion of degree, analogously to those already defined for block degree.

Definition 9.4. Let $s \in \mathbb{N}$ and let F be any CNF formula. Again let X_s^F denote the set of variables of $\text{REF}_s(F)$. A *node-degree- d indicator design* for $\text{REF}_s(F)$ is a functional $D : \mathcal{A}_{\leq d}^{\text{node}}(X_s^F) \rightarrow \mathbb{R}$ satisfying:

- *Normalization.* $D[1] = 1$
- *Linearity.* $D[\alpha p + \beta q] = \alpha D[p] + \beta D[q]$ for all $p, q \in \mathcal{A}_{\leq d}^{\text{node}}(X_s^F)$ and all $\alpha, \beta \in \mathbb{R}$.
- *REF-Nonwitnessing.* $D[C] = 0$ for every clause C_0 of $\text{REF}_s(F)$ and every conjunction $C \Rightarrow \overline{C}_0$ with $\text{ndeg}(C) \leq d$.

A *node-degree- d indicator pseudoexpectation* $\tilde{\mathbb{E}}$ is a node-degree- d indicator design for which the following additionally holds:

- *Nonnegativity.* $\tilde{\mathbb{E}}[C] \geq 0$ for every conjunction C with $\text{ndeg}(C) \leq d$.

To obtain this form of pseudoexpectation for $\text{REF}(F)$, we will use a particular form of decision tree reduction. For this we define a form of decision tree suitable for working with multivalued variables.

Definition 9.5. A *block-wise decision tree* over a set of multivalued variables $X = (x_1, S_1), \dots, (x_n, S_n)$ is a pair $(T, \mathcal{O})$, where $\mathcal{O}$ is some set, which we call the range of the decision tree, and T is a rooted tree. Each internal node of T is labelled by some variable x_i from X and, for each $v \in S_i$, has exactly one outgoing edge labelled by the indicator $\llbracket x_i = v \rrbracket$. We require that no node labels are repeated along any root-leaf branch of T . Each leaf node is labelled by some output value $o \in \mathcal{O}$. On input $x \in S_1 \times \dots \times S_n$, we evaluate T in the natural way, following the unique edge labelled by the indicator satisfied by x at each node, eventually reaching some leaf node and outputting its label.

We identify each branch π of T with the set of indicators labelling its edges. We denote by $\text{Br}(T)$ the set of all root-leaf branches of T , and for $o \in \mathcal{O}$ we denote by $\text{Br}_o(T)$ the set of branches ending in a leaf labelled by o . For a branch $\pi = \{\llbracket x_i = v_i \rrbracket\}_{i \in S}$ of T , we define $C_\pi = \prod_{i \in S} \llbracket x_i = v_i \rrbracket$ to be the conjunction which is 1 exactly when π is followed when evaluating T . When the range is unimportant or clear from context we will often omit it and simply refer to the tree as T .

The *block depth* of T is the maximum length of its branches.

Using these decision trees, we define the following form of reduction between indicator CNFs, based on the block-aware reductions of [dRGN⁺21]. These are in turn based on the standard notion of decision tree reductions between *false-clause* search problems, as seen in e.g. [GHJ⁺24], but are tailored specifically to relating block degree and node degree.

Definition 9.6. Let $F = C_1 \wedge \dots \wedge C_m$ be an indicator CNF over the variables $X = (x_1, S_{x_1}), \dots, (x_n, S_{x_n})$, and $G = D_1 \wedge \dots \wedge D_{m'}$ an indicator CNF over the variables $Y = (y_1, S_{y_1}), \dots, (y_{n'}, S_{y_{n'}})$. A *block-wise reduction* from F to G consists of a block-wise decision tree (T_i, S_{y_i}) over X for each $i \in [n']$, outputting a reduced value for the variable y_i , as well as a return decision tree $(T_j^{ret}, [m])$ over X for each $j \in [m']$, outputting the index of some clause C_i of F . These must satisfy that for any $a \in S_{x_1} \times \dots \times S_{x_n}$ and $j \in [m']$, if $T_j^{ret}(a) = i$ then

$$C_i(a) \implies D_j(T_1(a), \dots, T_{n'}(a))$$

The block depth of the reduction is the maximum block depth of $T_1, \dots, T_{n'}$ and $T_1^{ret}, \dots, T_{m'}^{ret}$. If $G = \text{REF}_s(H)$ for some $s \in \mathbb{N}$ and some CNF formula H , we also call this a *node-wise* reduction if for each node D_i , $i \in [s]$, the trees outputting values for the variables associated with D_i differ only on the labels of their leaf nodes. In other words, in a node-wise reduction we may combine the decision trees for each variable associated with D_i into a single decision tree outputting values for all variables of D_i at once, without increasing the block depth.

The main takeaway from block-wise reductions is that they permit us to convert pseudoexpectations for the source CNF F into pseudoexpectations for the target CNF G , only mildly reducing the degree of the resulting pseudoexpectation. Importantly for us, for node-wise reductions this conversion translates block-degree into node-degree. This will be crucial for lifting the resulting degree bound for $\text{REF}(F)$.

Lemma 9.7. Let $F = C_1 \wedge \dots \wedge C_m$ again be an indicator CNF over the variables $X = (x_1, S_{x_1}), \dots, (x_n, S_{x_n})$, and $G = D_1 \wedge \dots \wedge D_{m'}$ an indicator CNF over the variables $Y = (y_1, S_{y_1}), \dots, (y_{n'}, S_{y_{n'}})$. Suppose there is both a block-depth- k reduction from $S(F)$ to $S(G)$, and a block-degree- d indicator pseudoexpectation for F . Then there is also a block-degree- $((d - k)/k)$ indicator pseudoexpectation for G . Moreover, if $G = \text{REF}_s(H)$ for some $s \in \mathbb{N}$ and CNF formula H and the reduction from F to $\text{REF}_s(H)$ is node-wise, then there is a node-degree- $((d - k)/k)$ indicator pseudoexpectation for $\text{REF}_s(H)$.

Proof. Let $\tilde{\mathbb{E}}$ be the block-degree- d pseudoexpectation for F , and $T_1, \dots, T_{n'}, T_1^{ret}, \dots, T_{m'}^{ret}$ the block-depth- k , block-wise reduction from F to G . Intuitively, we would like to choose pseudoexpectation values for each indicator polynomial p over Y by “substituting” these decision trees into p , converting it into a polynomial over X . From there it suffices to simply imitate the pseudoexpectation for F . More formally, define the translation R of indicator polynomials over Y to indicator polynomials over X recursively as follows.

- $R[1] = 1$
- $R[0] = 0$
- For $i \in [n']$ and $v \in S_{y_i}$, $R[\llbracket y_i = v \rrbracket] = \sum_{\pi \in \text{Br}_v(T_i)} C_\pi$
- For all indicator polynomials p and $\alpha \in \mathbb{R}$, $R[\alpha p] = \alpha R[p]$.
- For all indicator polynomials p and q ,
 - $R[p \cdot q] = R[p] \cdot R[q]$, and
 - $R[p + q] = R[p] + R[q]$

To see that this is well-defined, it suffices to show that it preserves the identities of the indicator algebra of Y . We begin by showing that it preserves $1 - \sum_{v \in S_i} \llbracket y_i = v \rrbracket = 0$ for all $i \in [n']$. Under R , this becomes

$$R \left[\sum_{v \in S_i} \llbracket y_i = v \rrbracket \right] = \sum_{v \in S_i} \sum_{\pi \in \text{Br}_v(T_i)} C_\pi = \sum_{\pi \in \text{Br}(T_i)} C_\pi = 1$$

For the identities of the form $\llbracket y_i = v \rrbracket \llbracket y_i = v \rrbracket - \llbracket y_i = v \rrbracket = 0$, we have

$$\begin{aligned} R[\llbracket y_i = v \rrbracket \llbracket y_i = v \rrbracket - \llbracket y_i = v \rrbracket] &= \left(\sum_{\pi \in \text{Br}_v(T_i)} C_\pi \right)^2 - \sum_{\pi \in \text{Br}_v(T_i)} C_\pi \\ &= \sum_{\pi \in \text{Br}_v(T_i)} C_\pi^2 + \sum_{\pi \neq \tau \in \text{Br}_v(T_i)} C_\pi C_\tau - \sum_{\pi \in \text{Br}_v(T_i)} C_\pi \\ &= \sum_{\pi \in \text{Br}_v(T_i)} C_\pi - \sum_{\pi \in \text{Br}_v(T_i)} C_\pi = 0 \end{aligned}$$

where we used the fact that exactly one branch of a decision tree is followed on any input. Finally, for each identity of the form $\llbracket y_i = v \rrbracket \llbracket y_i = v' \rrbracket = 0$, $v \neq v' \in S_i$, we have

$$\begin{aligned} R[\llbracket y_i = v \rrbracket \llbracket y_i = v' \rrbracket] &= \left(\sum_{\pi \in \text{Br}_v(T_i)} C_\pi \right) \left(\sum_{\tau \in \text{Br}_{v'}(T_i)} C_\tau \right) \\ &= \sum_{\pi \in \text{Br}_v(T_i), \tau \in \text{Br}_{v'}(T_i)} C_\pi C_\tau \end{aligned}$$

Again using the fact that exactly one branch of a decision tree can be followed, each term of this sum is 0, so all identities are indeed preserved.

Using this translation, we define the reduced pseudoexpectation $\tilde{\mathbb{E}}_R$ by $\tilde{\mathbb{E}}_R[p] = \tilde{\mathbb{E}}[R[p]]$ and claim that this is a block-degree- $((d-k)/k)$ pseudoexpectation for G . Linearity follows immediately by linearity of $\tilde{\mathbb{E}}$ and R , non-negativity follows from non-negativity of $\tilde{\mathbb{E}}$, and clearly $\tilde{\mathbb{E}}_R[1] = 1$. To see that this is also non-witnessing, choose any $i \in [m']$ and any indicator conjunction C over Y such that $\text{bdeg}(\overline{D}_i C) \leq (d-k)/k$. By definition we have $\tilde{\mathbb{E}}_R[\overline{D}_i C] = \tilde{\mathbb{E}}[R[\overline{D}_i C]]$, and $\text{bdeg}(R[\overline{D}_i C]) \leq d-k$. Moreover, once more using the fact that exactly one branch of a decision tree is followed on any input, we can expand this to

$$\sum_{\pi \in \text{Br}(T_i^{\text{ret}})} C_\pi \cdot R[\overline{D}_i C]$$

which has block degree at most d , since T_i^{ret} has depth at most k .

We would like to show that each term of this sum is mapped to 0 by $\tilde{\mathbb{E}}$, so fix any branch $\pi \in \text{Br}(T_i^{\text{ret}})$, and suppose that π ends in a leaf labelled by some $j \in [m]$. Notice that for any assignment a to the variables of F , if $R[\overline{D}_i](a) = 1$ and $C_\pi(a) = 1$, then $\overline{D}_i(T_1(a), \dots, T_{n'}(a)) = 1$ and $T_i^{\text{ret}}(a) = j$. Since $T_1, \dots, T_{n'}, T_1^{\text{ret}}, \dots, T_{m'}^{\text{ret}}$ was a valid reduction, we must also have $\overline{C}_j(a) = 1$. Therefore $C_\pi \cdot R[\overline{D}_i C]$ witnesses C_j , so since $\tilde{\mathbb{E}}$ was non-witnessing for F and $\text{bdeg}(C_\pi \cdot R[\overline{D}_i C]) \leq d$, we have $\tilde{\mathbb{E}}[C_\pi \cdot R[\overline{D}_i C]] = 0$. Applying this to all terms of the above sum and using linearity, $\tilde{\mathbb{E}}_R[\overline{D}_i C]$ is also 0.

The fact that this is also a node-degree- $((d-k)/k)$ pseudoexpectation for node-wise reductions to $G = \text{REF}_s(H)$ follows analogously, using the following straightforward claim.

Claim 9.8. For any indicator conjunction C over the variables of $\text{REF}_s(H)$, $\text{bdeg}(R[C]) \leq k \cdot \text{ndeg}(C)$.

We show this by induction on $\text{ndeg}(C)$. When $\text{ndeg}(C) = 0$, we must have $C = 1$ or $C = 0$ and the claim trivially holds. Otherwise, suppose the claim holds for all C' with $\text{ndeg}(C') < w$ for some w , and suppose $\text{ndeg}(C) = w$. Fix any node D_j for which C queries some associated variable, and decompose C as $C_0 C_j$, where C_j is the sub-conjunction containing all indicators over variables associated with D_j , and C_0 is the remaining sub-conjunction with $\text{ndeg}(C_0) = w-1$. By the inductive hypothesis we have $\text{bdeg}(R[C_0]) \leq k \cdot (w-1)$, so it remains to show that $\text{bdeg}(R[C_j]) \leq k$. Let $y_{i_1}, \dots, y_{i_b}$ be the variables

queried by C_j , where $b = \text{bdeg}(C_j)$, so that $C_j = \llbracket y_{i_1} = v_1 \rrbracket \cdots \llbracket y_{i_b} = v_b \rrbracket$ for some choices of values $v_1 \in S_{y_{i_1}}, \dots, v_b \in S_{y_{i_b}}$. Then

$$R[C_j] = \prod_{\ell \in [b]} \left[\sum_{\pi \in \text{Br}_{v_\ell}(T_{i_\ell})} C_\pi \right] = \sum_{\substack{\pi_1 \in \text{Br}_{v_1}(T_{i_1}) \\ \vdots \\ \pi_b \in \text{Br}_{v_b}(T_{i_b})}} C_{\pi_1} \cdots C_{\pi_b}$$

But recall that the reduction was node-wise, so $T_{i_1}, \dots, T_{i_b}$ differ only on the labels of leaves, and for each term of this sum we can treat $\pi_1, \dots, \pi_b$ as coming from a single tree T of depth at most k , whose leaves are labelled by values in $S_1 \times \cdots \times S_\ell$. Therefore since exactly one branch of a decision tree can be followed on any input, the only case where $C_{\pi_1} \cdots C_{\pi_b} \neq 0$ is when $\pi_1 = \cdots = \pi_b$, leaving us with

$$R[C_j] = \sum_{\pi \in \text{Br}_{(v_1, \dots, v_\ell)}(T)} C_\pi$$

which has block degree at most k . □

10 Non-Automatability from Lossy(SoD) Pseudoexpectations

In the next section we will see a proof of the following theorem, stating that $\text{LOSSY}(\text{SoD})$ admits a polynomial-degree indicator pseudoexpectation.

Theorem 10.1. *There is a degree- $\Omega(\min\{n, m^{1/3}\})$ indicator pseudoexpectation for $\text{LOSSY}(\text{SoD})_{n,m}$.*

In the meantime we dedicate this section to showing how the existence of such a pseudoexpectation implies non-automatability for Sherali-Adams. As discussed in the introduction, we follow the Atserias-Müller program for proving non-automatability [AM20]. Namely, we will show that for any formula F there is a formula $R(F)$, constructible in polynomial time, which has polynomial-size Sherali-Adams refutations exactly whenever F is satisfiable. Therefore if Sherali-Adams were automatable, we could decide the satisfiability of F by constructing $R(F)$, then running the automating algorithm up to some polynomial time bound to determine whether it terminates in time. In our case, in the same vein as [dRGN⁺21], $R(F)$ will be the following lifted version of $\text{REF}(F)$.

Definition 10.2. Let F be a CNF formula and $s \in \mathbb{N}$ a size parameter. The boolean formula $\text{Lift}_s(F)$ can be obtained from $\text{REF}_s^{\text{bool}}(F)$ as follows. Let $x_{1,1}, \dots, x_{s,w}$ be the (boolean) variables of $\text{REF}_s^{\text{bool}}(F)$, where variable $x_{i,j}$ is associated with the node D_i . For each variable $x_{i,j}$ of $\text{REF}_s^{\text{bool}}(F)$, we introduce two new *lifted* variables $x_{i,j}^0$ and $x_{i,j}^1$, and for each node D_i , $i \in [s]$, we introduce a *selector* variable b_i . The idea is that b_i chooses, simultaneously for all variables $x_{i,j}$ associated with D_i , whether the value of $x_{i,j}^0$ or $x_{i,j}^1$ should be used when evaluating the constraints of $\text{REF}_s^{\text{bool}}(F)$. More formally, define the selector formula

$$g(b_i, x^0, x^1) := (\neg b_i \wedge x^0) \vee (b_i \wedge x^1),$$

which evaluates to x^0 if $b_i = 0$, and x^1 if $b_i = 1$. For each clause C of $\text{REF}_s^{\text{bool}}(F)$, define the composition of C with g as

$$C \circ g := C(g(b_1, x_{1,1}^0, x_{1,1}^1), \dots, g(b_1, x_{1,w}^0, x_{1,w}^1), \dots, g(b_s, x_{s,w}^0, x_{s,w}^1)).$$

Then the lifted formula is

$$\text{Lift}_s(F) = \bigwedge_{C \in \text{REF}_s^{\text{bool}}(F)} C \circ g$$

where we rewrite the result in CNF by distributing $\vee$ over the innermost $\wedge$.

The upper bound when F is satisfiable is already known, by composing the Resolution upper bound from [dRGN⁺21] with the standard simulation of Resolution by Sherali-Adams.

Theorem 10.3. [dRGN⁺21] *Let F be a satisfiable CNF formula, and let $s \in \mathbb{N}$. Then there is a size- $s^{O(1)}$ Sherali-Adams refutation of $\text{Lift}_s(F)$.*

The new contribution of this work is the size lower bound when F is unsatisfiable, for which we use the pseudoexpectation for LOSSY(SOD). To turn this pseudoexpectation into a size bound for $\text{Lift}_s(F)$, we first convert it into a pseudoexpectation for REF via the following reduction, due to [LLR26].

Theorem 10.4. [LLR26] *Let F be an unsatisfiable CNF formula on ℓ variables, and let n and m be positive integers. There is some size bound $s_0 = O(\ell nm + |F|)$ such that, for every $s \geq s_0$, there is a block-depth- $O(\ell)$, node-wise reduction from $\text{LOSSY}(\text{SOD})_{n,m}$ to $\text{REF}_s(F)$.*

Note that Li, Li, and Ren do not state this as a node-wise reduction, but it can be readily verified that their reduction also satisfies this statement. Their statement also requires that $\text{REF}_s(F)$ is unsatisfiable, but $\text{REF}_s(F)$ still admits a trivial reduction whenever it is satisfiable so we combine the cases for simplicity.

Combining this with Lemma 9.7 and Theorem 10.1, we obtain the following corollary.

Corollary 10.5. *For any unsatisfiable CNF formula F , there is some size bound $s_0 = |F|^{O(1)}$ such that, for all $s \geq s_0$, there is a node-degree- $\Omega(|F|)$ indicator pseudoexpectation for $\text{REF}_s(F)$.*

Proof. Set n to be the first power of 2 such that $n \geq |F|^2$, and set $m = n^3$. By Theorem 10.1, there is a degree- $\Omega(|F|^2)$ indicator pseudoexpectation for $\text{LOSSY}(\text{SOD})_{n,m}$. Applying Lemma 9.7 to this pseudoexpectation and the reduction promised by Theorem 10.4, we obtain a bound $s_0 = |F|^{O(1)}$ and a node-degree- d indicator pseudoexpectation for $\text{REF}_s(F)$ for all $s \geq s_0$, where $d = \Omega\left(\frac{\min\{n, m^{1/3}\} - |F|}{|F|}\right) = \Omega(|F|)$. $\square$

From here, we can lift this to a size bound for $\text{Lift}_s(F)$ using a standard random restriction argument, similarly to Theorem 8.6.

Theorem 10.6. *There is some size bound $s_0(n) = n^{O(1)}$ such that, for any unsatisfiable CNF formula F and all $s \geq s_0(|F|)$, every dual-variable Sherali-Adams refutation of $\text{Lift}_s(F)$ requires size $2^{\Omega(|F|)}$.*

Proof. As with Theorem 8.6, we only give a sketch as the proof is again standard. Let Π be a dual-variable Sherali-Adams refutation of $\text{Lift}_s(F)$ with S monomials, and let $x_{1,1}, \dots, x_{s,w}$ denote the variables of $\text{REF}_s(F)$, where each variable $x_{i,j}$ is associated with the node D_i as usual. Sample a random restriction ρ as follows:

1. Sample a choice of selector bits $(\rho(b_1), \dots, \rho(b_s)) \in \{0, 1\}^s$ uniformly at random.
2. Independently for each $i \in [s]$:
 - If $\rho(b_i) = 0$, set $\rho(x_{i,j}^0) = *$ and sample $\rho(x_{i,j}^1) \in \{0, 1\}$ independently and uniformly for each j .
 - If $\rho(b_i) = 1$, instead set $\rho(x_{i,j}^1) = *$ and sample each $\rho(x_{i,j}^0) \in \{0, 1\}$ uniformly at random.

We now claim the following.

$$\Pr_{\rho}[\text{any monomial } m \text{ of } \Pi \upharpoonright_{\rho} \text{ queries more than } \log_{4/3} 2S \text{ nodes}] < 1$$

As with indicator polynomials, we use $\text{ndeg}(m)$ to mean the number of nodes for which any associated variable is queried by the monomial m . Then if $\text{ndeg}(m) > \log_{4/3} 2S$, since every variable (including dual

variables) is set to 0 by ρ with probability at least $1/4$ and this event is independent for variables associated with distinct nodes,

$$\Pr[m \upharpoonright_\rho \neq 0] \leq (3/4)^{\deg(m)} \leq (3/4)^{\log_{4/3} 2S} \leq \frac{1}{2S}$$

so by a union bound over all such monomials in Π , the probability that any survive the restriction is at most $1/2$. Therefore there is some restriction ρ such that every monomial m of $\Pi \upharpoonright_\rho$ queries at most $\log_{4/3} 2S$ nodes. Furthermore, $\text{Lift}_s(F) \upharpoonright_\rho$ is simply a copy of $\text{REF}_s^{\text{bool}}(F)$ up to renaming variables. Since [Corollary 10.5](#) promised a node-degree- $\Omega(|F|)$ indicator pseudoexpectation for $\text{REF}_s(F)$, we must have $\log_{4/3} 2S = \Omega(|F|)$, so $S = 2^{\Omega(|F|)}$. $\square$

With both upper and lower bounds in hand, we conclude this section by proving [Theorem 1.2](#).

Proof of Theorem 1.2. Let $s_\ell(n)$ be the explicit size bound from [Theorem 10.6](#), and let $S_\ell(n)$ be the smallest power of 2 for which $S_\ell(n) \geq c_\ell n^{k_\ell}$, where $c_\ell, k_\ell \in \mathbb{N}$ are the smallest constants for which $s_\ell(n) \leq c_\ell n^{k_\ell}$ for sufficiently large n . Similarly, let $s_u(n)$ be the refutation size upper bound from [Theorem 10.3](#), and let $S_u(n) = c_u n^{k_u}$, where $k_u, c_u \in \mathbb{N}$ are the first constants such that $s_u(n) \leq c_u n^{k_u}$ for all sufficiently large n . We choose R to be the algorithm which simply constructs and outputs $\text{Lift}_{S_\ell(|F|)}(F)$.

We already have that $R(F)$ satisfies the two properties promised by the theorem, by [Theorems 10.3](#) and [10.6](#). To see why this implies NP-hardness of automating Sherali-Adams, suppose there were an efficient automating algorithm A . That is, on input F , A outputs a Sherali-Adams refutation $A(F)$ of F within some time bound $t_A(s_F) = s_F^{O(1)}$, where s_F denotes the size of the smallest Sherali-Adams refutation of F . Similarly to the other bounds, let $T_A(n) = c_A n^{k_A}$, where $c_A, k_A \in \mathbb{N}$ are the smallest constants such that $t_A(n) \leq T_A(n)$ for all n sufficiently large. Let $n_0 \in \mathbb{N}$ be the smallest input length for which $\text{REF}_{S_\ell(|F|)}(F)$ requires super-polynomial-size Sherali-Adams refutations and $s_\ell(n) \leq S_\ell(n)$, $s_u(n) \leq S_u(n)$, and $t_A(n) \leq T_A(n)$ for all $n \geq n_0$. Then we can decide the satisfiability of F as follows. If $|F|, S_\ell(|F|)$, or $S_u(S_\ell(|F|)) < n_0$, we simply brute-force all assignments to the variables of F . Otherwise we do the following.

1. Construct $R(F) = \text{Lift}_{S_\ell(|F|)}(F)$
2. Run A on input $R(F)$ for $T_A(S_u(S_\ell(|F|))) = c_A c_u^{k_A} c_\ell^{k_u k_A} |F|^{k_A k_u k_\ell}$ steps.
 - If A outputs a valid refutation within this time bound, accept.
 - If A has yet to terminate (or terminated without producing a refutation), reject.

Clearly this runs in time polynomial in $|F|$, and the correctness of this algorithm follows from the established size bounds for $R(F)$: if F were satisfiable, A must terminate in time since we have an upper bound of $s_u(S_\ell(|F|))$ on the size required to refute $\text{Lift}_{S_\ell(|F|)}(F)$ in Sherali-Adams. Otherwise, by the lower bound A does not even have time to write an entire refutation of $R(F)$ into memory, so A certainly cannot terminate in time. $\square$

11 The Lossy(SoD) Pseudoexpectation

11.1 Intermediate Measures for SoD

Rather than defining a new sequence of distributions to interpolate as before, our strategy for LOSSY(SOD) will be to instead define a new measure based on independent copies of the measure μ we already constructed for SOD. For the remainder of this section we will therefore take $\mathcal{D}_k$ to be as defined in [Section 4](#). Before specifying the full measure, however, we first define several intermediate measures and prove some crucial relationships between them.

While the simple choice of interpolation nodes $t_i = i$ was sufficient in [Part I](#), the argument in this section will crucially require that the negative Lagrange coefficients shrink with increasing d , where d will be the target block degree of the pseudoexpectation. To satisfy this, we begin by redefining μ itself with more carefully chosen interpolation nodes.

Definition 11.1. Let $d = \lfloor \min \{ \frac{1}{4}n, \frac{1}{20}m^{1/3} \} \rfloor$ and define $N = d + 1$. Define $t_0 = 1$, and for $i = 1, \dots, N$, define $t_i = 16di^2$. Set

$$\gamma_0 = \ell_0(0) = \prod_{j>0} \frac{t_j}{t_j - 1} = \prod_{j>0} \frac{16dj^2}{16dj^2 - 1}$$

and for $i = 1, \dots, N$ we set

$$\gamma_i = \ell_i(0) = \prod_{j \neq i} \frac{t_j}{t_j - t_i} = -\frac{1}{16di^2 - 1} \prod_{0 < j \neq i} \frac{j^2}{j^2 - i^2}.$$

Using these, we define the interpolated measure as

$$\mu(x) = \sum_{i=0}^N \gamma_i \mathcal{D}_{t_i}(x)$$

As we have discussed in the introduction, the quadratic spacing of nodes is inspired by [\[BT22\]](#), and will allow us to bound the magnitude of the contribution from the negative γ_i coefficients.

Proposition 11.2. *If $\alpha = \sum_{i: \gamma_i < 0} |\gamma_i|$, then $\alpha \leq \frac{1}{4d}$.*

Proof. By definition, $\gamma_i < 0$ only if i is odd, and for all odd i we have

$$\gamma_i = -\frac{1}{16di^2 - 1} \prod_{0 < j \neq i} \frac{j^2}{j^2 - i^2} = (-1)^i \frac{1}{16di^2 - 1} \cdot \frac{2(N!)^2}{(N-i)!(N+i)!}.$$

The third factor is ≤ 2 , and so

$$\begin{aligned} \alpha &= \sum_{\substack{i \leq N \\ i \text{ odd}}} |\gamma_i| \leq \sum_{i=1}^N \frac{2}{16di^2 - 1} \leq \frac{1}{4d} \sum_{i=1}^N \frac{2}{4i^2 - 1} \\ &= \frac{1}{4d} \sum_{i=1}^N \left(\frac{1}{2i-1} - \frac{1}{2i+1} \right) = \frac{1}{4d} \left(1 - \frac{1}{2(N+1)} \right) \leq \frac{1}{4d}. \end{aligned} \quad \square$$

The choice of $d \leq (1/20)m^{1/3}$ in this definition is purely to ensure that $\mathcal{D}_{t_i}$ is well-defined for all $i \leq d+1$: if $d \leq (1/20)m^{1/3}$, then $t_i \leq 16d(d+1)^2 < m$ for sufficiently large m . Aside from this, our only requirement on d in terms of m is that $d \leq m/100$, so that we can appeal to the results of [Section 4.1](#). The restriction to $d \leq n/4$, however, will be necessary for our proof of non-negativity.

Since the variables of LOSSY(SOD) additionally specify an encoding pointer for each node of the SOD grids, our intermediate measures will incorporate these into μ in order to extend it to a measure on $\mathcal{I}_m^{\text{enc}}$. To this end, we define several distributions which extend $\mathcal{D}_k$ to sample values for both the variables of SOD and the encoding pointers of each node in the $[m] \times [m]$ grid.

Definition 11.3. For $m \in \mathbb{N}$, define the following two families of distributions on $\mathcal{I}_m^{\text{enc}}$.

For any choice of code $\sigma \in [n/2]$ let $\mathcal{D}_{k,\sigma}$ be the distribution defined by the following sampling procedure:

1. Sample $x_0 = (a, \pi) \sim \mathcal{D}_k$.
2. For any node $u \in [m] \times [m]$ which witnesses that x_0 falsifies SOD, set $f_u = \sigma$.
3. Independently for each non-witness u , sample a uniformly random encoding $f_u \in [n/2]$.

In particular, since there are exactly k witnesses for each x_0 in the support of $\mathcal{D}_k$, for any x in the support of $\mathcal{D}_{k,\sigma}$ we have

$$\mathcal{D}_{k,\sigma}(x) = (n/2)^{-(m^2-k)} \mathcal{D}_k(x)$$

For an integer $k > 0$ and $x \in \mathcal{I}_m^{enc}$, additionally define

$$\mathcal{D}_{k,*}(x) = (n/2)^{-m^2} \mathcal{D}_k(x)$$

so that $x = (a, \pi, f) \sim \mathcal{D}_{k,*}$ is sampled as follows:

1. Sample $(a, \pi) \sim \mathcal{D}_k$.
2. Independently for each $u \in [m] \times [m]$, sample a uniformly random value $f_u \in [n/2]$.

The reason for defining two separate families of distributions is that, roughly speaking, the final pseudodistribution will work by first sampling a choice of decoding function ϕ , and then sampling values for x_p , for each plaintext $p \in [n]$, according to whether p was matched by ϕ . If p was matched to some σ by ϕ , then the associated encoding pointers should agree with ϕ , so we use $\mathcal{D}_{k,\sigma}$. Otherwise there is no canonical choice for the encoding pointers, so we use $\mathcal{D}_{k,*}$ to ensure that all encoding pointers are independent. Of course, if we sampled directly from these distributions the result would not be non-witnessing, and we will need an extra trick to ensure non-negativity, so instead we interpolate these distributions to obtain the following measures on $\mathcal{I}_m^{enc}$.

Definition 11.4. For any code $\sigma \in [n/2]$, define

$$\mu_\sigma(x) = \sum_{i=0}^N \gamma_i \mathcal{D}_{t_i,\sigma}(x)$$

and similarly define

$$\mu_*(x) = \sum_{i=0}^N \gamma_i \mathcal{D}_{t_i,*}(x) = (n/2)^{-m^2} \mu(x).$$

Additionally, let $\alpha = \sum_{i:\gamma_i < 0} |\gamma_i|$. We capture the negative contribution of μ_σ by the normalized measure

$$\mu_\sigma^-(x) = \frac{1}{\alpha} \sum_{i:\gamma_i < 0} |\gamma_i| \mathcal{D}_{t_i,\sigma}(x)$$

It follows from our choice of normalization that $\sum_x \mu_\sigma^-(x) = 1$, and clearly $\mu_\sigma^-(x) \geq 0$ for all x , so μ_σ^- defines a true distribution on $\mathcal{I}_m^{enc}$. We also immediately have that $\mu_\sigma(x) \geq -\alpha \mu_\sigma^-(x)$ for all x , giving us a grasp on the magnitude of the negative coordinates of μ_σ . We will later show in [Lemma 12.9](#) that this observation generalizes to a bound on $\mathbf{E}_{\mu_*}[C]$ in terms of $\mathbf{E}_{\mu_\sigma^-}[C]$, for all sufficiently low-degree conjunctions C , which will be key to proving that our pseudoexpectation is non-negative.

11.2 The Final Construction for Lossy(SoD)

Now that we have defined our intermediate measures, we can finally define the combined measure Λ and the pseudoexpectation it yields. Intuitively, Λ will correspond to the following “sampling” procedure:

1. Sample an injective mapping $\phi : [n/2] \rightarrow [n]$ uniformly at random, and assign the decoding pointers according to g^ϕ .
2. Independently for each plaintext p :
 - If ϕ matches p to some σ , sample $x_p \sim \mu_\sigma^-$, to be consistent with ϕ .
 - Otherwise if p is unmatched in ϕ , “sample” x_p according to μ_* .

Of course, this is not a valid sampling procedure, as it requires us to “sample” according to the signed measure μ_* . Therefore we instead give a direct definition of Λ which captures this idea.

Definition 11.5. Let $\phi : [n/2] \rightarrow [n]$ be a choice of injective decoding function and $x_1, \dots, x_n \in \mathcal{I}_m^{enc}$ be choices of SOD_m assignments and encoding pointers. Then we define

$$\Lambda(g^\phi, x_1, \dots, x_n) = \frac{(n/2)!}{n!} \prod_{p \in \text{im}(\phi)} \mu_{\phi^{-1}(p)}^-(x_p) \prod_{p \in [n] \setminus \text{im}(\phi)} \mu_*(x_p)$$

and for all remaining inputs $(g, x_1, \dots, x_n)$ (i.e. those for which ϕ_g is not injective), we set $\Lambda(g, x_1, \dots, x_n) = 0$.

For any conjunction C over the variables of $\text{LOSSY}(\text{SOD})_{n,m}$, recall

$$\mathbf{E}_\Lambda[C] = \sum_{x \in [n]^{n/2} \times (\mathcal{I}_m^{enc})^n} C(x) \Lambda(x)$$

The main idea for why this would yield a pseudoexpectation is as follows. $\mathbf{E}_\Lambda$ is zero on witnessing conjunctions for the same reason that $\mathbf{E}_\mu$ was: any violation of $\text{LOSSY}(\text{SOD})$ must involve some unmatched plaintext p . Therefore any witnessing conjunction C must also witness a violation in the SOD instance attached to some unmatched p . Since we still “sample” x_p according to μ_* when p is unmatched, we can appeal to [Corollary 4.4](#) to see that any low-degree conjunction witnessing a violation of $\text{LOSSY}(\text{SOD})$ must map to 0 under $\mathbf{E}_\Lambda$. Meanwhile, observe that the argument just outlined is completely agnostic of how we sample x_p for matched plaintexts p , so long as the encoding pointers for witnesses to SOD are sampled consistently with ϕ . This means that we are free to sample the remaining variables according to our choice of a true distribution to counteract the negative contribution from μ_* . By sampling according to μ_σ^- in particular, we will be able to appeal to [Lemma 12.9](#) to bound the negative contribution of the unmatched plaintexts to $\mathbf{E}_\Lambda$ in terms of the positive contribution from the matched plaintexts.

12 Proof of Lossy(SoD) Degree Lower Bounds

Our goal for this section is to prove the following, more precise version of [Theorem 10.1](#).

Theorem 12.1. Let $d = \lfloor \min \{ \frac{1}{4}n, \frac{1}{20}m^{1/3} \} \rfloor$. For all n, m sufficiently large, $\mathbf{E}_\Lambda$ is a degree- d indicator pseudoexpectation for $\text{LOSSY}(\text{SOD})_{n,m}$.

Clearly $\mathbf{E}_\Lambda[1] = 1$ and linearity is satisfied by definition. It remains to show that $\mathbf{E}_\Lambda$ also satisfies the remaining properties of an indicator pseudoexpectation, captured by the following lemmas.

Lemma 12.2. Let m be sufficiently large, and let $d \leq \frac{1}{20}m^{1/3}$, $d \in \mathbb{N}$. Then for any conjunction C of block degree at most d which witnesses a clause of $\text{LOSSY}(\text{SOD})_{n,m}$, $\mathbf{E}_\Lambda[C] = 0$.

Lemma 12.3. Let n and m be sufficiently large and suppose $d \in \mathbb{N}$ satisfies $d \leq \min \{ \frac{1}{4}n, \frac{1}{20}m^{1/3} \}$. Then for any conjunction C over the variables of $\text{LOSSY}(\text{SOD})$ of block degree at most d , $\mathbf{E}_\Lambda[C] \geq 0$.

12.1 Setup and Technical Lemmas

Before proving that $\mathbf{E}_\Lambda$ satisfies the properties of an indicator pseudoexpectation, we introduce some more notation. We will often separate a conjunction C into sub-conjunctions querying different subsets of variables, as follows.

Definition 12.4. Let C be a conjunction over the variables of $\text{LOSSY}(\text{SOD})_{n,m}$. Then we may write $C = C_{\text{dec}}C_1 \cdots C_n$, where C_{dec} is a conjunction over the variables $\{g_\sigma\}_{\sigma \in [n/2]}$ and for each plaintext $p \in [n]$, C_p is a conjunction over the variables $\{a_{p,u}\}_{u \in [m] \times [m]} \cup \{s_{p,u}\}_{u \in [m-1] \times [m]} \cup \{f_{p,u}\}_{u \in [m] \times [m]}$.

The following notation, allowing us to condition on a choice of decoding function $\phi : [n/2] \rightarrow [n]$, will also be useful.

Definition 12.5. Let $C = C_{\text{dec}}C_1 \cdots C_n$ be a conjunction over the variables of $\text{LOSSY}(\text{SOD})_{n,m}$. Then for any injective $\phi : [n/2] \rightarrow [n]$, define

$$\mathbf{E}_\Lambda[C \mid \phi] = C_{\text{dec}}(g^\phi) \prod_{p \in \text{im}(\phi)} \mathbf{E}_{\mu_{\phi^{-1}(p)}^-} [C_p] \prod_{p \notin \text{im}(\phi)} \mathbf{E}_{\mu_*} [C_p]$$

As the notation would suggest, we can “marginalize” $\mathbf{E}_\Lambda$ to rewrite it in terms of these conditional expectations.

Lemma 12.6. Let $C = C_{\text{dec}}C_1 \cdots C_n$ be a conjunction over the variables of $\text{LOSSY}(\text{SOD})_{n,m}$. Then we can write

$$\mathbf{E}_\Lambda[C] = \sum_{\substack{\phi: [n/2] \rightarrow [n] \\ \phi \text{ injective}}} C_{\text{dec}}(g^\phi) \cdot \frac{(n/2)!}{n!} \prod_{p \in \text{im}(\phi)} \mathbf{E}_{\mu_{\phi^{-1}(p)}^-} [C_p] \prod_{p \notin \text{im}(\phi)} \mathbf{E}_{\mu_*} [C_p]$$

In particular,

$$\mathbf{E}_\Lambda[C] = \mathbb{E}_\phi \left[\mathbf{E}_\Lambda[C \mid \phi] \right]$$

where ϕ is sampled uniformly at random from the set of all injective mappings from $[n/2]$ to $[n]$.

Proof. We can expand $\mathbf{E}_\Lambda[C]$ as

$$\begin{aligned} \mathbf{E}_\Lambda[C] &= \sum_{\substack{\phi: [n/2] \rightarrow [n] \\ \phi \text{ injective}}} \sum_{x_1, \dots, x_n \in \mathcal{I}_m^{enc}} C_{\text{dec}}(g^\phi) \prod_{p \in [n]} C_p(x_p) \cdot \frac{(n/2)!}{n!} \prod_{p \in \text{im}(\phi)} \mu_{\phi^{-1}(p)}^-(x_p) \prod_{p \notin \text{im}(\phi)} \mu_*(x_p) \\ &= \sum_{\phi} C_{\text{dec}}(g^\phi) \cdot \frac{(n/2)!}{n!} \prod_{p \in \text{im}(\phi)} \left[\sum_{x_p \in \mathcal{I}_m^{enc}} C_p(x_p) \mu_{\phi^{-1}(p)}^-(x_p) \right] \prod_{p \notin \text{im}(\phi)} \left[\sum_{x_p \in \mathcal{I}_m^{enc}} C_p(x_p) \mu_*(x_p) \right] \\ &= \sum_{\phi} C_{\text{dec}}(g^\phi) \cdot \frac{(n/2)!}{n!} \prod_{p \in \text{im}(\phi)} \mathbf{E}_{\mu_{\phi^{-1}(p)}^-} [C_p] \prod_{p \notin \text{im}(\phi)} \mathbf{E}_{\mu_*} [C_p] = \mathbb{E}_\phi \left[\mathbf{E}_\Lambda[C \mid \phi] \right] \quad \square \end{aligned}$$

Finally, as in [Section 6](#), we define several notions of consistency so that we can restrict our attention to conjunctions which are “supported” by Λ or compatible with a particular choice of decoding function ϕ .

Definition 12.7. A conjunction C is Λ -inconsistent if it contains any of the following conjunctions as a factor:

1. $\llbracket a_{p,(1,1)} = 0 \rrbracket \llbracket f_{p,(1,1)} = \sigma \rrbracket \llbracket g_\sigma = p' \rrbracket$ for some $p \neq p' \in [n]$ and $\sigma \in [n/2]$

2. $\llbracket a_{p,u} = 1 \rrbracket \llbracket s_{p,u} = j \rrbracket \llbracket a_{p,(r(u)+1,j)} = 0 \rrbracket \llbracket f_{p,u} = \sigma \rrbracket \llbracket g_\sigma = p' \rrbracket$ for $u \in [m-1] \times [m]$, $p \neq p' \in [n]$, $j \in [m]$, and $\sigma \in [n/2]$

For an injective mapping $\phi : [n/2] \rightarrow [n]$, C is ϕ -inconsistent if it contains any of the following factors:

1. $\llbracket g_\sigma = p \rrbracket$ for some $\sigma \in [n/2]$ and $p \neq \phi(\sigma)$
2. $\llbracket a_{p,u} = 1 \rrbracket \llbracket f_{p,u} = \sigma \rrbracket$ for $u \in [m] \times [m]$ with $r(u) = m$, $p \in \text{im}(\phi)$, and $\sigma \neq \phi^{-1}(p)$.

Lemma 12.8. *If C is Λ -inconsistent, then $\mathbf{E}_\Lambda[C] = 0$. Similarly, if C is ϕ -inconsistent, then $\mathbf{E}_\Lambda[C \mid \phi] = 0$.*

Proof. For C which is Λ -inconsistent, this follows immediately from the definitions of Λ , $\mathcal{D}_{k,*}$, and $\mathcal{D}_{k,\sigma}$. Otherwise, suppose C is ϕ -inconsistent for some injective $\phi : [n/2] \rightarrow [n]$. If it contains a factor of the first form, we are done as $C_{dec}(g^\phi) = 0$. Otherwise, it contains a factor of the second form, so let $\llbracket a_{p,u} = 1 \rrbracket \llbracket f_{p,u} = \sigma \rrbracket$ be this factor. Since $p \in \text{im}(\phi)$, $\mathbf{E}_\Lambda[C \mid \phi]$ contains the factor $\mathbf{E}_{\mu_{\phi^{-1}(p)}^-}[C_p]$. But clearly for any k and any $\sigma' \neq \sigma$, $\mathbb{E}_{\mathcal{D}_{k,\sigma'}}[\llbracket a_{p,u} = 1 \rrbracket \llbracket f_{p,u} = \sigma \rrbracket] = 0$, so $\mathbf{E}_{\mu_{\phi^{-1}(p)}^-}[C_p] = 0$. $\square$

12.2 The Lossy(SoD) Pseudoexpectation is Non-Witnessing

With the basic lemmas in hand, our first goal is to show that $\mathbf{E}_\Lambda$ is non-witnessing. Intuitively, this should follow straightforwardly from [Corollary 4.4](#), but formally verifying this requires us to unfold the definition of Λ .

Lemma 12.2. *Let m be sufficiently large, and let $d \leq \frac{1}{20}m^{1/3}$, $d \in \mathbb{N}$. Then for any conjunction C of block degree at most d which witnesses a clause of $\text{LOSSY}(\text{SOD})_{n,m}$, $\mathbf{E}_\Lambda[C] = 0$.*

Proof. By [Lemma 12.6](#) we can write

$$\mathbf{E}_\Lambda[C] = \mathbb{E}_\phi \left[\mathbf{E}_\Lambda[C \mid \phi] \right]$$

Therefore it suffices to show that for any injective mapping ϕ , $\mathbf{E}_\Lambda[C \mid \phi] = 0$, so fix any such ϕ to condition on. If C was ϕ -inconsistent, then we are done by [Lemma 12.8](#). Otherwise, since C is witnessing, it must contain a factor of the form $\llbracket a_{p,u} = 1 \rrbracket \llbracket f_{p,u} = \sigma \rrbracket \llbracket g_\sigma = q \rrbracket$ for some pair of plaintexts $p \neq q$, some node u with $r(u) = m$, and some code σ (otherwise it is Λ -inconsistent and $\mathbf{E}_\Lambda[C] = 0$ by [Lemma 12.8](#)). Therefore, since C was ϕ -consistent, we must have $\phi(\sigma) = q$ and $p \notin \text{im}(\phi)$, so $\mathbf{E}_\Lambda[C \mid \phi]$ contains a factor $\mathbf{E}_{\mu_*}[C_p]$. We show how to apply [Corollary 4.4](#) to see that this factor is 0. Begin by expanding

$$\mathbf{E}_{\mu_*}[C_p] = \sum_{i=0}^N \gamma_i \mathbb{E}_{\mathcal{D}_{t_i,*}}[C_p].$$

Now we can decompose C_p as $C_{p,enc}C_{p,SoD}$, where $C_{p,enc}$ is a conjunction over encoding variables $\{f_{p,u}\}_{u \in [m] \times [m]}$ and $C_{p,SoD}$ is a conjunction over the variables of SOD. Since each encoding variable is independent of all other variables in $\mathcal{D}_{t_i,*}$, we have

$$\mathbb{E}_{\mathcal{D}_{t_i,*}}[C_p] = \mathbb{E}_f[C_{p,enc}] \mathbb{E}_{\mathcal{D}_{t_i}}[C_{p,SoD}]$$

where we sample $f \in [n/2]^{m^2}$ uniformly at random, and

$$\mathbf{E}_{\mu_*}[C_p] = \mathbb{E}_f[C_{p,enc}] \sum_{i=0}^N \gamma_i \mathbb{E}_{\mathcal{D}_{t_i}}[C_{p,SoD}] = \mathbb{E}_f[C_{p,enc}] \mathbf{E}_\mu[C_{p,SoD}]$$

where we recall that μ is as in [Part I](#), modified to use the new interpolation nodes. But now notice that $C_{p,SoD}$ inherits the factor $\llbracket a_{p,u} = 1 \rrbracket$, so it is witnessing for SOD. Therefore, since $C_{p,SoD}$ clearly has block degree at most $d < m/100$ for sufficiently large m , [Corollary 4.4](#) applies and $\mathbf{E}_{\mu_*}[C_p] = \mathbb{E}_f[C_{p,enc}] \cdot 0 = 0$. $\square$

12.3 The Lossy(SoD) Pseudoexpectation is Non-Negative

The final property to prove is non-negativity.

Lemma 12.3. *Let n and m be sufficiently large and suppose $d \in \mathbb{N}$ satisfies $d \leq \min \{ \frac{1}{4}n, \frac{1}{20}m^{1/3} \}$. Then for any conjunction C over the variables of LOSSY(SOD) of block degree at most d , $\mathbf{E}_\Lambda[C] \geq 0$.*

As discussed previously, the argument for non-negativity will crucially use the following fact, generalizing the observation that $\mu_\sigma(x) \geq -\alpha\mu_\sigma^-(x)$ for all x .

Lemma 12.9. *Again let $d \leq \frac{1}{20}m^{1/3}$ and suppose C is a conjunction of block degree at most d with variables taking values in $\mathcal{I}_m^{\text{enc}}$. Then for m sufficiently large,*

$$\mathbf{E}_{\mu_*}[C] \geq -\alpha \mathbf{E}_{\mu_\sigma^-}[C]$$

for all $\sigma \in [n/2]$.

Proof. Recall that by the choice of normalization factor α in the definition of μ_σ^- , $\mu_\sigma(x) \geq -\alpha\mu_\sigma^-(x)$. We will show that furthermore, for any conjunction C of block degree at most d , $\mathbf{E}_{\mu_*}[C] = \mathbf{E}_{\mu_\sigma}[C]$. By combining these two facts, the lemma then follows, so we turn our attention to proving this equality.

Write the conjunction as $C = C_0 C_{\text{enc}}$, where we can additionally write

$$C_{\text{enc}} = \prod_{u \in V} \llbracket f_u = f^C(u) \rrbracket,$$

where $V \subseteq [m] \times [m]$ is the set of nodes whose encoding pointers are queried by C , $f^C : V \rightarrow [n/2]$ is the function mapping each queried node to the value asserted by C , and C_0 is the remaining sub-conjunction over the variables of SOD_m . If no such function f^C exists, because C asserts that some variable takes multiple values, then the Lemma trivially holds since $C \equiv 0$ on the supports of both $D_{k,*}$ and $D_{k,\sigma}$ for all k .

Let $V_m \subseteq V$ be the set of nodes $u \in V$ with $r(u) = m$ (recall that $r(u)$ denotes the row index of u). Note that we may assume that this set is non-empty, as otherwise $\mathbb{E}_{D_{k,\sigma}}[C] = \mathbb{E}_{D_{k,*}}[C]$ for all k and we are done.

First, observe that we can write

$$\mathbb{E}_{\mathcal{D}_{t_i,\sigma}}[C] = \mathbb{E}_{(a,\pi) \sim \mathcal{D}_{t_i}} \left[C_0(a, \pi) \mathbb{E}[C_{\text{enc}} \mid a, \pi] \right],$$

since C_0 only depends on the variables of SOD_m . For any assignment $\beta \in \{0, 1\}^{V_m}$, let

$$A_\beta = \prod_{u \in V_m} \llbracket a_u = \beta_u \rrbracket.$$

Given any assignment to the SOD variables with activities β on the last row, the probability that the encoding pointers f_u satisfy C_{enc} under $\mathcal{D}_{t_i,\sigma}$ depends only on β , independently of t_i . Letting $h_\sigma(\beta)$ denote this probability, we substitute $\mathbb{E}[C_{\text{enc}}(f) \mid \beta, \pi]$ with $h_\sigma(\beta)$, and rewrite:

$$\mathbb{E}_{\mathcal{D}_{t_i,\sigma}}[C] = \sum_{\beta} h_\sigma(\beta) \mathbb{E}_{\mathcal{D}_{t_i}}[C_0 A_\beta].$$

By multiplying through by γ_i and summing up, we have

$$\begin{aligned}\mathbf{E}_{\mu_\sigma}[C] &= \sum_{i=0}^N \gamma_i \mathbf{E}_{\mathcal{D}_{t_i, \sigma}}[C] \\ &= \sum_{i=0}^N \gamma_i \left(\sum_{\beta} h_{\sigma}(\beta) \mathbf{E}_{\mathcal{D}_{t_i}}[C_0 A_{\beta}] \right) = \sum_{\beta} h_{\sigma}(\beta) \mathbf{E}_{\mu}[C_0 A_{\beta}]\end{aligned}$$

When $\beta \neq 0^{|V_m|}$, the conjunction $C_0 A_{\beta}$ is either identically 0, or it witnesses a falsified activity constraint on the final row of the SOD_m instance. Moreover, we have

$$\text{bdeg}(C_0 A_{\beta}) \leq \text{bdeg}(C_0) + |V_m| \leq \text{bdeg}(C) \leq d,$$

and so if $C_0 A_{\beta}$ witnesses a falsified activity constraint, then by [Corollary 4.4](#) we have $\mathbf{E}_{\mu}[C_0 A_{\beta}] = 0$. Hence all terms vanish except when $\beta = 0^{|V_m|}$. Since $\sum_{\beta} A_{\beta} = 1$, this implies that

$$\mathbf{E}_{\mu}[C_0] = \sum_{\beta} \mathbf{E}_{\mu}[C_0 A_{\beta}] = \mathbf{E}_{\mu}[C_0 A_{0^{|V_m|}}].$$

Moreover, for $\beta = 0^{|V_m|}$, $h_{\sigma}(\beta) = (2/n)^{|V|}$, and so combining everything together we have

$$\mathbf{E}_{\mu_\sigma}[C] = \left(\frac{2}{n}\right)^{|V|} \mathbf{E}_{\mu}[C_0 A_{0^{|V_m|}}] = \left(\frac{2}{n}\right)^{|V|} \mathbf{E}_{\mu}[C_0] = \mathbf{E}_{\mu_*}[C],$$

completing the proof. □

The rough idea of how we apply this fact is as follows. Using [Lemma 12.6](#), we may marginalize $\mathbf{E}_{\Lambda}[C]$ to isolate the contribution K_{ϕ} of each mapping $\phi : [n/2] \rightarrow [n]$. The strategy is to then argue that for each ϕ with $K_{\phi} < 0$, there are many similar mappings ϕ' for which $K_{\phi'}$ is positive and such that $K_{\phi}/K_{\phi'} = \mathbf{E}_{\mu_*}[C] / \mathbf{E}_{\mu_{\sigma}^-}[C]$ for some code $\sigma \in [n/2]$. We can then bound this ratio using [Lemma 12.9](#) to argue that the contributions from these similar mappings are sufficient to counteract that of ϕ .

Proof of Lemma 12.3. Again begin by decomposing C as $C_{dec} C_1 \cdots C_n$. Let $P_C \subseteq [n]$ be the set of plaintexts p for which C queries some variable from x_p (i.e. C_p is non-empty), and let $\Sigma_C \subseteq [n/2]$ be the set of codes σ for which C queries g_{σ} . Define $\phi_C : \Sigma_C \rightarrow [n]$ to be the mapping of each code $\sigma \in \Sigma_C$ to the plaintext which C asserts it decodes to.

Applying [Lemma 12.6](#), we may write

$$\begin{aligned}\mathbf{E}_{\Lambda}[C] &= \mathbb{E}_{\phi} \left[\mathbf{E}_{\Lambda}[C \mid \phi] \right] = \frac{(n/2)!}{n!} \sum_{\substack{\phi: [n/2] \rightarrow [n] \\ \phi \text{ injective}}} C_{dec}(g^{\phi}) \mathbf{E}_{\Lambda}[C \mid \phi] \\ &= \frac{(n/2)!}{n!} \sum_{\phi \supseteq \phi_C} \mathbf{E}_{\Lambda}[C \mid \phi]\end{aligned}$$

where the last step is because $C_{dec}(g^{\phi}) = 0$ for any mapping ϕ which disagrees with ϕ_C on any coordinate in Σ_C , and we are summing over injective $\phi \supseteq \phi_C$.

Our goal at this point is to bound the total contribution of negative terms to this sum, and show that this is outweighed by the contribution from positive terms. Therefore let Φ_+ be the set of injective mappings

$\phi \supseteq \phi_C$ for which $\mathbf{E}_\Lambda[C \mid \phi] \geq 0$, and Φ_- the set of those for which $\mathbf{E}_\Lambda[C \mid \phi] < 0$. To compare the total contributions of each, we will consider pairs of $\phi_+ \in \Phi_+$ and $\phi_- \in \Phi_-$ which are *C-adjacent*, meaning that there is some choice of plaintexts $p_- \notin P_C \cup \text{im}(\phi_C)$ and $p_+ \in P_C \setminus \text{im}(\phi_C)$, as well as some code $\sigma \notin \Sigma_C$, for which

- (a) $\phi_-(\sigma) = p_-$ and $\phi_+(\sigma) = p_+$,
- (b) $\phi_+(\sigma') = \phi_-(\sigma')$ for all $\sigma' \neq \sigma$, and
- (c) $\mathbf{E}_{\mu_*}[C_{p_+}] < 0$.

In other words, ϕ_- and ϕ_+ are adjacent if we may obtain ϕ_+ from ϕ_- by choosing some matched plaintext p_- which was not queried by C , then exchanging it with some unmatched plaintext p_+ which was queried by C and contributes a negative factor to $\mathbf{E}_\Lambda[C \mid \phi_-]$ (such a choice must exist, since $\mathbf{E}_\Lambda[C \mid \phi_-] < 0$). Recall that

$$\mathbf{E}_\Lambda[C \mid \phi] = C_{dec}(g^\phi) \prod_{p \in \text{im}(\phi)} \mathbf{E}_{\mu_{\phi^{-1}(p)}}^-[C_p] \prod_{p \notin \text{im}(\phi)} \mathbf{E}_{\mu_*}^-[C_p]$$

so since ϕ_- and ϕ_+ only differ on σ , and between p_- and p_+ only p_+ was queried by C , $\mathbf{E}_\Lambda[C \mid \phi_-]$ only differs from $\mathbf{E}_\Lambda[C \mid \phi_+]$ by a factor of $\mathbf{E}_{\mu_*}[C_{p_+}] / \mathbf{E}_{\mu_\sigma}^-[C_{p_+}]$.

Therefore by [Lemma 12.9](#), for every pair ϕ_+, ϕ_- of adjacent mappings we have

$$\mathbf{E}_\Lambda[C \mid \phi_-] \geq \frac{\mathbf{E}_{\mu_*}[C_{p_+}]}{\mathbf{E}_{\mu_\sigma}^-[C_{p_+}]} \cdot \mathbf{E}_\Lambda[C \mid \phi_+] \geq -\alpha \mathbf{E}_\Lambda[C \mid \phi_+] \quad (10)$$

Notice also that for each $\phi_- \in \Phi_-$ there are at least $n/2 - d$ choices of $\phi_+ \in \Phi_+$ adjacent to ϕ_- (since $|P_C \cup \text{im} \phi_C| \leq d$), and for each $\phi_+ \in \Phi_+$ there are at most $dn/2$ choices of adjacent $\phi_- \in \Phi_-$ (first choose some queried plaintext $p_+ \in \text{im}(\phi_+)$, and then some $p_- \notin \text{im}(\phi_+)$ to exchange it with). Therefore by summing [Equation \(10\)](#) over all pairs of adjacent mappings we obtain

$$(n/2 - d) \sum_{\phi_- \in \Phi_-} \mathbf{E}_\Lambda[C \mid \phi_-] \geq -\frac{\alpha dn}{2} \sum_{\phi_+ \in \Phi_+} \mathbf{E}_\Lambda[C \mid \phi_+].$$

Thus, so long as $(n/2 - d) \geq \alpha dn/2$ we have $\sum_{\phi_- \in \Phi_-} |\mathbf{E}_\Lambda[C \mid \phi_-]| \leq \sum_{\phi_+ \in \Phi_+} \mathbf{E}_\Lambda[C \mid \phi_+]$, and $\mathbf{E}_\Lambda[C] \geq 0$. But $(n/2 - d) \geq \alpha dn/2$ follows since $d \leq n/4$ by assumption and $\alpha \leq 1/4d$ by [Proposition 11.2](#). □

[Theorem 12.1](#) now follows.

References

- [AAdRK25] Noel Arteché, Albert Atserias, Susanna de Rezende, and Erfan Khaniki. The Proof Analysis Problem. In *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2555–2576, December 2025.
- [AB04] Albert Atserias and María Luisa Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2):182–201, March 2004.
- [ABE02] Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. Lower Bounds for the Weak Pigeonhole Principle and Random Formulas beyond Resolution. *Information and Computation*, 176(2):136–152, August 2002.

- [AH20] Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for sums-of-squares and positivstellensatz proofs. In *Proceedings of the 34th Computational Complexity Conference, CCC '19*, pages 1–20, Dagstuhl, DEU, September 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [Ajt94] M. Ajtai. The complexity of the Pigeonhole Principle. *Combinatorica*, 14(4):417–433, December 1994.
- [AM20] Albert Atserias and Moritz Müller. Automating Resolution is NP-Hard. *Journal of the ACM (JACM)*, 67(5):31:1–31:17, September 2020.
- [AMO15] Albert Atserias, Moritz Müller, and Sergi Oliva. Lower Bounds for Dnf-Refutations of a Relativized Weak Pigeonhole Principle. *The Journal of Symbolic Logic*, 80(2):450–476, 2015.
- [AR08] Michael Alekhnovich and Alexander A. Razborov. Resolution Is Not Automatizable Unless $W[P]$ Is Tractable. *SIAM Journal on Computing*, 38(4):1347–1363, January 2008.
- [BB25] Ilario Bonacina and Maria Luisa Bonet. Strength and limitations of Sherali-Adams and Nullstellensatz proof systems. *Annals of Pure and Applied Logic*, 176(4):103538, April 2025.
- [BDG⁺04] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-Automatizability of Bounded-Depth Frege Proofs. *computational complexity*, 13(1-2):47–68, December 2004.
- [BHK⁺19] Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K. Kothari, Ankur Moitra, and Aaron Potechin. A Nearly Tight Sum-of-Squares Lower Bound for the Planted Clique Problem. *SIAM Journal on Computing*, 48(2):687–735, January 2019.
- [BIK⁺96] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower Bounds on Hilbert’s Nullstellensatz and Propositional Proofs. *Proceedings of the London Mathematical Society*, s3-73(1):1–26, July 1996.
- [BP96] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 274–282, October 1996.
- [BPR00] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On Interpolation and Automatization for Frege Systems. *SIAM Journal on Computing*, 29(6):1939–1967, January 2000.
- [BS97] Roberto J. Bayardo and Robert C. Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI’97/IAAI’97*, pages 203–208, Providence, Rhode Island, July 1997. AAAI Press.
- [BT22] Mark Bun and Justin Thaler. Approximate Degree in Classical and Quantum Computing. *Foundations and Trends in Theoretical Computer Science*, 15(3-4):229–423, December 2022.
- [Bus97] Samuel Buss. Lower Bounds on Nullstellensatz Proofs by Designs. 39, November 1997.
- [CdR25] Gaia Carenini and Susanna de Rezende. On the Automatability of Tree-Like k-DNF Resolution. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:21, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

- [CLRS13] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate Constraint Satisfaction Requires Large LP Relaxations. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 350–359, Berkeley, CA, USA, October 2013. IEEE.
- [Dan07] Stefan S. Dantchev. Rank complexity gap for Lovász-Schrijver and Sherali-Adams proof systems. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 311–317, New York, NY, USA, June 2007. Association for Computing Machinery.
- [dIVK07] Wenceslas Fernandez de la Vega and Claire Kenyon-Mathieu. Linear programming relaxations of maxcut. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 53–61, USA, January 2007. Society for Industrial and Applied Mathematics.
- [DMR09] Stefan Dantchev, Barnaby Martin, and Mark Rhodes. Tight rank lower bounds for the Sherali-Adams proof system. *Theoretical Computer Science*, 410(21):2054–2063, May 2009.
- [dR21] Susanna de Rezende. Automating Tree-Like Resolution in Time $n(\log)$ Is ETH-Hard. *Procedia Computer Science*, 195:152–162, 2021.
- [DR23] Ben Davis and Robert Robere. Colourful TFNP and Propositional Proofs. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference (CCC 2023)*, volume 264 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [dRGN⁺21] Susanna de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 209–222, New York, NY, USA, June 2021. Association for Computing Machinery.
- [dRLNS21] Susanna de Rezende, Massimo Lauria, Jakob Nordström, and Dmitry Sokolov. The Power of Negative Reasoning. *LIPIcs, Volume 200, CCC 2021*, 200:40:1–40:24, 2021.
- [dRPR23] Susanna de Rezende, Aaron Potechin, and Kilian Risse. Clique Is Hard on Average for Unary Sherali-Adams. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 12–25, November 2023.
- [FKP19] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic Proofs and Efficient Algorithm Design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, December 2019.
- [Gar24] Michal Garlík. Failure of Feasible Disjunction Property for k-DNF Resolution and NP-Hardness of Automating It. In Rahul Santhanam, editor, *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [GGKS18] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 902–911, Los Angeles CA USA, June 2018. ACM.

- [GHJ⁺24] Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in Proof Complexity and TFNP. *Journal of the ACM*, 71(4):26:1–26:45, August 2024.
- [GHP01] Dima Grigoriev, Edward Hirsch, and Dmitrii V. Pasechnik. Complexity of semi-algebraic proofs, December 2001.
- [GKMP20] Mika Göös, Sajin Korothe, Ian Mertz, and Toniann Pitassi. Automating cutting planes is NP-hard. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 68–77, New York, NY, USA, June 2020. Association for Computing Machinery.
- [GL10a] Nicola Galesi and Massimo Lauria. On the Automatizability of Polynomial Calculus. *Theory of Computing Systems*, 47(2):491–506, August 2010.
- [GL10b] Nicola Galesi and Massimo Lauria. Optimality of size-degree tradeoffs for polynomial calculus. *ACM Transactions on Computational Logic (TOCL)*, 12(1):4:1–4:22, November 2010.
- [GM08] Konstantinos Georgiou and Avner Magen. Limitations of the Sherali-Adams Lift and Project System: Compromising Local and Global Arguments. Technical Report, University of Toronto, 2008.
- [GMT09] Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. Optimal Sherali-Adams Gaps from Pairwise Independence. In Irit Dinur, Klaus Jansen, Joseph Naor, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 125–139, Berlin, Heidelberg, 2009. Springer.
- [Gri01] Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1):613–622, May 2001.
- [Hak22] Tuomas Hakoniemi. *Size Bounds for Algebraic and Semialgebraic Proof Systems*. PhD thesis, Universitat Politècnica de Catalunya, March 2022.
- [Hås25] Johan Håstad. On Small-depth Frege Proofs for PHP. *TheoretiCS*, Volume 4, December 2025.
- [HKP⁺17] Samuel B. Hopkins, Pravesh K. Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The Power of Sum-of-Squares for Detecting Hidden Structures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 720–731, October 2017.
- [KMOW17] Pravesh K. Kothari, Ryuhei Mori, Ryan O’Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 132–145, New York, NY, USA, June 2017. Association for Computing Machinery.
- [KP98] Jan Krajíček and Pavel Pudlák. Some Consequences of Cryptographical Conjectures for S12 and EF. *Information and Computation*, 140(1):82–94, January 1998.
- [KPW95] Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures & Algorithms*, 7(1):15–39, August 1995.

- [Kra01] Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-2):123–140, 2001.
- [KST07] Jan Krajíček, Alan Skelley, and Neil Thapen. NP search problems in low fragments of bounded arithmetic. *The Journal of Symbolic Logic*, 72(2):649–672, June 2007.
- [LLR26] Jiawei Li, Yuhao Li, and Hanlin Ren. Finding Bugs in Short Proofs: The Metamathematics of Resolution Lower Bounds. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, STOC '26, pages 788–798, New York, NY, USA, June 2026. Association for Computing Machinery.
- [MMZ⁺01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Conference on Design Automation - DAC '01*, pages 530–535, Las Vegas, Nevada, United States, 2001. ACM Press.
- [MPW02] Alexis Maciel, Toniann Pitassi, and Alan R. Woods. A New Proof of the Weak Pigeonhole Principle. *Journal of Computer and System Sciences*, 64(4):843–872, June 2002.
- [MPW19] Ian Mertz, Toniann Pitassi, and Yuanhao Wei. Short Proofs Are Hard to Find. *LIPIcs, Volume 132, ICALP 2019*, 132:84:1–84:16, 2019.
- [MS99] J.P. Marques-Silva and K.A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, May 1999.
- [NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, December 1994.
- [Pap24] Theodoros Papamakarios. Depth-d Frege Systems Are Not Automatable Unless $P = NP$. In Rahul Santhanam, editor, *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:17, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3(2):97–140, June 1993.
- [Pot19] Aaron Potechin. Sum of Squares Lower Bounds from Symmetry and a Good Story. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [Pud03] Pavel Pudlák. On reducibility and symmetry of disjoint NP pairs. *Theoretical Computer Science*, 295(1):323–339, February 2003.
- [SA90] Hanif D. Sherali and Warren P. Adams. A Hierarchy of Relaxations between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, August 1990.
- [SBI04] Nathan Segerlind, Sam Buss, and Russell Impagliazzo. A Switching Lemma for Small Restrictions and Lower Bounds for k -DNF Resolution. *SIAM Journal on Computing*, 33(5):1171–1200, January 2004.
- [Sch08] Grant Schoenebeck. Linear Level Lasserre Lower Bounds for Certain k -CSPs. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602, October 2008.

- [Spa08] Robert Spalek. A Dual Polynomial for OR. March 2008.
- [Tha16] Neil Thapen. A Tradeoff Between Length and Width in Resolution. *Theory of Computing*, 12(1):1–14, 2016.