Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 106 (2025)

Exponential Lower Bounds on the Size of ResLin Proofs of Nearly Quadratic Depth

Sreejata Kishor Bhattacharya [*] Arkadev Chattopadhyay [†]

December 6, 2025

Abstract

Itsykson and Sokolov [IS14] identified resolution over parities, denoted by Res($\oplus$), as a natural and simple fragment of $AC^0[2]$-Frege for which no super-polynomial lower bounds on size of proofs are known. Building on a recent line of work ([EGI24], [BCD24], [AI25]), Efremenko and Itsykson [EI25] proved lower bounds of the form $\exp(N^{\Omega(1)})$, on the size of Res($\oplus$) proofs whose depth is upper bounded by $O(N \log N)$, where N is the number of variables of the unsatisfiable CNF formula. The hard formula they used was Tseitin on an appropriately expanding graph, lifted by a 2-stifling gadget. They posed the natural problem of proving super-polynomial lower bounds on the size of proofs that are $\Omega(N^{1+\epsilon})$ deep, for any constant $\epsilon > 0$.

We prove the first such lower bounds. In fact, we show that Res($\oplus$) refutations of Tseitin formulas on constant-degree expanders on m vertices, lifted with Inner-Product gadget of size $O(\log m)$, must have size $\exp(\tilde{\Omega}(N^{\epsilon}))$, as long as the depth of the Res($\oplus$) proofs are $O(N^{2-\epsilon})$, for every $\epsilon > 0$. Here $N = \Theta(m \log m)$ is the number of variables of the lifted formula.

More generally, we prove the following lifting theorem that requires a gadget g to have sufficiently small correlation with all parities: we introduce a notion of hardness against ordinary decision trees that we call (p, q)-DT hardness. We show that if a formula Φ is (p, q)-DT hard, then every Res($\oplus$) refutation of size s of the lifted formula $\Phi \circ g$ must be $\Omega\big(pq/\log(s)\big)$-deep. Our concrete lower bound is obtained by showing that Tseitin formulas on constant degree-expanders are $\big(\Omega(m), \Omega(m)\big)$-hard.

An important ingredient in our work is to show that arbitrary distributions *lifted* with such gadgets fool *safe* affine spaces, an idea which originates in the earlier work of Bhattacharya, Chattopadhyay and Dvorak [BCD24].

[*]TIFR, Mumbai. Email: sreejata.bhattacharya@tifr.res.in. Supported by the Department of Atomic Energy, Govmt. of India, under project #RTI4001 and by a Google PhD Fellowship.

[†]TIFR, Mumbai. Email: arkadev.c@tifr.res.ints. Supported by the Department of Atomic Energy, Govmt. of India, under project #RTI4001 and by a Google India Faculty Award.

ISSN 1433-8092

Contents

1 Introduction

One of the simplest proof systems in propositional proof complexity is Resolution. Haken [Hak85] obtained the first super-polynomial lower bounds on the size of proofs in this system for a CNF encoding of the pigeon-hole-principle forty years ago. Since then it has been very well studied with many beautiful results (see for example [Urq87], [BW01], [Ale+04]). Yet, seemingly slight strengthenings of resolution seem to frustrate current techniques in obtaining non-trivial lower bounds. We will consider one such strengthening, that was introduced by Itsykson and Sokolov [IS14], about ten years ago. This system is called resolution over parities, denoted by Res($\oplus$). It augments resolution by allowing the prover to make $\mathbb{F}_2$-linear inferences, while working with $\mathbb{F}_2$-linear clauses. Proving superpolynomial lower bounds for Res($\oplus$) remains a challenge. It is easy to see that Res($\oplus$) is a subsystem of $AC^0[2]$-Frege. While we know strong lower bounds for AC^0-Frege (see for example [Bea+92]), obtaining super-polynomial lower bounds for $AC^0[2]$-Frege for any unsatisfiable formula in CNF would be a major breakthrough (see for example [MP97]). Thus, Res($\oplus$) is in some sense the weakest natural subfragment of $AC^0[2]$-Frege for which proving strong lower bounds has remained challenging. It is also natural to hope that a successful resolution of this challenge would give rise to new techniques and insight for taking on $\mathrm{AC}^0[2]$-Frege.

Itsykson and Sokolov proved exponential lower bounds on the size of tree-like Res($\oplus$) proofs using customized arguments for some formulas. General techniques for tree-like Res($\oplus$) proofs were developed in the independent works of Beame and Koroth [BK23] and Chattopadhyay, Mande, Sanyal and Sherif [Cha+23] that lifted lower bounds on the height of ordinary tree-like resolution proofs of a formula to that of the size of tree-like Res($\oplus$) proofs of the same formula lifted with an appropriate gadget. A more recent line of work ([EGI24], [BCD24], [AI25], [EI25]) has focused on proving lower bounds against subsystems of Res($\oplus$) that are stronger than tree-like but weaker than general Res($\oplus$). Gryaznov, Pudlak and Talebanfard [GPT22] had proposed several notions of regular proofs for the Res($\oplus$) system as appropriate first target for proving lower bounds. Efremenko, Garlik and Itsykson [EGI24] established lower bounds against such a subsystem of Res($\oplus$) known as bottom-regular Res($\oplus$). Bhattacharya, Chattopadhyay and Dvorak [BCD24] exhibited a CNF which is easy for resolution but hard for bottom-regular Res($\oplus$) - thereby strictly separating unrestricted Res($\oplus$) from bottom-regular Res($\oplus$). Subsequently, Alekseev and Itsykson [AI25] significantly extended the reach of techniques by showing $\exp(N^{\Omega(1)})$ lower bounds against Res($\oplus$) refutations whose depth is restricted to be at most $O(N \log\log N)$, where N is the number of variables of the unsatisfiable CNF. This depth restriction was further improved to $O(N \log N)$ by Efremenko and Itsykson [EI25].

A natural way towards proving lower bounds for unrestricted Res($\oplus$) would be improving the depth restriction all the way to $N^{\omega(1)}$. However, the techniques of Efremenko and Itsykson [EI25] seem to get stuck at $O(N \log N)$. Efremenko and Itsykson [EI25] posed the natural open problem of proving superpolynomial lower bounds against Res($\oplus$) refutations whose depth is restricted to $O(N^{1+\epsilon})$ where $\epsilon > 0$ is some constant .

Our main result, stated below, achieves such a bound.

Theorem 1.1. *Let Φ be the Tseitin contradiction on a (m, d, λ) expander with $\lambda < 0.95$ a small enough constant and m odd. Let $n = md/2$ be the number of edges (which is also the number of variables in Φ). Let IP be the inner product gadget on $b = (4+\eta)\log(n)$ bits where $\eta > 0$ is an arbitrarily small constant. Let $\Psi = \Phi \circ IP$ be the lift of Φ by IP. Let $N = nb$ be the number of variables in Ψ. Then, any Res($\oplus$) refutation of Ψ of depth $\leq O(N^{2-\epsilon})$ requires size* $\exp(\tilde{\Omega}(N^{\epsilon}))$

This pushes the frontier of depth of proofs against which super-polynomial lower bounds on size for Res($\oplus$) can be obtained, from $O(N \log(N))$ to $\tilde{O}(N^2)$. In particular, our lower bound

achieves the best possible lower bound of N^2 on depth of efficient proofs that can be obtained using the random-walk-with-restart framework of Alekseev and Itsykson [AI25]. Another way of interpreting our result is to say that any Res($\oplus$) proof of the hard formula Ψ of size $\exp(N^{o(1)})$ has to be almost N^2 deep, which is significantly super-critical.

1.1 Brief Overview of Our Technique

Our work combines the approaches of Alekseev and Itsykson [AI25], Efremenko and Itsykson [EI25] and Bhattacharya, Chattopadhyay and Dvorak [BCD24] - along with a new equidistribution lemma for *safe* affine spaces.

Currently, the only known technique that can prove super-polynomial lower bounds on the size of DAG-like Res($\oplus$) proofs of even just linear depth is the very recent random-walk-with-restart method of Alekseev and Itsykson [AI25]. To understand the main innovation of our work, we first outline how a barrier of handling proofs deeper than $N \log N$ shows up naturally using this method. In fact, this barrier shows up even when we try implementing the random-walk-with-restart method on ordinary resolution proof DAGs. Let us, therefore, first quickly review this technique using ordinary resolution DAGs.

The main idea of the random-walk-with-restart is as follows: consider a hard CNF formula Φ like the Tseitin contradiction on a d-regular expander graph. For such a Φ it is known that if we take a random assignment to its N input variables, then even after making αN ordinary queries to the input variables, with probability at least $p = 2^{-O(N/d)}$ a decision tree would not be able to locate a falsified clause of Φ. Therefore, a walk of length αN on an ordinary resolution proof DAG starting from the root, upon given such a random assignment would reach a node that is labeled by a sub-cube whose fixed variables do not reveal any falsified clause of Φ, with probability p. We call such a sub-cube *good*. However, if there as few as s nodes in the whole DAG, then by union bound there must exist a good node v which is reached by at least a p/s fraction of all assignments. But then a simple counting argument shows that the co-dimension (i.e. the number of input variables fixed) of the cube associated with v, denoted by A_v, is at most $\log(s/p)$. If d, the degree of the expander G, is $O(\log n)$ and m is the number of the vertices of G, and s is at most 2^m, then the co-dimension of A_v is $O(N/\log N)$. That is although in the walk we may have queried αN edges in total, the DAG is constrained to forget many of them and finally remembers only $O(N/\log N)$ many of them with sufficient probability. The beautiful idea of Alekseev and Itsykson is that we could then make a fresh start of our walk from this node v, i.e. now sample a random assignment from A_v. They were able to show roughly the following: as long as the co-dimension of A_v is smaller than βN for some constant $\beta > 0$ and the cube A_v is good, a random walk of length αN starting from v reaches a good node with probability p . Doing the whole co-dimension counting argument again, we conclude that there exists a good node w, such that co-dim$(A_w) \leq$ co-dim$(A_v) + \log(s/p)$. Hence, one could repeat this argument $\log N$ times as long as $s = 2^{o(N/\log N)}$. As each time we take αN steps, this ensures that there exists a node in the proof DAG at depth $\Omega(N \log N)$. With many more technical ideas from linear algebra like the notion of a *closure* of a linear system on lifted variables, Alekseev and Itsykson were able to lift this idea to Res($\oplus$) DAGs where cubes were replaced by affine spaces and ordinary queries were replaced by parity queries. But there was a loss incurred in the process and they could only prove a depth lower bound of $\Omega(N \log \log N)$. This was improved to match the lower bound of $\Omega(N \log N)$, the best possible given the low success probability p of reaching a good node, using more sophisticated notions like *amortized closure* of a lifted linear system by Efremenko and Itsykson [EI25].

We observe that if we sample a uniformly random input, then even starting from the root of an ordinary resolution DAG, the success probability of reaching a good node on making αN queries is no larger than $2^{-\Omega(N/d)}$. This is because the DAG could make d queries on edges incident to a single vertex in G. The probability that the clause of Φ associated with that

vertex is falsified is then precisely $1/2$, even conditioned on previous queries. Thus, if the DAG processes t vertices of G, expending td queries, the probability of not falsifying any clause is at most 2^{-t}. And unless we're able to boost the success probability p, we cannot handle depth beyond $N \log N$ even for ordinary resolution!

Armed with this observation, we design non-uniform distributions μ_v, one for each good cube A_v of small co-dimension, so that the success probability of reaching a good node w starting from node v is boosted all the way to a constant like $1/3$. The design of this distribution and the analysis of the associated random walk is non-trivial. But even after doing this, there are two challenges. First, the co-dimension counting argument goes for a toss! We were not able to find a simple way around this and fixing this is the most involved and non-trivial contribution of our work. The way we get around this is by going to a lifted space where the lifting is by not a stifling gadget as was being done in almost all previous work on Res($\oplus$), but with a gadget that has small correlation with all parities, like Inner-Product. Roughly speaking, we show that over such lifted spaces, any *lifted distribution* restores the co-dimension counting argument in the sense that if we replace the co-dimension of an affine space by the amortized closure of an affine space, then its growth with re-starts proceeds more or less similar to how co-dimension's growth happened under the uniform distribution. This is driven by our main technical ingredient called the Conditional Fooling Lemma, stated and proved in Section 5. Finally, the second challenge is to lift the analysis of the random walk over ordinary resolution DAG under suitable family of non-uniform distribution to the analysis of a random walk over a Res($\oplus$) DAG under a lifted distribution. Interestingly, overcoming this second challenge is crucially aided by an equidistribution property (Lemma 5.2) that we establish for lifted affine spaces, en route to proving the Conditional Fooling Lemma. This equidistribution property of lifted affine spaces is independently interesting. Even more, it turns out that this lifting can be done in a very general way that doesn't depend on the specific base formula, in our case the Tseitin formula, but simply follows from the propery of the lifted space of the gadget.

A bit more concretely, we say that a formula Φ is (p, q)-DT hard if it satisfies the following: there exists a set of partial assignments (or equivalently cubes) $P \subseteq \{0, 1, *\}^n$ which is downward closed, no $\rho \in P$ falsifies any clause of Φ, and even more, for each $\rho \in P$ that fixes at most p variables there exists a distribution μ_ρ over assignments consistent with ρ that is hard for every ordinary decision tree T of depth q in the following sense: if we sample an input from μ_q, the partial assignment obtained by additionally fixing the variables that T queried, also lies in P with probability at least $1/2$. What we are able to show in Section 8 is that, the lifted formula $\Phi \circ g$ becomes 'analogously' hard for parity decision trees, when g has sufficiently small correlation with parities. We call this analogous hardness notion as (p, q)-PDT hardness. This notion was inspired by the analysis of certain combinatorial games in the work of Alekseev and Itskyson [AI25]. In Section 7, we show that this notion of PDT-hardness yields lower bounds on depth of Res($\oplus$) proofs when its size is small. Chaining together the pieces yields the following lifting theorem, somehwat informally stated.

Theorem 1.2 (Informal version of Theorem 8.9)**.** *Let Φ be $(p, p+q)$-DT hard CNF. Then, any Res($\oplus$) refutation of $\Phi \circ g$ of size s must have depth $\Omega\left(\frac{pq}{\log s}\right)$, provided g has sufficiently small size and small correlation wrt all parities.*

It is worth stressing that the only place that the specifics of the CNF contradiction Φ, which is the Tseitin formula over a constant-degree suitably expanding graph, enters into the argument is at proving that it's $\big((\Omega(m), \Omega(m)\big)$-DT-hard, which we do by a novel argument at the very end in Section 9.

1.2 Some Other Related Work

Our work makes use of the notion of amortized closure that was introduced by Efremenko and Itsykson [EI25]. Apart from improving the depth lower bounds of small size Res($\oplus$) proofs,

[EI25] used this notion to give an alternative proof of a lifting theorem of Chattopadhyay and Dvorák [CD25] and their proof works for a broader class of gadgets. The lifting theorem is used in [CD25] to prove super-criticial tradeoffs between depth and size of tree-like Res($\oplus$) proofs.

Our work also crucially uses lifted distributions to boost the success probability of random walks with restarts. In particular, it uses an analytic property of the gadget to argue equidistribution of pre-images in a *safe* affine space in the lifted world of a point $z \in \{0,1\}^n$ in the unlifted world. Such equidistribution, albeit wrt rectangles, have been earlier implicitly proved (see for example [Göö+16; Cha+21]) as well as explicitly proved in [Cha+17]. The analytic property of the gadget used in these works was essentially small discrepancy wrt rectangles (or being a 2-source extractor), something that seems to be significantly stronger than what we need of the gadget in this work.

Very recent work: Soon after an initial version of our manuscript was uploaded, Byramji and Impagliazzo [BI25a] proved a superlinear lower bound of $\Omega(N^{3/2-\epsilon})$ on the depth of Res($\oplus$) proofs of size $o\big(\exp(N^{2\epsilon})\big)$ for the bit-Pigeon-hole Principle (BPHP). This work continued to use the random-walk-with-restart paradigm of [AI25] under a nearly *uniform distribution*, but there were two differences: the length of each walk was about $\sqrt{N}$ and the probability of the walk ending at a good node was as large as a constant. What prevented them from going beyond $\sqrt{N}$ length walk in each phase was a birthay paradox like phenomenon under their distribution. Very recently, in a later version [BI25b], they seemed to have improved the depth bound for the BPHP to nearly quadratic, employing a more non-uniform distribution. The also prove a lifting theorem starting from our notion of (p,q)-DT-Hardness, using lifted distributions. However, the way they handle bottleneck counting under lifted distributions seems very different from the way we do. In particular, they seem to be able to get away with simpler co-dimension fooling property of a lifted distribution, first observed in [BCD24] in the context of a walk without restart, even when re-starting random walks. This way of bottleneck counting apparently allows them to prove their lifting theorem, similar to our Theorem 1.2, but with just constant-size gadgets. This reduced size allows them to get a nearly quadratic lower bound on depth even in terms of formula size for a lifted Tseitin formula, whereas our lower bound is quadratic in just number of variables of the CNF. It is worth remarking that for this last result, the latest version of the Byramji-Impagliazzo work uses our result, proved in Section 9, that Tseitin formulas are $(\Omega(n),\Omega(n))$-DT-hard.

2 Preliminaries

2.1 General Notation

- For a probability distribution μ, when we sample a point x according to μ, we denote it by $x \leftarrow \mu$.
- When an input is sampled according to some distribution μ conditioned on lying in some set S, we denote the resulting conditional distribution by $\mu \cap S$. Throughout this paper, whenever we encounter such expressions, it will be guaranteed (and easy to see) that $\mathrm{supp}(\mu) \cap S \neq \phi$, so the conditioning is valid.
- When x is sampled according to uniform distribution over a set T, we denote it by $x \sim T$.
- Throughout this paper, we shall identify a linear form $\ell \in (\mathbb{F}_2^{nb})^*$ by its canonical representation wrt the standard inner product, i.e., $\ell(x) = \langle \ell, x\rangle$ where $\langle x, y\rangle = \sum_j x_j y_j \pmod 2$.
- For a partial assignment $\rho \in \{0,1,*\}^n$ we denote

$$\mathrm{free}(\rho) = \{i \in [n] | \rho(i) = *\}$$

$$\text{fix}(\rho) = \{i \in [n] | \rho(i) \neq *\}$$

2.2 Resolution over parities

Definition 2.1. A linear clause ℓ_C is an expression of the form

$$\ell_C(x) = [\langle \ell_1, x \rangle = b_1] \vee [\langle \ell_2, x \rangle = b_2] \cdots \vee [\langle l_k, x \rangle = b_k]$$

Here $x, \ell_1, \cdots, \ell_k \in \mathbb{F}_2^n$. Note that the negation of ℓ_C, $\neg \ell_C$ is an affine space:

$$\neg \ell_C = \{x \in \mathbb{F}_2^n | \langle \ell_1, x \rangle = 1 - b_1, \cdots, \langle \ell_k, x \rangle = 1 - b_k\}$$

Also notice that every ordinary clause is also a linear clause.

Res($\oplus$) (defined in [IS14]) is a proof system where every proof line is a linear clause. The derivation rules are as follows:

1. **Weakening:** From ℓ_C, derive ℓ_D where ℓ_D is any linear clause semantically implied by ℓ_C ($\ell_C \implies \ell_D$).

   This step can be verified efficiently because it is equivalent to $\neg \ell_D \subseteq \neg \ell_C$, and this is a statement about containment of one affine space within another - which can be checked by Gaussian elimination.

2. **Resolution:** From $\ell_C^{(1)}(x) = \ell_C(x) \vee [\langle \ell, x \rangle = b]$ and $\ell_C^{(2)}(x) = \ell_c(x) \vee [\langle \ell, x \rangle = 1 - b]$, derive $\ell_C(x)$

A Res($\oplus$) refutation of a CNF Φ starts with the axioms being the clauses of Φ (which, as noted above, are also linear clauses) and applies a sequence of derivation rules to obtain the empty linear clause $\emptyset$.

Affine DAGs

For an unsatisfiable CNF Φ define the search problem

$$\text{Search}(\Phi) = \{(x, C) | C \text{ is a clause of } \Phi, C(x) = 0\}.$$

Just as a resolution refutation of Φ can be viewed as a cube-DAG for solving Search(Φ), a Res($\oplus$) refutation can be viewed as an affine-DAG for solving Search(Φ).

Definition 2.2. An affine DAG for Search(Φ) is a DAG where there is a distinguished root r, each node v has an associated affine space A_v, and each node has outdegree either 2, 1, or 0. Each outdegree 0 node w is labelled with a clause of Φ, C_w. The following requirements are satisfied:

1. If v has two children v_1, v_2 then $A_v = A_{v_1} \cup A_{v_2}$.
2. If v has only one child w, then $A_v \subseteq A_w$.
3. If v has no children, then for any $x \in A_v$, $C_v(x) = 0$ where C_v is the clause labelled on v.
4. The affine space labelled on the root is the entire space $\mathbb{F}_2^n$.

A Res($\oplus$) refutation for Φ can be viewed as an affine DAG for Search(Φ) by viewing the sequence of derivations as a DAG: for each node, the associated affine space is the negation of the linear clause derived at that node. The leaves are the axioms - the clause labelled at each leaf is simply the corresponding axiom.

We classify nodes based on their outdegree as follows.

1. A node with no children is called a leaf.

2. A node with one child is called a *weakening node*. (Because in the Res($\oplus$) refutation this node was derived by weakening.)

3. Let v be a node with two children v_1, v_2. In this case it holds that $A_v = A_{v_1} \cup A_{v_2}$; $A_{v_1} = A_v \wedge [\langle \ell, x \rangle = b]$ and $A_{v_2} = A_v \wedge [\langle \ell, x \rangle = 1 - b]$ for some $\ell \in \mathbb{F}_2^n$. Such a node is called a *query node*; we say the affine DAG queries ℓ at node v. (In the Res($\oplus$) refutation, node v was obtained by resolving the linear form $\langle \ell, x \rangle$.)

Path of an input

Here we consider any affine DAG that arises from some Res($\oplus$) refutation. For any node v and any $x \in A_v$, we define the path of x starting from v as follows:

- Start with the currrent node v.
- If the current node is w and w has no children, terminate the path.
- If the current node is w and has two children w_1, w_2, we know that $A_w = A_{w_1} \cup A_{w_2}$. In this case it will hold that $A_{w_1} = A_w \cap \{\tilde{x} | \langle \ell, \tilde{x} \rangle = b\}$ and $A_{w_2} = A_w \cap \{\tilde{x} | \langle \ell, \tilde{x} \rangle = 1 - b\}$ for some $\ell \in \mathbb{F}_2$. If $\langle \ell, x \rangle = b$, the next node in the path is w_1. Otherwise, the next node in the path is w_2
- If the current node w has only one child w_1, the next node in the path is w_1.

The way the path is defined ensures that if the path of x visits the node w, $x \in A_w$. Consequently, for any x and v such that $x \in A_v$, the path of x starting from v visits a leaf whose clause is falsified by x.

In particular, for any x, if we follow the path traversed by x from the root, we end up at a clause falsified by x.

Definition 2.3. We define the length of a path to be the number of query nodes encountered on the path. (The weakening nodes do not contribute to the length.)

Definition 2.4. The depth of a node v is the largest length of a path from the root to v. The depth of the refutation is the depth of the deepest node.

2.3 Notations about lifted spaces

In this paper, we shall be working with a gadget $g : \mathbb{F}_2^b \to \mathbb{F}_2$. The base space will be $\mathbb{F}_2^n$. The lifted space will be $\mathbb{F}_2^N$ where $N = nb$. The coordinates of the lifted space are $\{(i, j) | i \in [n], j \in [b]\}$.

Definition 2.5. The set of coordinates $\{x_{i,j} | j \in [b]\}$ is called to be *the block of* i. The i-th block will be denoted as $x(i) \in \mathbb{F}_2^b$

The gadget g naturally induces a function $g^n : \mathbb{F}_2^{nb} \to \mathbb{F}_2^n$ by independent applications of g on the n different blocks. We shall abbreviate g^n by G.

Definition 2.6. For any assignment $\beta \in \mathbb{F}_2^{S \times [b]}$ to the variables in blocks of S, we define the partial assignment $G(\beta) \in \{0, 1, *\}^n$ as follows:

$$G(\beta)_i = \begin{cases} * & \text{if } i \notin S \\ g(\beta(i)) & \text{otherwise} \end{cases}$$

Definition 2.7. For any assignment $z \in \{0,1\}^n$ define $G^{-1}(z)$ to be the set of preimages of z:

- Then, $G^{-1}(z) \subseteq \mathbb{F}_2^{nb}$:

$$G^{-1}(\alpha) = \{y | y \in \mathbb{F}_2^{nb}, g(y(i)) = z_i \ \forall i \in [n]\}$$

Definition 2.8. For any distribution μ on $\mathbb{F}_2^n$ define the lifted distribution $G^{-1}(\mu)$ on $\mathbb{F}_2^{nb}$ as the outcome of the following sampling procedure:

1. Sample $z \leftarrow \mu$.
2. Sample x uniformly at random from $G^{-1}(z)$.

Any distribution of the form $G^{-1}(\mu)$ is called a *lifted distribution*.

Definition 2.9. For a partial assignment $y \in \mathbb{F}_2^{S \times [b]}$ to some set of blocks in the lifted space, we define $G(y) \in \mathbb{F}_2^S$ to be the corresponding partial assignment in the unlifted world: $G(y)_i = g(y(i))) \ \forall i \in S$.

Caution: G will also be used to denote a graph in one part of the paper. It should not cause confusion, however, because in that section we will not be using this interpretation of G.

Definition 2.10. For a partial assignment $y \in \mathbb{F}_2^S$ ($S \subseteq N$), we define the *cube of* y, C_y to be the set of points consistent with y:

$$C_y = \{x \in \mathbb{F}_2^N | x_i = y_i \ \forall i \in S\}$$

Definition 2.11. For an affine space $A \subseteq \mathbb{F}_2^{nb}$ and a partial assignment $y \in \{0,1,*\}^{nb}$, call y A-extendable if there exists $x \in A$ consistent with y

Definition 2.12. For an affine space $A \subseteq \mathbb{F}_2^{nb}$ and an extendable partial assignment $y \in \mathbb{F}_2^S$ (where $S \subseteq [nb]$) define $A_y \subseteq \mathbb{F}_2^{[nb] \setminus S}$ as follows:

$$A_y = \{\tilde{x} | (\tilde{x}, y) \in A\}$$

2.4 Linear algebraic facts about lifted spaces

In this subsection we import facts about closure and amortized closure proved by [EGI24], [AI25] and [EI25].

- **Safe set of linear forms**

  Definition 2.13. (from [EGI24]) A set of linear forms $V = \{\ell_1, \ell_2, \cdots, \ell_m\} \subseteq \mathbb{F}_2^{nb}$ [1]. is *safe* if for any k linearly independent forms $w_1, w_2, \cdots, w_k \in \text{span}(S)$, $\text{supp}(w_1) \cup \text{supp}(w_2) \cup \cdots \cup \text{supp}(w_k)$ includes at least k distinct blocks.

- **Equivalent definition of *safe*:** Let

$$M = \begin{bmatrix} \ell_1 \\ \hline \ell_2 \\ \hline \cdots \\ \hline \ell_m \end{bmatrix} \in \mathbb{F}_2^{m \times nb}$$

  Let $r = \text{rank}(M)$. V is nice iff there exist indices $c_1, c_2, \cdots, c_r \in [nb]$, each lying in different blocks, such that the set $\{Me_{c_1}, \cdots, Me_{c_r}\} \subseteq \mathbb{F}_2^m$ is linearly independent. (Me_j is the j-th column of M) The proof of equivalence of these two definitions can be found in Theorem 3.1 in [EGI24].

[1] Technically, a linear form ℓ should lie in the dual space $(\mathbb{F}_2^N)^*$. In this, we identify a linear form ℓ as an element of $\mathbb{F}_2^N$ by its canonical representation w.r.t the standard inner product: $\ell(x) = \langle \ell, x \rangle$

Fact 2.14. *Whether or not a set of linear forms is safe depends only on their span. This is clear from the second equivalent definition.*

- **Safe affine spaces:**

  Definition 2.15. Let $A \subseteq \mathbb{F}_2^{nb}$ be an affine space. Let $A = \{x | Mx = b\}$. Then, A is called a *safe affine space* if and only if the rows of M are safe (i.e., the set of linear forms defining A is safe). This does not depend on a specific choice of M by Lemma 4.1 in [EGI24].

- **Deviolator:** For a subset of the blocks $S \subseteq [n]$ and a linear form $\ell \in \mathbb{F}_2^{nb}$, define $\ell[\backslash S] \in \mathbb{F}_2^{(n-|S|)b}$ to be the projection of v on the coordinates of $[n] \setminus S$.

  Definition 2.16. A subset $S \subseteq [n]$ is a deviolator for $V = \{\ell_1, \ell_2, \cdots, \ell_m\} \subseteq \mathbb{F}_2^{nb}$ if $\{\ell_1[\backslash S], \ell_2[\backslash S], \cdots, \ell_m[\backslash S]\}$ is a nice set of linear forms.

- **Closure of a set of linear forms:**

  Definition 2.17. (from [EGI24]) Closure of a set of linear forms $V = \{\ell_1, \ell_2, \cdots, \ell_m\}$ is the minimal deviolator for V. (It is known that this deviolator is unique, and also it depends only on span(V) - Lemma 4.1 in [EGI24].)

- **Closure of an affine space:**

  Definition 2.18. For an affine space A given by the set of equations $A = \{x | Mx = b\}$, define $\mathrm{Cl}(A)$ to be the closure of the set of rows of M (i.e., $\mathrm{Cl}(A)$ is the closure of the set of defining linear forms of A). This does not depend on a specific choice of M by Lemma 2.11 in [EI25].

- **Closure Assignment**

  Definition 2.19. For an affine space A, a *closure assignment* y is any assignment to the the coordinates in $\mathrm{Cl}(A) \times [b]$: $y \in \mathbb{F}_2^{\mathrm{Cl}(A) \times [b]}$.

- **Amortized Closure of a set of linear forms**

  Definition 2.20. (from [AI25]) Let $V = \{\ell_1, \ell_2, \cdots, \ell_k\} \in \mathbb{F}_2^{nb}$ be a set of linear forms. We define $\tilde{\mathrm{Cl}}(V) \subseteq [n]$ as follows: Let

$$M = \begin{bmatrix} v_1 \\ \hline v_2 \\ \hline \cdots \\ \hline v_t \end{bmatrix}$$

  Call a set of blocks $S = \{s_1, s_2, \cdots, s_k\} \subseteq [n]$ *acceptable* if there exist columns $c_1, c_2, \cdots, c_k$, such that c_j lies in block s_j and the set $\{Me_{c_1}, Me_{c_2}, \cdots, Me_{c_k}\}$ is linearly independent. The amortized closure of V, $\tilde{\mathrm{Cl}}(V)$, is the lexicographically largest acceptable set of blocks.

  It is known that $\tilde{\mathrm{Cl}}(V)$ depends only on span(V) (Lemma 2.11 in [EI25])

- **Amortized Closure of An Affine Space**

Definition 2.21. Let $A \subseteq \mathbb{F}_2^{nb}$ be an affine space; $A = \{x | Mx = b\}$. The amortized closure of A, $\tilde{\text{Cl}}(A)$, is defined to be the amortized closure of the set of rows of M. This does not depend on a specific choice of M (Lemma 2.11 in [EI25])

Now we import some facts and lemmas about closure and amortized closure from [EGI24], [EI25] and [AI25].

Lemma 2.22. *If y is an A-extendable closure assignment, A_y is a safe affine subspace. (Follows from definition.)*

Lemma 2.23. *For any affine space, $Cl(A) \subseteq \tilde{Cl}(A)$ (Lemma 2.15 in [EI25])*

Lemma 2.24. *If A, B are affine spaces with $B \subseteq A$, then $\tilde{Cl}(A) \subseteq \tilde{Cl}(B)$ (Corollary 2.19 in [EI25]) and $Cl(A) \subseteq Cl(B)$ (Lemma 4.2 in [EGI24])*

Lemma 2.25. *Let $V \subseteq \mathbb{F}_2^N$ be a set of linear forms with $Cl(V) = S$. Let $W = V \cup \{e_{j,k} | j \in S, k \in [b]\}$. Then, $\tilde{Cl}(V) = \tilde{Cl}(W), Cl(V) = Cl(W)$.*

Lemma 2.25 becomes clear once one examines the proof of Lemma 2.15 in [EI25] closely. For completeness we include a self-contained proof in Appendix A.

Lemma 2.26. *Let $V \subseteq W \subseteq \mathbb{F}_2^{nb}$ be sets of linear forms with $|W| = |V| + 1$. Then, $|\tilde{Cl}(W)| \leq |\tilde{Cl}(V)| + 1$, and moreover, if $|\tilde{Cl}(W)| = |\tilde{Cl}(V)| + 1$ then $Cl(W) = Cl(V)$ (Theorem 2.18 and Lemma 2.17 in [EI25]).*

We now state a useful corollary of the above.

Corollary 2.27. *Let $B \subseteq A$ be affine spaces such that $codim(B) = codim(A) + 1$ and $|\tilde{Cl}(B)| = |\tilde{Cl}(A)| + 1$. Let y be any A-extendable closure assignment. Then, A_y, B_y are both nice affine subspaces and $codim(B_y) = codim(A_y) + 1$.*

A proof of Corollary 2.27 is included in Appendix A.

3 A More Detailed Overview

At a high level, our proof combines the approaches of Alekseev and Itsykson [AI25], Efremenko and Itsykson [EI25] and Bhattacharya, Chattopadhyay and Dvorak [BCD24]. It does so by boosting the success probability of the 'random walk with restart' method of [AI25] by sampling inputs from a lifted distribution. The idea of using lifted distribution to do random walks appeared in [BCD24]. The bottleneck counting uses the notion of amortized closure instead of codimension of an affine space as done in [EI25]. However, combining these approaches requires significant new ideas – along with a new equidistribution lemma for gadgets with sufficiently small Fourier coefficients (Lemma 5.2). In this section we give a brief overview of how these approaches fit together.

Approach of Alekseev and Itsykson [AI25] Recall the overall idea of the random-walk-with-restart as outlined for ordinary Resolution proof DAGs in Section 1.1. We start by describing how this approach was implemented by [AI25] for Res($\oplus$) DAGs by developing required linear algebraic machinery, and why it fell short of handling DAGs of depth $O(N \log N)$ that was relative easily possible for resolution DAGs.The main idea in [AI25] is this: they take the CNF Ψ to be Tseitin contradiction over an $(n, \log(n), O(\log(n))$-expander lifted with an appropriate gadget; they assume we are given a size s Res-($\oplus$) refutation Π of Ψ, and they locate a path of length $n \log\log(n)$ in Π. They do this inductively: at Phase j, they locate a vertex v_j at depth $\Omega(nj)$. Given this vertex v_j, they show that as long as $\text{codim}(A_{v_j})$ is not too large, there is another vertex v_{j+1} which is at distance $\Omega(n)$ from j. They show they can inductively

find one more vertex as long as $j \leq O(\log\log(n))$ - and this gives the depth lower bound.

Let us describe it in a bit more detail. Alekseev and Itsykson carefully choose a set of partial assignments in the unlifted world, $P \subseteq \{0,1,*\}^n$ with the idea that any partial assignment $\rho \in P$ leaves *some uncertainty* about which clause of the unlifted Tseitin formula would be falsified if one were to extend ρ at random to a full assignment.

In Phase j, Alekseev and Itsykson [AI25] have located a vertex v_j at depth $\Omega(jn)$. They want the codimension of A_{v_j}, the affine space that the proof Π associates with v_j, to be small ($\leq O(j(\log(s/p))(b+1)^j)$, which is less than jn when j is small enough; p is a parameter we shall specify soon). Small co-dimension implies a small closure, i.e. $\mathrm{codim}(A_{v_j}) \geq |\mathrm{Cl}(A_{v_j})|$. We assume that variables in the unlifted world that correspond to blocks in $\mathrm{Cl}(A_{v_j})$ are revealed, but variables that correspond to blocks outside of the closure, i.e. in $[n] - \mathrm{Cl}(A_{v_j})$ are yet not revealed. Hence, Alekseev and Itsykson fix an extendable closure assignment $y_j \in \mathbb{F}_2^{\mathrm{Cl}(A_{v_j})\times[b]}$ such that $G(y_j)$ lies in P. They show (using a combinatorial argument) that the following happens when we uniformly sample a point $x \in A_{v_j} \cap C_y$ and follow the path of x from v_j for $\Theta(n)$ steps or until it lands at a leaf node, whichever happens earlier: let w be the vertex reached. Let $\tilde{x} \in \mathbb{F}_2^{\mathrm{Cl}(A_w)\times[b]}$ be the restriction of x to the variables of $\mathrm{Cl}(A_w)$. Let $\rho \in \{0,1,*\}^n$ be the partial assignment that leaves all variables outside of $\mathrm{Cl}(A_w)$ free and $\rho|_{\mathrm{Cl}(A_w)} = G(\tilde{x})$. With probability at least p, ρ is in P, i.e. this ρ reveals little about where a potential falsified clause may be. Let us call such a node w to be *good*. For this combinatorial argument to work, it is essential that the starting partial assignment, $G(y_j)$ lies in P and it does not fix too many bits: $|\mathrm{Cl}(A_{v_j})| \leq O(n/\log(n))$.

One such good w will be the next node, v_{j+1} - and the next closure assignment y_{j+1} could be anything in $\mathbb{F}_2^{\mathrm{Cl}(A_w)\times[b]}$ such that $G(y_{j+1}) \in P$ and y_{j+1} is extendible in A_w. The existence of such a y_{j+1} trivially follows as $x|_{\mathrm{Cl}(A_w)\times[b]}$ satisfies those requirements. As w is good, it cannot be a leaf node for no falsified clause can be identified at w. Hence, all good w's are at distance $\Omega(n)$ from v_j - so the only condition Alekseev and Itsykson need to maintain is that the codimension of A_w is not too high. They show the existence of such a w using a simple bottleneck argument: there exists a w such that a uniformly random $x \in A_{v_j} \cap C_y$ reaches A_w with probability $\geq p/s$ as there are at most s many nodes at any given distance from node v_j. In particular, $|A_w| \geq \frac{p}{s}|A_{v_j} \cap C_y|$, which implies $\mathrm{codim}(A_w) \leq (b+1)\mathrm{codim}\big(A_{v_j}\big) + \log(s/p) \leq O((j+1)(\log(s/p))(b+1)^{j+1})$.

Let us now briefly explain why this approach fails to go beyond depth $O(n\log\log n)$. Once $\mathrm{codim}(A_{v_j})$ exceeds $n/\log(n)$, the underlying combinatorial argument in [AI25] to get the next node fails. Hence, the depth lower bound obtained by this argument depends on the number of iterations until which $\mathrm{codim}(A_{v_j})$ is guaranteed to be less than $n/\log(n)$. In this case, there are two factors causing rapid growth of (the guaranteed upper bound on) $\mathrm{codim}(A_{v_j})$: first, at each step, the codimension of the next node can increase geometrically. Second, the success probability p in [AI25] is pretty low: around $2^{-O(n/\log(n))}$ - this also contributes to the growth of the valid upper bound on $\mathrm{codim}(A_{v_j})$. Please recall from Section 1.1, that the estimate of the success probability of the random walk being $2^{-O(n/\log n)}$ is tight, even on an ordinary resolution DAG.

Improvement to depth $\Omega(N\log N)$: In 2025, Efremenko and Itsykson [EI25] bypassed the first barrier (of the codimension growing geometrically at each step) by introducing a new notion, different from co-dimension, to track progress: this notion is the amortized closure $\tilde{\mathrm{Cl}}(A)$. Notice that the reason why the codimension could be growing geometrically in [AI25] was that fixing the bits of $\mathrm{Cl}(A_v)$ to y adds $b|\mathrm{Cl}(A_v)|$ more constraints, which could be as large as $b \times \mathrm{codim}(A_v)$. One of the key lemmas in [EI25] is that if $|\tilde{\mathrm{Cl}}(A_w)| = |\tilde{\mathrm{Cl}}(v)| + k$, then

$\Pr_{x\sim A_v\cap C_y}[x \in A_w] \leq 2^{-k}$. In other words, if $|\tilde{\text{Cl}}(A_w)| = |\tilde{\text{Cl}}(A_v)| + k$, among the equations defining A_w, there exist k linearly independent equations and moreover, these equations are also linearly independent from the equations of $A_v \cap C_y$ as the properties of amortized closure ensure $\tilde{\text{Cl}}(A_v) = \tilde{\text{Cl}}(A_v \cap C_y)$. Now, [EI25] runs the same argument again. This time, it yields the following recursion: $|\tilde{\text{Cl}}(A_{v_{j+1}})| \leq |\tilde{\text{Cl}}(A_{v_j})| + \log(\frac{s}{p})$, which prevents a geometric growth on the size of the amortized closure (as was happening with codimension earlier). This ensures that $|\tilde{\text{Cl}}(A_{v_j})| \leq O(j\log(s/p))$ at Phase j. This enabled Efremenko and Itsykson [EI25] to find a vertex at depth $\Omega(N\log(N))$ assuming s was $\exp(N^{o(1)})$. Thus, the same simple bound on depth achieved on an ordinary resolution DAG by walking under the uniform distribution was achieved for Res($\oplus$) DAGs albiet only after lifting the base Tseitin formula with an appropriately stifling gadget and using the sophisticated notion of an amortized closure of a lifted linear system. However, the second barrier still remained, as we argued earlier that it remained even for ordinary resolution: the success probability p of each phase of the random walk was very small; around $2^{-n/\log(n)}$. Thus, this argument could not go beyond depth $N\log(N)$.

Our approach for depth $N^{2-\epsilon}$: One of the main contributions of this work is getting around this low success probability barrier. As explained in Section 1.1, we do that by going to non-uniform distributions. However, as said before, the bottleneck counting argument goes for a toss (even for sub-cube DAGs of ordinary resolution). To restore the ability of bottleneck counting, a basic requirement seems that our non-uniform distribution should at least fool the simplest linear algebraic notion of co-dimension. Such a fooling was devised in the work of Bhattacharya, Chattopadhyay, and Dvorak [BCD24]. In [BCD24], the authors prove a separation between a restricted class of Res($\oplus$) refutations (known as bottom-regular refutations) and general Res($\oplus$) refutations. Their proof also employed a bottleneck argument, but instead of sampling from the uniform distribution, they were sampling from a lifted distribution. The key observation in [BCD24] was that if $g : \mathbb{F}_2^b \to \{0,1\}$ is an appropriate gadget, then for any lifted distribution $\bar{\mu}$ and any affine space A, $\Pr_{x\leftarrow\bar{\mu}}[x \in A] \leq 2^{-\Omega(\text{codim}(A)/b)}$. This would be sufficient, as it essentially was for [BCD24], to do bottleneck counting at the end of the first phase of the random walk, starting from the root of the DAG.

To do bottleneck counting at subsequent restarts, one needs a conditional version of the above fooling statement in which one samples from the lifted fooling distribution, conditioned on the sampled input lying in some affine space A. Here, A corresponds roughly to the affine space associated with a node of the DAG walk from where we need to re-start. One might hope that conditional version is true: if $B \subseteq A$ are two affine spaces and $\tilde{\mu}$ is a lifted distribution, then $\Pr_{x\leftarrow\tilde{\mu}}[x \in B | x \in A] \leq 2^{-\Omega(\text{codim}(B)-\text{codim}(A))/b}$. If this were true, we could modify the proof of [AI25]: instead of sampling the input uniformly from $A_v \cap C_y$, we could sample from a lifted distribution tailored to our needs - which can hopefully boost the success probability. Unfortunately, such a statement cannot be true for any gadget, as the following counterexample shows.

Counterexample to a naive idea of conditional fooling

Let $t \in \mathbb{F}_2^b$ be a point, such that the first bit (wlog) is g-sensitive at t, i.e. $g(t) \neq g(t \oplus e_{\{1\}})$. WLOG, let $g(t) = 0$. The equations for A are as follows: for all $i \in [n], j \in [t] \setminus \{1\}, x_{ij} = t_j$. In B, we add the following extra equations: for all $i \in [n]$, $x_{i1} = t_1$. Let $\bar{\mu}$ be the uniform distribution on $G^{-1}(0^n)$. Then, even though $\text{codim}(B) = \text{codim}(A) + n$,

$$\Pr_{x\leftarrow\bar{\mu}}[x \in B | x \in A] = 1$$

Intuitively, the reason why conditional fooling does not happen in this counterexample is that A fixes too many linear forms in a block - and thus, when sampling from $G^{-1}(0^n) \cap A$,

the distribution on each block is not controllable. One might imagine if the equations defining A do not concentrate too much on any single block, the distribution $G^{-1}(z) \cap A$ behaves more nicely. One notion of the equations defining A not concentrating on any single block is that A is a safe affine space. Indeed, it turns out that the conditional fooling conjecture is actually true when A and B are both safe affine spaces (Lemma 5.8) and the gadget g is *nice*, i.e. all Fourier coefficients of g are sufficiently small in n, the size of the unlifted space.Given this, it is not hard to show that lifted distributions fool amortized closure. In particular, we shall show that if $\bar{\mu}$ is any g-lifted distribution, g being nice, and $|\tilde{\mathrm{Cl}}(A_w)| = |\tilde{\mathrm{Cl}}(A_v)| + k$, then $\Pr_{x \leftarrow \bar{\mu} \cap C_y}[x \in A_w | x \in A_v] \leq (3/4)^k$ (Conditional Fooling Lemma, i.e. Lemma 5.1).

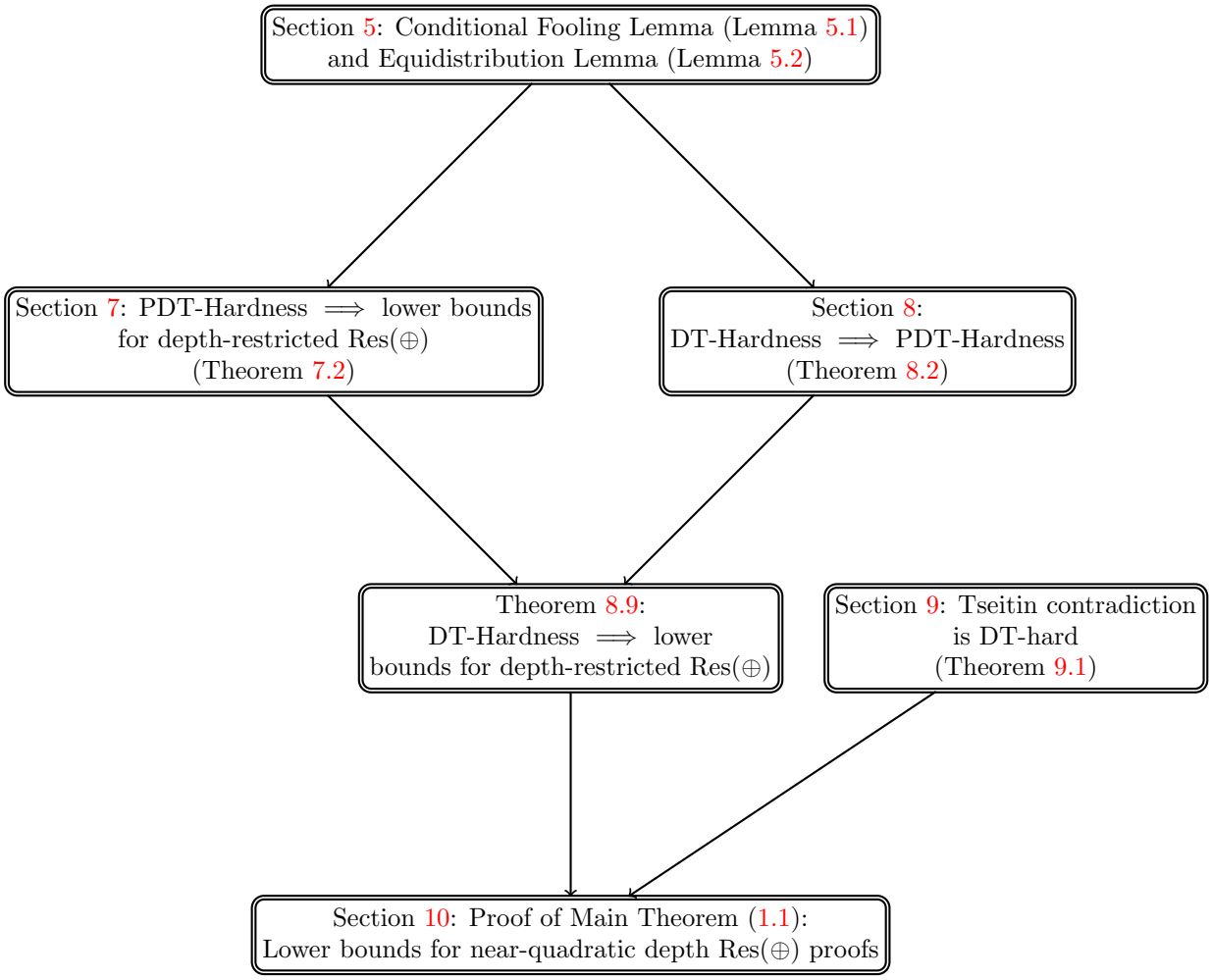
Thus, the Conditional Fooling Lemma crucially allows us to do bottleneck counting wrt the amortized closure of affine spaces at the end of a random walk after re-start from a node v, when the walk is being triggered by a suitably lifted distribution. However, we still need to find a concrete lifted distribution that would boost the success probability of the walk on a Res($\oplus$) DAG. Designing such a lifted distribution and analyzing the corresponding random walk is our second contribution. To do this, we first formulate a certain notion of hardness against *parity decision trees* (PDT) of depth q that are promised that the input comes from a g-lifted distribution, conditioned on lying inside an affine space A (corresponding to the affine space of the node of the DAG from where a walk would restart) with $|\tilde{\mathrm{Cl}}(A)| \leq p$. We call this notion as (p, q)-PDT hardness, wrt the gadget g. This notion was inspired by the analysis of certain combinatorial games by Alekseev and Itsykson [AI25]. Section 7 shows that Res($\oplus$) proofs of size s for a lifted CNF $\Phi \circ g$ must have depth $\Omega\left(\frac{pq}{\log s}\right)$, whenever g is nice and (Φ, g) is (p, q)-PDT-Hard. This requires the use of the Conditional Fooling Lemma. With its help, we have reduced the problem of proving size-depth tradeoff lower bounds for Res($\oplus$) DAGs to that of proving certain kind of hardness against PDTs wrt lifted distributions.

At this point, it is natural to wonder if the problem can be reduced even further to proving appropriate hardness against ordinary decision trees (DT). This is exactly what we do after formulating such a notion of hardness. This notion we call (p, q)-DT-Hardness which was stated in Section 1.1 and is formally stated in Section 8. Finally, we prove in Section 8 that if any CNF formula Φ is $(p, p+q)$-DT-hard, then the pair (Φ, g) is (p, q)-PDT Hard. For this reduction, we make use of our Equidistribution Lemma, Lemma 5.2, which is independently interesting.

In this way, we get a novel lifting theorem: (p, q)-DT hardness of any CNF Φ lifts to yield that size s Res($\oplus$) proofs of $\Phi \circ g$ must have depth $\Omega\left(\frac{pq}{\log s}\right)$ as long as g is a nice gadget whose block size is not too large. All that remains is to find a CNF that is appropriately DT hard and to find a g that is nice and does not have a very large block size. The latter is simple to find: any bent function like Inner-product on $O(\log n)$ many bits suffices. To find the former, we show in Section 9, that a Tseitin formula defined over a (n, d, λ) expander graph, for some constant $\lambda < 1$, is $(\Omega(n), \Omega(n))$-DT-Hard. This, finally, gives us the required formula for proving our main result of exponential lower bounds on the size of Res($\oplus$) proofs that have depth $N^{2-\epsilon}$ for any constant $\epsilon > 0$, where N is the number of variables. The number of clauses of the lifted formula is $N^{O(1)}$. It is worth noting that this last result of ours has been directly used very recently by Byramji and Impagliazzo [BI25b].

4 Organization of the Rest of our Paper

We first describe the CNF in our final result in Section 6. We prove our final result (Theorem 1.1) in Section 10, which uses machinery developed in Sections 5, 7, 8, and 9. The following figure, denoting the dependency graph of the various components needed for our main result, depicts the organization of our work.



Section 9 can be read independently of all other sections.

5 Conditional fooling lemma

Throughout this section, assume the gadget $g : \mathbb{F}_2^b \to \mathbb{F}_2$ has the following property.

- For all $S \subseteq [b]$, $|\hat{g}(S)| \leq \frac{1}{n^{2+\eta}}$ for some constant $\eta > 0$

An explicit example of such a gadget is the Inner Product gadget on $(4 + 2\eta)\log(n)$ bits.

In this section, we will establish a key result that shows that lifted distributions *fool* amortized closure (Lemma 5.1).

In the following, we state a fact that will be, in some sense, a significant generalization of the following simple, well known fact: If $B \subseteq A \subseteq \mathbb{F}_2^{nb}$ are two affine spaces, then $Pr_{x\sim A}[x \in B] \leq 2^{\text{codim}(A)-\text{codim}(B)}$. This fact was generalized recently by Efremenko and Itsykson [EI25]. Let y be an extendible assignment to the variables in closure of A, i.e. $\text{Cl}(A)$. Then, Lemma 5.1 of [EI25], that they point out is their key lemma for improving the lower bound on resolution over parities, shows the following:

$$\Pr_{x\sim A\cap C_y}[x \in B] \leq 2^{|\tilde{\text{Cl}}(A)|-|\tilde{\text{Cl}}(B)|}. \tag{5.1}$$

Note here we cannot hope to work with co-dimension of B and A as shown in the counter-example to the naive idea of conditional fooling as discussed earlier in Section 3. The argument in [EI25] uses a convenient property of amortized closure, combined with simple linear algebra. In another direction, Bhattacharya, Chattopadhyay and Dvorak [BCD24] showed the following: if g is a gadget with certain properties, then the following is true for every $z \in \{0,1\}^n$:

$$\Pr_{x \sim G^{-1}(z)}[x \in B] \leq 2^{-\Omega(\mathrm{codim}(B)/b)} \tag{5.2}$$

Below, we prove our main lemma which has the features of both (5.1) and (5.2).

Lemma 5.1 (Conditional Fooling Lemma)**.** *Let $B \subseteq A \subseteq \mathbb{F}_2^{nb}$ be affine subspaces such that $|\tilde{Cl}(B)| = |\tilde{Cl}(A)| + k$. Let $y \in \mathbb{F}_2^{Cl(A)\times[b]}$ be an A-extendable closure assignment, and let μ be a distribution on $\mathbb{F}_2^n$ such that $z|_{Cl(A)} = G(y)$ for every $z \in supp(\mu)$. Then,*

$$Pr_{x \sim G^{-1}(\mu) \cap C_y}[x \in B | x \in A] \leq \left(\frac{3}{4}\right)^k$$

Currently, we do not know of a short argument to prove this. We prove it here in steps, establishing some equidistribution properties of gadgets with small Fourier coefficients wrt *safe* affine spaces that seem independently interesting.

Lemma 5.2 (Equidistribution Lemma)**.** *Let $A \subseteq \mathbb{F}_2^{nb}$ be a safe affine space. Then, for all $z \in \mathbb{F}_2^n$,*

$$Pr_{x \sim A}[G(x) = z] \in \left[1 \pm o(n^{-1-\eta/2})\right] \frac{1}{2^n}$$

Proof. Let $\mathrm{codim}(A) = m$. Fix a $z \in \mathbb{F}_2^n$. Since $\Pr_{x \leftarrow \mathbb{F}_2^{nb}}[x \in A] = 2^{-m}$, it suffices to show that

$$\Pr_{x \sim \mathbb{F}_2^{nb}}[x \in A \wedge G(x) = z] \in \left[\frac{1 - o(n^{-1-\eta/2})}{2^{m+n}}, \frac{1 + o(n^{-1-\eta/2})}{2^{m+n}}\right].$$

Let $M \in \mathbb{F}_2^{m \times nb}$ be a matrix for the equations defining A. Since A is safe, there exist m blocks such that one can choose one column from each block, such that those columns are linearly independent. WLOG (for notational convenience) assume those blocks are $1, 2, \cdots, m$, and from block j we choose column a_j.

We first rewrite the system of equations in a more convenient form. Since the matrix M restricted to column set $S = \{(j, a_j) | 1 \leq j \leq m\}$ is invertible, we can perform row operations on M so that the submatrix $M_{[m],S}$ becomes I_m. Let ℓ_i denote the i-th row of this modified matrix. Thus, for every $i \in [m]$, there exists a $c \in [b]$ such that l_i has a non-zero entry at coordinate (i, c), and for every $i' \neq i$, $l_{i'}$ has a zero entry at coordinate (i, c). An easy but crucial consequence of this is the following.

Observation 5.3. For every subset $T \subseteq [m]$, the vector $\sum_{j \in T} \ell_j$ has a non-zero coordinate in the j-th block for each $j \in T$.

Suppose the system of equations in this basis is

$$\begin{aligned} \langle \ell_1, x \rangle &= c_1 \\ \langle \ell_2, x \rangle &= c_2 \\ &\cdots\cdots \\ \langle \ell_m, x \rangle &= c_m \end{aligned}$$

Notation: for an assignment $x \in \mathbb{F}_2^{nb}$, we denote by $x(i) \in \mathbb{F}_2^b$ the restriction of x to the i'th block.

Let $p \coloneqq \Pr_{x \sim \mathbb{F}_2^{nb}}[x \in A \wedge G(x) = z]$. We have

$$p = \mathrm{E}_x \left(\prod_{j=1}^{n} \left(\frac{1 + (-1)^{g(x(i)) + z_i}}{2} \right) \prod_{j=1}^{m} \left(\frac{1 + (-1)^{\ell_j(x) + c_j}}{2} \right) \right)$$

Expanding the RHS, we get the following expression:

$$p - \frac{1}{2^{n+m}} = \sum_{\substack{S \subseteq [n] \\ T \subseteq [m] \\ S \cup T \neq \phi}} \mathrm{E}_x \left[\frac{(-1)^{\sum_{i \in S} (g(x(i)) + z_i) + \sum_{j \in T} (\ell_j(x) + c_j)}}{2^{n+m}} \right]$$

For $S \subseteq [n], T \subseteq [m]$ let $f_{S,T}(x) \coloneqq (-1)^{\sum_{i \in S} g(x(i)) + \sum_{j \in T} \ell_j(x)}$ and $u_{S,T} \coloneqq \sum_{i \in S} z_i + \sum_{j \in T} c_j$. We have

$$p - \frac{1}{2^{n+m}} = \frac{1}{2^{n+m}} \sum_{\substack{S \subseteq [n] \\ T \subseteq [m] \\ S \cup T \neq \phi}} (-1)^{u_{S,T}} \mathrm{E}_x[f_{S,T}(x)]$$

We start by showing that $\mathrm{E}_x[f_{S,T}(x)]$ vanishes unless $T \subseteq S$. This is where we use the safety of A.

Claim 5.4. If $T \not\subseteq S$, $\mathrm{E}_x[f_{S,T}(x)] = 0$

Proof. Let $u \in T \setminus S$. By Observation 5.3, there exists a coordinate k in the u-th block on which $\sum_{j \in T} \ell_j$ is non-zero: $\sum_{j \in T} (\ell_j)_{(u,k)} = 1$. Since $u \notin S$, this coordinate does not affect $\sum_{i \in S} g(x(i))$. So we have that for all x, $f_{S,T}(x) = -f_{S,T}(x \oplus e_{u,k})$. Therefore, exactly half of the x's have $f_{S,T}(x) = 1$ and the result follows. □

It now suffices to bound the terms where $T \subseteq S$. We do this using the fact that all Fourier coefficients of g are small.

Claim 5.5. If $T \subseteq S$, $|\mathrm{E}_x[f_{S,T}(x)]| \leq n^{-(2+\eta)|S|}$.

Proof. Let $g^{\oplus S} : \mathbb{F}_2^{b \times |S|} \to \mathbb{F}_2$ be the XOR of $|S|$ disjoint copies of g; for $y \in \mathbb{F}_2^{b \times |S|}$,

$$g^{\oplus S}(y) = \left(\sum_{i \in S} g(y(i)) \right) \pmod 2$$

Note that $||\widehat{g^{\oplus S}}||_\infty = (||\hat{g}||_\infty)^{|S|} \leq n^{-(2+\eta)|S|}$. Therefore,

$$|\mathrm{E}_x[f_{S,T}(x)]| = \left| \widehat{g^{\oplus S}} \left(\mathrm{supp} \left(\sum_{j \in T} l_j \right) \right) \right| \leq n^{-(2+\eta)|S|}.$$

□

Now, we upper bound the magnitude of the error as follows. If $|S| = k$, Claim 5.4 implies that there are at most 2^k possible values of T for which $\mathrm{E}_x[f_{S,T}(x)] \neq 0$. Claim 5.5 implies that the magnitude of each of these terms is at most $n^{-(2+\eta)k}$. Thus, we get that

$$\left|p - \frac{1}{2^{n+m}}\right| = \frac{1}{2^{n+m}} \left| \sum_{\substack{S\subseteq[n]\\ T\subseteq[m]\\ S\cup T\neq\phi]}} (-1)^{u_{S,T}} \mathrm{E}_x[f_{S,T}(x)] \right|$$

$$\begin{aligned}
&\leq \frac{1}{2^{n+m}} \sum_{k=1}^{n} \binom{n}{k} 2^k n^{-(2+\eta)k} \\
&\leq \frac{1}{2^{n+m}} \sum_{k=1}^{n} \exp(k(\log(n) + 1 - (2+\eta)\log(n))) \\
&\leq \frac{1}{2^{n+m}} o(n^{-1-\eta/2})
\end{aligned}$$

This completes the proof.

□

We will now show that the set of pre-images of an arbitrary $z \in \{0,1\}^n$, are approximately equidistributed among the various translates of a safe affine space in the lifted world.

Lemma 5.6. *Let $A \subseteq \mathbb{F}_2^{nb}$ be a safe affine space with codimension m, and let $z \in \mathbb{F}_2^n$ be a target point. Then,*

$$Pr_{x\sim G^{-1}(z)}[x \in A] \in \left[\frac{1 - o(n^{-1-\eta/3})}{2^m}, \frac{1 + o(n^{-1-\eta/3})}{2^m}\right]$$

Proof. Let $A_1 = A, A_2, \cdots, A_M$ be the $M = 2^m$ translates of A. Let $S_j = G^{-1}(z) \cap A_j$. Lemma 5.2 implies $\frac{|S_j|}{|A|} \in \left[\frac{1 - o(n^{-1-\eta})}{2^n}, \frac{1 + o(n^{-1-\eta})}{2^n}\right]$ for all j. We have

$$\begin{aligned}
\Pr_{x\sim G^{-1}(z)}[x \in A] = \frac{|S_1|}{\sum_j |S_j|} &\in \left[\frac{1 - o(n^{-1-\eta/2})}{1 + o(n^{-1-\eta/2})} \times \frac{1}{2^m}, \frac{1 + o(n^{-1-\eta/2})}{1 - o(n^{-1-\eta/2})} \times \frac{1}{2^m}\right] \\
&= \left[\frac{1 - o(n^{-1-\eta/3})}{2^m}, \frac{1 + o(n^{-1-\eta/3})}{2^m}\right]
\end{aligned}$$

□

Using the above, we show below that if $B \subset A$ are two safe affine spaces, then B cannot significantly distinguish the distributions $x \sim (G^{-1}(z) \cap A)$ and $x \sim A$.

Lemma 5.7. *Let $B \subseteq A \in \mathbb{F}_2^{nb}$ be safe affine subspaces such that $codim(B) = codim(A) + 1$. Let $z \in \mathbb{F}_2^n$ be any point. Then,*

$$Pr_{x\sim G^{-1}(z)}[x \in B | x \in A] \leq \frac{1}{2} + o(n^{-1-\eta/4})$$

Proof. Let $m = \mathrm{codim}(A)$. Lemma 5.6 implies $\Pr_{x\sim G^{-1}(z)}[x \in A] \geq \frac{1 - o(n^{-1-\eta/3})}{2^m}$ and $\Pr_{x\sim G^{-1}(z)}[x \in B] \leq \frac{1 + o(n^{-1-\eta/3})}{2^{m+1}}$ Thus,

$$\Pr_{x\sim G^{-1}(z)}[x \in B | x \in A] = \frac{\Pr_{x\sim G^{-1}(z)}[x \in B]}{\Pr_{x\sim G^{-1}(z)}[x \in A]} \leq \frac{1 + o(n^{-1-\eta/3})}{1 - o(n^{-1-\eta/3})} \times \frac{1}{2} \leq \frac{1}{2} + o(n^{-1-\eta/4})$$

□

The structure of safe affine spaces that we have discovered so far allows us to say the following about any two arbitrary affine spaces that are not necessarily safe.

Lemma 5.8. *Let $B \subseteq A \in \mathbb{F}_2^{nb}$ be affine spaces such that $|\tilde{Cl}(B)| = |\tilde{Cl}(A)|+1$ and $codim(B) = codim(A)+1$. Let y be an A-extendable closure assignment, and let $z \in \mathbb{F}_2^n$ be a point such that $z|_{Cl(A)} = G(y)$. Then,*

$$Pr_{x\sim G^{-1}(z)\cap C_y}[x \in B | x \in A] \leq \frac{1}{2} + o(n^{-1-\eta/4})$$

Proof. Let $z = (G(y), w)$. Rewrite the desired probability expression as

$$\Pr_{\tilde{x}\sim G^{-1}(w)}[x \in B_y | x \in A_y]$$

By Corollary 2.27, A_y, B_y are both safe affine subspaces, and $\text{codim}(B_y) = \text{codim}(A_y)+1$. Now the result follows from Lemma 5.7. □

An easy corollary is that the result still holds if we condition only on a subset of the blocks in $\text{Cl}(A)$ instead of all the blocks in $\text{Cl}(A)$.

Corollary 5.9. *Let $B \subseteq A \in \mathbb{F}_2^{nb}$ be affine spaces such that $|\tilde{Cl}(B)| = |\tilde{Cl}(A)| + 1$ and $codim(B) = codim(A) + 1$. Let $S \subseteq Cl(A)$ and let $y \in \mathbb{F}_2^{S\times[b]}$ be a partial assignment. Let $z \in \mathbb{F}_2^n$ be a point such that $z|_S = G(y)$ and $G^{-1}(z) \cap C_y \cap A \neq \phi$. Then,*

$$Pr_{x\sim G^{-1}(z)\cap C_y}[x \in B | x \in A] \leq \frac{1}{2} + o(n^{-1-\eta/4})$$

Proof. Sampling x from $G^{-1}(z) \cap C_y$ can be done as follows: first sample $y^{(1)} \in \mathbb{F}_2^{\text{Cl}(A)\times[b]} \cap C_y$ according to $G^{-1}(z)$, then sample x from $G^{-1}(z) \cap C_{y^{(1)}}$. For each possible $y^{(1)}$ use Lemma 5.8 to upper bound the conditional probability of lying in B conditioned on $y^{(1)}$. Formally, let $\mathcal{D}$ denote the distribution of $x|_{\text{Cl}(A)}$ as $x \sim G^{-1}(z) \cap C_y$. Then,

$$\begin{aligned}\Pr_{x\sim G^{-1}(z)\cap C_y}[x \in B | x \in A] &= \mathrm{E}_{y^{(1)}\leftarrow\mathcal{D}}[\Pr_{x\sim G^{-1}(z)\cap C_{y^{(1)}}}[x \in B | x \in A]] \\ &\leq \frac{1}{2} + o(n^{-1-\eta/4})\end{aligned}$$

□

Now we prove the final result of this section.

Proof. of Lemma 5.1 By convexity, it suffices to prove the statement in the case that μ is concentrated on a single point. Thus, we have to show the following:

> Let $B \subseteq A \subseteq \mathbb{F}_2^{nb}$ be affine spaces with $|\tilde{\text{Cl}}(B)| \geq |\tilde{\text{Cl}}(A)| + k$. Let $y \in \mathbb{F}_2^{\text{Cl}(A)\times[b]}$ be an A-extendable closure assignment. Let $z \in \mathbb{F}_2^n$ be a point such that $G(y) = z|_{\text{Cl}(A)}$. Then,
>
> $$\Pr_{x\leftarrow G^{-1}(z)\cap C_y}[x \in B | x \in A] \leq \left(\frac{3}{4}\right)^k$$

Let $B = W_0 \subseteq W_1 \subseteq W_2 \subseteq \cdots \subseteq W_{l-1} \subseteq W_l = A$ be a sequence of affine subspaces such that $\mathrm{codim}(W_j) = \mathrm{codim}(W_{j+1}) + 1$. We have

$$\Pr_{x \sim G^{-1}(z) \cap C_y}[x \in B | x \in A] = \prod_{j=0}^{l-1} \Pr_{x \sim G^{-1}(z) \cap C_y}[x \in W_j | x \in W_{j+1}]$$

We assume there exists a point in $G^{-1}(z) \cap C_y \cap B$ (as otherwise the conditional probability is 0), so in particular, for all j there exists a point in $G^{-1}(z) \cap C_y \cap C_j$.

By Lemma 2.26 there exist k indices $j \in \{0, 1, \cdots, l-1\}$ such that $|\tilde{\mathrm{Cl}}(W_j)| = |\tilde{\mathrm{Cl}}(W_{j+1})| + 1$. Note that $\mathrm{Cl}(A) \subseteq \mathrm{Cl}(W_{j+1})$ by Lemma 2.24. Invoking Corollary 5.9 for each such index j, where W_j plays the role of B and W_{j+1} that of A, we have

$$\Pr_{x \sim G^{-1}(z) \cap C_y}[x \in W_j | x \in W_{j+1}] \leq \frac{1}{2} + o(1) \leq \frac{3}{4}$$

So, in the product $\prod_{j=0}^{l-1} \Pr_{x \sim G^{-1}(z) \cap C_y}[x \in W_j | x \in W_{j+1}]$, at least k terms are $\leq 3/4$. The result follows. □

6 Description of CNF

6.1 Lifting CNFs

Definition 6.1. For a base CNF Φ on variables $\{z_1, z_2, \cdots, z_n\}$ and a gadget $g : \mathbb{F}_2^n \to \mathbb{F}_2$ we define the lifted CNF $\Phi \circ g$ as follows.

- The set of variables is $\{x_{i,j} | i \in [n], j \in [b]\}$
- For each clause C in Φ, we define the set of clauses $C \circ g$ as follows: let the variables involved in C be $\{x_i | i \in S\}$ and let $\alpha \in \mathbb{F}_2^S$ be the unique assignment to those variables that falsifies C. The set of clauses $C \circ g$ will involve variables from the set $\{x_{i,j} | i \in S, j \in [b]\}$. For every choice of $(a_i | i \in S)$ where $a_i \in g^{-1}(\alpha_i)$ we add the following clause to $C \circ g$:

$$\vee_{i \in S} \left[\vee_{j \in [b]} [x_{i,j} \neq \alpha_{i,j}] \right]$$

- The lifted CNF $\Phi \circ g$ is the conjunction of $C \circ g$ for every $C \in \Phi$.

The semantic interpretation of $\Phi \circ g$ is as follows:

$$\Phi \circ g(x) = \Phi(g(x_{1,1}, x_{1,2}, \cdots, x_{1,b}), g(x_{2,1}, x_{2,2} \cdots, x_{2,b}), \cdots, g(x_{n,1}, x_{n,2}, \cdots, x_{n,b}))$$

Thus if Φ is unsatisfiable, so is $\Phi \circ g$.

If the largest width of a clause in Φ is w and Φ has m clauses, the number of clauses in $\Phi \circ g$ will be at most $m 2^{bw}$. In particular, if $m \leq \mathrm{poly}(n), b = O(\log(n))$ and $w = O(1)$ then the number of clauses of $\Phi \circ g$ is bounded by $\mathrm{poly}(n)$.

6.2 Choice of our base CNF

The base CNF we shall use is the Tseitin contradiction over an expander graph, lifted with an appropriate gadget. Let $G = (V, E)$ be a $(|V|, d, \lambda < 0.95)$ expander with $|V|$ odd and $d = O(1)$. The base CNF Φ has variables $z_{u,v}$ for $(u, v) \in E$. For each $v \in V$ we express the constraint $\sum_{(v,w) \in E} z_{v,w} \equiv 1 \pmod 2$ using $2^d = O(1)$ clauses. This system is unsatisfiable because adding

up all the equations yields $0 \equiv 1 \pmod 2$.

The property of G we shall use is isoperimetric expansion (which follows from Cheeger's inequality [Che71]):

Lemma 6.2. *For any $S \subseteq V$, the cut $E(S, V \setminus S)$ has at least $\frac{d}{40}\min(|S|, |V \setminus S|)$ edges.*

Explicit constructions of such graphs were provided in [LPS88] and [Mar73].

6.3 Lifted CNF

We lift Φ with an appropriate gadget. We will take the gadget $g : \mathbb{F}_2^b \to \mathbb{F}_2$ to be the Inner Product function $g = \text{IP}$ on $b = (4+\eta)\log(n)$bits for some arbitrarily small constant $\eta > 0$.

$$\text{IP}(x_1, x_2, \cdots, x_{b/2}, y_1, y_2, \cdots, y_{b/2}) = (x_1 y_1 + \cdots + x_{b/2} y_{b/2}) \pmod 2$$

This satisfies the property mentioned in Section 5: $||\hat{g}||_\infty \leq n^{-2-\eta/2}$.

Theorem 6.3. *For $g = IP$ on b bits, $||\hat{g}||_\infty \leq 2^{-b/2}$*

The CNF for which we will prove our depth restricted lower bounds is $\Psi = \Phi \circ g$.

If we take $d = 4$ we get graphs with $\lambda \leq 0.87$. Then, the final resulting CNF has number of variables $N = O(m \log m)$, width $16\log(m)$ and number of clauses $O(m^{17}) = \tilde{O}(N^{17})$.

If we took $d = 3$ instead, we would get number of clauses $\tilde{O}(N^{13})$. An issue with $d = 3$ is that we are taking the right hand sides of each parity constraint to be 1, and 3-regular graphs with m vertices do not exist when m is odd. One can get around this by defining general Tseitin contradictions where the right hand sides can be anything which sum up to 1 (mod 2). The proof we present in our paper can be easily modified to work for general Tseitin contradictions. For simplicity we do not do this here.

7 The Utility of (p, q)-PDT Hardness

Alekseev and Itsykson [AI25] introduced the 'random walk with restarts' approach to prove superlinear lower bounds on depth of Res($\oplus$) proofs of small size. To analyze their random walk with restarts, [AI25] uses certain elaborate games. We find it more convenient to analyze random walks using the language of decision trees. In particular, this allows us naturally to bring in the notion of a hard distribution that seems crucial to boost the success probability of our random walk with restart significantly, all the way from $2^{-n/\log(n)}$ to a constant. In this section, we formalize our notion which we call (p, q)-PDT hardness. We point out that our notion here is a significant refinement of the ideas of Bhattacharya, Chattopadhyay and Dvorák [BCD24] where as well random walks on lifted distributions were analyzed, but without restarts.

We first set up some notation to define our hardness notion. For a parity decision tree T and a point x, define the affine subspace $A_x(T)$ to be the one corresponding to the set of inputs y that traverse the same path in T as x does. More formally, $A_x(T)$ is defined as follows: suppose on input x, T queries the linear forms $\ell_1, \cdots, \ell_d$ and gets responses $c_1, c_2, \cdots, c_d$ respectively. Then, $A_x(T) = \{y | \langle \ell_j, y \rangle = c_j \forall j \in [d]\}$.

We are ready now to introduce the notion of a hard set of partial assignments that will abstract our requirements for finding a deep node in the proof DAG.

Definition 7.1. Let Φ be a CNF formula on n variables and $g : \mathbb{F}_2^b \to \mathbb{F}_2$ be a gadget. A set of partial assignments $P \subseteq \{0,1,*\}^n$ is (p,q)-*PDT-hard* wrt (Φ, g) if the following properties hold:

- **Non-emptiness:** $P \neq \phi$
- **No falsification:** No partial assignment in P falsifies any clause of Φ.
- **Downward closure:** If $\rho \in P$ and $\tilde{\rho}$ is obtained from ρ by unfixing some of the bits set in ρ, then $\tilde{\rho} \in P$
- **Hardness against parity decision trees:** Let $A \subseteq \mathbb{F}_2^{nb}$ be an affine space with $|\tilde{\text{Cl}}(A)| \leq p$. Let $y \in \mathbb{F}_2^{\text{Cl}(A)\times[b]}$ be an A-extendable closure assignment such that $\alpha = G(y) \in P$. Then, there exists a distribution μ on $\mathbb{F}_2^n$ such that the following properties hold:
  1. $z|_{\text{Cl}(A)} = \alpha$ for all $z \in \text{supp}(\mu)$
  2. Let T be any parity decision tree (with input nb bits) of depth $\leq q$. For any x, define $\tilde{A}(x) = A_T(x) \cap A \cap C_y$. With probability $\geq 1/3$, as x is sampled from $G^{-1}(\mu) \cap A \cap C_y$, it holds that $G(x)|_{\text{Cl}(\tilde{A}(x))} \in P$ [2].

The pair (Φ, g) is (p,q)-*PDT-hard* if it admits a (p,q)-PDT-hard set of partial assignments.

We now state the main result of this section that shows (p,q)-PDT-hardness of a CNF is sufficient to get us good lower bound on depth of a refutation of the lifted formula, assuming the size of the refutation is small.

Theorem 7.2. *Let Φ be a CNF on n variables and let $g : \mathbb{F}_2^b \to \mathbb{F}_2$ be a gadget with $||\hat{g}||_\infty \leq n^{-2-\eta}$ for some $\eta > 0$. Suppose (Φ, g) is (p,q)-PDT-hard. Then, any Res($\oplus$) refutation of $\Phi \circ g$ of size s must have depth at least $\Omega\left(\frac{pq}{\log(s)}\right)$.*

To prove the above, we will first establish the following lemma. This lemma essentially tells us that as long as we are at a node whose associated affine space satisfies some convenient properties, we are assured to find another node at a distance q from our starting node whose corresponding affine space continues to have reasonably convenient properties.

Lemma 7.3. *Suppose P is a set of partial assignments that is (p,q)-PDT-hard wrt (Φ, g) where $g : \mathbb{F}_2^b \to \mathbb{F}_2$ is a gadget with $||\hat{g}||_\infty \leq n^{-2-\eta}$ for some constant $\eta > 0$. Let Π be a Res($\oplus$) refutation of $\Phi \circ g$ of size s. Let v be a node in Π such that $|\tilde{Cl}(A_v)| \leq p$, and let $y \in \mathbb{F}_2^{Cl(A_v)\times[b]}$ be an extendable closure assignment for A_v such that $G(y) \in P$. Then, there exists another node w in Π such that:*

1. *There exists a length q path from v to w in Π.*
2. *There exists an extendable closure assignment for A_w, $\tilde{y}$, such that $G(\tilde{y}) \in P$.*
3. *$|\tilde{Cl}(A_w)| \leq |\tilde{Cl}(A_v)| + 2\log(s)$*

Proof. Let μ be the hard distribution guaranteed to exist by the definition of (p,q)-PDT-hardness.Let T be the following parity decision tree: on any input x, it simulates the queries made by Π starting from node v for q steps. For any $x \in A_v$, define $\text{END}_q(x)$ to be the node of Π reached by x starting from v after q steps. (In case Π on x reaches a leaf within q steps starting from v, define $\text{END}_q(x)$ to be that leaf.)

[2] $G^{-1}(\mu) \cap A \cap C_y$ is non-empty by Lemma 5.2 applied on the nice affine space A_y

We have $A_T(x) \cap A_v \subseteq A_{\mathrm{END}_q(x)}$. Let $\mathrm{GOOD} = \{x | G(x|)_{\mathrm{Cl}(\tilde{A}_v(x))} \in P\}$ (recall, $\tilde{A}_v(x) = A_T(x) \cap A_v \cap C_y$). The definition of (p,q)-PDT-hardness guarantees that $\Pr_{x \leftarrow G^{-1}(\mu) \cap A \cap C_y}[x \in \mathrm{GOOD}] \geq 1/3$.
Let $\mathcal{N} = \{\mathrm{END}_q(x) | x \in \mathrm{GOOD}\}$. Note that since no assignment in P falsifies any clause of Φ, no vertex in $\mathcal{N}$ is a leaf - and therefore, there is a length q walk from v to w for all $w \in \mathcal{N}$ (i.e., the parity decision tree does not terminate before q queries if $x \in \mathrm{GOOD}$). Also, $A_T(x) \cap A_v \cap C_y \subseteq A_{\mathrm{END}_q(x)}$, so $\mathrm{Cl}(A_{\mathrm{END}_q(x)}) \subseteq \mathrm{Cl}(A_T(x) \cap A_v \cap C_y)$, so $x \in \mathrm{GOOD}$ implies $G(x)|_{\mathrm{Cl}(\mathrm{END}_q(x))} \in P$ (since P is downward closed). Thus, properties (i) and (ii) are satisfied for all $w \in \mathcal{N}$. To complete the proof, we have to find a $w \in \mathcal{N}$ such that $|\tilde{\mathrm{Cl}}(A_w)| \leq |\tilde{\mathrm{Cl}}(A_v)| + 2\log(s)$.
Since $|\mathcal{N}| \leq s$, there exists a $w \in \mathcal{N}$ such that $\Pr_{x \leftarrow G^{-1}(\mu) \cap C_y \cap A}[\mathrm{END}_q(x) = w] \geq \frac{1}{3s}$. In particular, this implies

$$\Pr_{x \leftarrow G^{-1}(\mu) \cap C_y}[x \in A_w | x \in A] \geq \frac{1}{3s}$$

Lemma 5.1 then implies $|\tilde{\mathrm{Cl}}(A_w)| \leq |\tilde{\mathrm{Cl}}(A_v)| + 2\log(s)$

□

Now we are ready to prove our main result for this section, by repeatedly making use of Lemma 7.3.

Proof of Theorem 7.2. Let Π be a Res($\oplus$) refutation of $\Phi\circ$. We shall inductively find vertices $v_1, v_2, \ldots, v_j$ in Π for $j \leq \frac{p}{2\log(s)}$ such that:

- $\mathrm{depth}(v_j) \geq jq$
- $|\tilde{\mathrm{Cl}}(A_{v_j})| \leq 2j\log(s)$
- There exists an extendable closure assignment y_j for A_{v_j} such that $G(y_j) \in P$

For $j = 0$ we pick the root. To get v_{j+1} we apply Lemma 7.3 to v_j. We can continue this way as long as $|\tilde{\mathrm{Cl}}(A_{v_j})| \leq p$. Hence, we do this for $j = \left\lfloor \frac{p}{2\log s} \right\rfloor$ many steps. In the end, we get a node at depth $\Omega\left(\frac{pq}{\log(s)}\right)$. □

8 Lifting DT-hardness to PDT-hardness

Note that (p,q)-PDT-hardness is a notion that measures the hardness of a set of partial assignments against parity decision trees. We define an analogous notion of hardness against ordinary decision trees - which we call DT hardness. We then exhibit a lifting theorem: a DT-hard formula is also PDT-hard. Given this, to establish lower bounds against depth-restricted Res($\oplus$), is suffices to argue against ordinary decision trees.

Definition 8.1 ((p,q)-DT hardness)**.** For a CNF Φ on n variables, call a set of partial assignments $P \subseteq \{0,1,*\}^n$ to be (p,q)-DT-hard if the following hold:

- **Non-emptiness:** $P \neq \phi$
- **No falsification:** No partial assignment $\rho \in P$ falsifies any clause of Φ.
- **Downward closure:** For any $\rho \in P$ and any $j \in [n]$, if $\tilde{\rho}$ is obtained by setting $\rho(j) \leftarrow *$, then $\tilde{\rho} \in P$
- **Hard for decision trees:** For any $\rho \in P$ which fixes at most p variables, there exists a distribution μ_ρ on the assignment to unfixed variables such that the following holds:

– Let T be a decision tree of depth q querying the unfixed variables. If we sample an assignment to the unfixed variables from μ_ρ and run T for q steps, the partial assignment seen by the tree [3] also lies in P with probability $\geq 1/2$.

The CNF Φ is (p, q)-DT hard if it admits a set of (p, q)-DT-hard partial assignments.

The primary goal in this section is to establish a *lifting theorem* from DT-hardness to PDT-hardness. The main result of this section is the following.

Theorem 8.2. *If Φ is $(p, p+q)$-DT-hard, then (Φ, g) is (p, q)-PDT-hard where $g : \mathbb{F}_2^b \to \mathbb{F}_2$ is any gadget with $||\hat{g}||_\infty \leq n^{-2-\eta}$ for some constant $\eta > 0$ and $b = O(\log n)$.*

Remark 8.1. *Note that in the definition of PDT hardness, the hard distribution for an affine space A and an extendable closure assignment y was allowed to depend on both A and y. In our proof of Theorem 8.2, our hard distribution is independent of A and depends only on $G(y)$. The hard distribution in the proof simply is the corresponding DT-Hard distribution.*

Conventions

- Throughout this section, we shall be analyzing the execution of parity decision trees on some input. Whenever we say a phrase like {intersection of first t queries made by T}, we mean the following affine subspace: let $\ell_1, \cdots, \ell_t$ be the first t linear forms queried by T and let $b_1, b_2, \cdots, b_r$ be the responses received. Then,

  $$\{\text{intersection of first } t \text{ queries made by } T\} \triangleq \{x | \langle l_j, x\rangle = b_j] \ \forall j \in [t]\}\}$$

  Whenever we talk about the *current subspace* of a PDT T, we shall mean the subspace

  $$A_{\text{safe}} \cap \{\text{intersection of queries made by } T \text{ so far}\}$$

  (Here A_{safe} is a safe affine subspace which will be clear from the context.)

- For $x \in \mathbb{F}_2^{nb}$ and $S \subseteq [n]$ let $\text{PROJ}(x, S) \subseteq \mathbb{F}_2^{S \times [b]}$ denote the projection of x on the blocks in S.

- For an affine space $A \subseteq \mathbb{F}_2^{nb}$ and $S \subseteq [n]$ define $A[\backslash S]$ to be the projection of A on the blocks in $[n] \setminus S$, i.e. $A[\backslash S] \triangleq \{\text{PROJ}(x, [n] \setminus S) \,|\, x \in A\}$.

8.1 High level overview

Before proceeding, we give a high-level overview of our proof. Suppose Φ is $(p, p+q)$-DT-hard. Let $P \subseteq \{0, 1, *\}^n$ be a hard set of assignments, and for each $\rho \in P$ with $|\rho| \leq p$ let μ_ρ be the hard distribution. We show the same family of hard distributions establishes (p, q)-PDT-hardness. Suppose this is not the case. Then, the following things exist:

- A $\rho \in P$ with $|\rho| \leq p$. Let $\mu = \mu_\rho$ be its DT-hard-distribution.
- An affine subspace $A \subseteq \mathbb{F}_2^{nb}$ with $\text{Cl}(A) = \text{fix}(\rho)$ and $|\tilde{\text{Cl}}(A)| \leq p$.
- An A-extendable closure assignment $y \in \mathbb{F}_2^{\text{Cl}(A) \times [b]}$ such that $G(y) = \rho$
- A PDT T_{par} of depth q such that as x is sampled from $G^{-1}(\mu) \cap A \cap C_y$, with probability $\geq 2/3$, the following holds:

[3] The partial assignment seen by the tree is the partial assignment formed by bits queried by the tree and bits fixed by ρ.

– Let $A(q)$ be the affine space seen by T_{par} after q queries. Then, $G(x)|_{\text{Cl}(A(q))} \notin P$

Given this, we attempt to construct an ordinary decision tree T_{ord} of depth $p+q$ which contradicts the DT-hardness of the distribution μ_ρ. Informally, we want that as z is sampled from μ, with probability $\geq 1/2$ the partial assignment seen by T_{ord} (this includes the bits queried by T_{ord} and bits fixed by ρ) does not lie in P.

A first approach would be to try to use query-to-communication lifting results such as [GPW17; Cha+21] which show that one can simulate the execution of any communication protocol on a uniformly random preimage of z, with only a few queries to z (where the x variables in each IP gadget belong to Alice and y variables belong to Bob). Note that if the variables are distributed between Alice and Bob in any arbitrary fashion, a linear query can be processed using only one bit of communication - so low depth PDTs can be seen as low cost communication protocols.

The issue is that the input to T_{par} is not coming from $G^{-1}(z)$ - it is coming from $G^{-1}(z) \cap A \cap C_y$. The set $A \cap C_y$ is in general not a product set over the Alice and Bob variables- so it is unclear how to use known query-to-communication lifting results in a blackbox fashion here (it might be possible to modify the proof of [GPW17] to work for this case – but as we explain soon, this will not be needed). However, we have more structure: we are working with linear queries instead of general communication protocols, and moreover, $A \cap C_y$ projected on the blocks outside $\text{Cl}(A)$ is a safe subspace, so we have tools such as the Equidistribution Lemma in our hand.

It turns out that the tools developed in Section 5 do indeed help us do a [GPW17]-style simulation even when the input is sampled from $G^{-1}(z) \cap A \cap C_y$. Before formally describing the simulation, we provide an informal version.

We are given a PDT T_{par}, and we want to simulate the execution of $T_{\text{par}}(x)$ when x is sampled from $G^{-1}(z) \cap A \cap C_y$, by making only a few queries to z. Let's make some straightforward simplifications: we can assume WLOG the linear queries are supported on blocks outside $\text{Cl}(A)$, since the other variables are fixed. So we can assume the input is sampled from $G^{-1}(z') \cap A_y$ where $z' = z|_{[n]\setminus \text{Cl}(A)}$. Denoting A_y by A_{safe} our goal becomes the following:

Given an unknown input z, a safe affine space A_{safe} and a PDT T_{par}, simulate the execution of $T_{\text{par}}(x)$ when x is sampled from $G^{-1}(z) \cap A_{\text{safe}}$, by making a small number of queries to z.

First, we modify T_{par} into a canonical form (which we call CANONIZE$(T_{\text{par}}, A_{\text{safe}})$ below). The execution of CANONIZE$(T_{\text{par}}, A_{\text{safe}})$ proceeds as follows:

- Suppose, ℓ is the linear form queried by T_{par} at time t.
- Let, $A(t)$ be the affine subspace visited by T_{par} at time t. That is,

$$A(t) = A_{\text{safe}} \cap \{\text{intersection of first } t \text{ queries made by } T_{\text{par}}\}$$

- **Case 1:** Suppose, querying ℓ changes $\text{Cl}(A(t-1))$ to CL_{new}. Then, CANONIZE$(T_{\text{par}}, A_{\text{safe}})$ first queries all coordinates in the blocks of $\text{CL}_{new} \setminus \text{Cl}(A(t-1))$ - and then it queries ℓ
- **Case 2:** Suppose querying ℓ does not change $\text{Cl}(A(t-1))$ (note that this depends only on ℓ and not on the response received). Then, CANONIZE$(T_{\text{par}}, A_{\text{safe}})$ simply queries ℓ.

Now the simulator tries to simulate CANONIZE$(T_{\text{par}}, A_{\text{safe}})$ instead of T_{par}. The advantage of working with CANONIZE$(T_{\text{par}}, A_{\text{safe}})$ is that it has a fixed closure assignment at any

given point of execution. (It is easy to see that at any point of execution, the subspace of $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$ is simply the subspace of T_{par} with all the closure variables fixed.) Conditioned on this closure assignment, the subspace projected on the non-closure blocks is a safe affine subspace - and to analyze that, we have the machinery developed in Section 5 in our hands.

Looking ahead, it turns out that the simulator does not need to query anything in Case 2. It simply samples a uniform random bit $b \in \mathbb{F}_2$ and feeds it to T_{par}. In Case 1, the simulator queries z_i for all $i \in \text{CL}_{new}$ (except those that were previously queried). Now it has to feed a value $\beta \in \mathbb{F}_2^{\text{Cl}_{new} \times [b]}$ to T_{par} to continue the simulation. It has to generate this value in such a manner that the distribution of β matches the distribution of the actual response if x were sampled from $G^{-1}(z) \cap A(t-1)$. We shall show that this distribution is actually the same (upto small ℓ_1 distance) for all w such that $w_i = z_i \; \forall i \in \text{CL}_{new}$. Thus, knowledge of the coordinates of z in CL_{new} suffices to generate β accurately so that the simulation can proceed - the simulator simply picks an arbitrary w which agrees with z on CL_{new} and samples from $G^{-1}(w) \cap [A(t-1)]$ (this is how the simulator saves on the total number of queries). Proving this requires tools developed in Section 5.

We describe the conversion of a PDT to its canonical form in more detail in Section 8.3. We prove our main lifting theorem in Section 8.4. Section 8.2 contains an auxiliary lemma required later in the proof; the reader is advised to skip this section until it is finally used (in Section 8.4).

8.2 Auxiliary Lemmata

Lemma 8.3. *Let $A \subseteq \mathbb{F}_2^{nb}$ be an affine space and let $S \subseteq [n]$ be a set of blocks such that $A[\setminus S]$ is safe. Let $g : \mathbb{F}_2^b \to \mathbb{F}_2$ be a gadget with $||\hat{g}||_\infty \leq n^{-2-\eta}$ for some constant $\eta > 0$. Let $y \in \mathbb{F}_2^{S \times [b]}$ be an A-extendable assignment. Let $z, w \in \mathbb{F}_2^n$ such that $z_i = w_i = g(y(i)) \; \forall i \in S$.*

Then,

$$\frac{|G^{-1}(z) \cap A \cap C_y|}{|G^{-1}(w) \cap A \cap C_y|} \in \left[1 - n^{-1-\eta/6}, 1 + n^{-1-\eta/6}\right] \tag{8.1}$$

Proof. Let $z = (G(y), \tilde{z})$ and $w = (G(y), \tilde{w})$. Then, $|G^{-1}(z) \cap A \cap C_y| = |G^{-1}(\tilde{z}) \cap A_y| \in \left[1 \pm n^{-1-\eta/3}\right] \frac{|A_y|}{2^{n-|S|}}$, where the second inequality follows from the Equidistribution Lemma as A_y is a safe affine space. We have similar bounds for $|G^{-1}(w) \cap A \cap C_y|$. Combining them, we get that the LHS of (8.1) is in $\left[\frac{1 - n^{-1-\eta/3}}{1 + n^{-1-\eta/3}}, \frac{1 + n^{-1-\eta/3}}{1 - n^{-1-\eta/3}}\right] \subseteq [1 - n^{-1-\eta/6}, 1 + n^{-1-\eta/6}]$, as desired.

□

8.3 Canonical Parity Decision Trees

Definition 8.4. Let $A \subseteq \mathbb{F}_2^{nb}$ be an affine space. A set of blocks $S \subseteq [n]$ is *A-compatible* if the following holds: let $y \in \mathbb{F}_2^{S \times [b]}$ be any A-extendable assignment. Then, $\text{Cl}(A \cap C_y) = S$. (This holds for one A-extendable assignment if and only this holds for all A-extendable assignments.)

Informally, S is A-compatible if and only if querying all the bits in S changes $\text{Cl}(A)$ to S. Observe that any A-compatible set must contain $\text{Cl}(A)$.

Definition 8.5. Let $A \subseteq \mathbb{F}_2^{nb}$ be an affine space. A linear form ℓ is A-stationary if $\text{Cl}(A) = \text{Cl}(A \cap \{x | \langle \ell, x \rangle = 0\})$

Informally, ℓ is A-stationary if querying ℓ does not change the closure of A.

Let $A_{\text{safe}} \subseteq \mathbb{F}_2^{nb}$ be a *safe* affine subspace.

Definition 8.6. An A_{safe}-canonical PDT T is a PDT with the following properties:

- The execution of T can be divided into *phases*. At each phase, T either executes a *Type 1 query* or a *Type 2 query*.
- Define $A(t) = A_{\text{safe}} \cap \{\text{affine queries made by } T \text{ till Phase } t\}$
- **Type 1 query:** Select an $A(t-1)$-compatible subset of blocks $S \subseteq [n]$. Query all variables in the blocks in $S \setminus \text{Cl}(A(t-1))$.
- **Type 2 query:** Select an $A(t-1)$-stationary linear form ℓ and query ℓ.

Observe that in a canonical PDT, the affine space $A(t)$ fixes all bits in $\text{Cl}(A(t))$. Thus, a canonical PDT maintains a *closure assignment* at each phase.

Definition 8.7. Let $A_{\text{safe}} \subseteq \mathbb{F}_2^{nb}$ be a safe affine space and let T_{par} be a PDT. Define $\text{SPACE}(t) = A_{\text{safe}} \cap \{\text{first } t \text{ queries made by } T_{\text{par}}\}$. We define its canonized version, $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$ to be a canonical PDT which simulates T_{par}, whose description is given below:

Throughout the execution, $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$ maintains a counter called *count*, starting at 0. It maintains the following invariant:

- At every point of time, $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$ has queried the first *count*-many queries of T_{par}.
- Moreover, it holds that

$$\text{Cl}\big(\text{current subspace of } \text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})\big) = \text{Cl}(\text{SPACE}(\textit{count}))$$

where current subspace of $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$

$$= A_{\text{safe}} \cap \{\text{intersection of all queries made by } \text{CANONIZE}(T_{\text{par}}, A_{\text{safe}}) \text{ so far}\}$$

We formally describe the execution of $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$ below.

$\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$

Input: Query access to $x \in \mathbb{F}_2^{nb}$

Internal variables:

- *count*= number of queries of T_{par} executed so far. Initialized to 0.
- v: Node of T_{par} the tree is currently executing

Execution

While v is not a leaf of T_{par} :

- **Let:**
  - A_{cur} = current affine subspace of $\text{CANONIZE}(T_{\text{par}}, A_{\text{safe}})$
  - $S_{cur} = \text{Cl}(A_{cur}) = \text{Cl}(\text{SPACE}(\textit{count}))$
  - β = assignment to S fixed by A_{cur}
  - ℓ = linear form queried at v by T_{par}
- **Case 1:** ℓ is determined by A_{cur}

- Increase *count* by 1.
- Let b = fixed value of ℓ. Update v according to b.

- **Case 2:** ℓ is a A_{cur}-stationary linear form
  - Start a new phase.
  - Execute a *type 2 query* by querying ℓ. Let b be the response received.
  - Increase counter by 1.
  - Update v according to b.

- **Case 3:** Querying ℓ changes $\mathrm{Cl}(A_{cur})$
  - Start a new phase.
  - Let S_{new} be the new closure. Execute a *type 1 query* by querying all coordinates in the blocks of $S_{new} \setminus S_{cur}$.
  - Start a new phase.
  - Execute a *type 2 query* by querying ℓ. Let b be the response received.
  - Increase counter by 1.
  - Update v according to b.

Let T be any canonical PDT. We define $\mathrm{TRANSCRIPT}(x, T, t)$ to be the transcript of T after t phases on input x. This consists of the following data:

- $v(t)$ = vertex of T reached after t phases
- $A(t) = A_{\text{safe}} \cap \{\text{intersection of queries made during first } t \text{ phases}\}$
- $S_t = \mathrm{Cl}(A(t))$
- $\beta(t)$ = fixed assignment to S_t. (Recall that $A(t)$ fixes all bits in S_t.)
- $\text{LIN-QUERY}(t) = \{(\ell_1, b_1), (\ell_2, b_2), \cdots, (\ell_r, b_r)\}$ is the set of type 2 queries made by T during first t phases and the responses received.

Technically, this above data is redundant because $A(t), S_t, \beta(t), \text{LIN-QUERY}(t)$ are all determined by $v(t)$. But we still describe the transcript in this manner because it facilitates describing how the transcript gets updated at each phase. Let the transcript of T on x be the transcript generated at the unique leaf T reaches on x.

8.4 Proof of Lifting Theorem

In the following, we are given a canonical PDT T which arises as the *canonization* of another PDT T_{par}. We construct an ordinary randomized DT T_{ord}, which, on input z, outputs a transcript of T_{par}. Our goal is the following: over the internal coin tosses of T_{ord}, the output distribution of $T_{\text{ord}}(z)$ should be close in statistical distance to the transcript distribution of $T_{\text{par}}(x)$ when x is sampled from $G^{-1}(z) \cap A_{\text{safe}}$. This theorem is inspired from [GPW17]. We formalize the statement below.

Theorem 8.8. *Let $A_{safe} \subseteq \mathbb{F}_2^{nb}$ be a safe affine space with $|\tilde{Cl}(A_{safe})| \leq p$. Let T_{par} be a PDT with depth $\leq q$ and $T = CANONIZE(T_{par}, A_{safe})$. Let $g : \mathbb{F}_2^b \to \mathbb{F}_2$ be a gadget with $||\hat{g}||_\infty \leq n^{-2-\eta}$ for some constant $\eta > 0$. There exists an ordinary randomized decision tree*

T_{ord} of depth $\leq p+q$ such that the following two distributions have statistical distance $\leq b \cdot n^{-\eta/20}$ for all z:

- ***Distribution 1:*** *Sample $x \leftarrow G^{-1}(z) \cap A_{safe}$. Output* $\boxed{\textit{Transcript of } T \textit{ on } x}$
- ***Distribution 2:*** *Output $T_{ord}(z)$*

Proof. Throughout this proof, we denote by $A(t)$ the subspace after first t phases of T, i.e.

$$A(t) = A_{\text{safe}} \cap \{\text{intersection of queries made by } T \text{ in first } t \text{ phases}\}$$

.

The decision tree T_{ord} runs the simulation routine SIMULATE(T, A_{safe}, z) as described below.

SIMULATE(T, A_{safe}, z)

Input:

- Safe affine space $A_{\text{safe}} \subseteq \mathbb{F}_2^{nb}$
- A_{safe}-canonical PDT T
- Query access to a point $z \in \mathbb{F}_2^n$

Algorithm

- Define $\text{TR}_0 = \text{initial transcript}, v(0)$:
  - $A(0) = A_{\text{safe}}$
  - $S_0 = \phi$
  - $\beta(0) = \text{empty assignment}$
  - LIN-QUERY$(0) = \phi$
  - $v(0) = \text{root of } T$
- For $t = 1, 2, \cdots, d = \text{depth}(T)$,

$$\text{TR}_t = \text{UPDATE-SIMULATION}(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z)$$

- Return TR_d

Here UDPATE-SIMULATION$(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z)$ is a randomized algorithm which takes as input a transcript at phase $t-1$ and outputs a transcript at phase t. This is implemented as follows:

UPDATE-SIMULATION$(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z)$

Input:

- $A_{\text{safe}} \subseteq \mathbb{F}_2^{nb}$ safe affine space
- T: A_{safe}-canonical PDT
- TR_{t-1}: transcript at phase $t-1$
- Query access to a point $z \in \mathbb{F}_2^n$

Output:

- TR_t: transcript at phase t

Algorithm:

- **Case 0:** $v(t-1)$ is a leaf.
  - Do nothing. Return input transcript.
- **Case 1:** T makes a type 1 query at $v(t-1)$.
  - Update S_{t-1} to S_t. Sample the new closure assignment $\beta(t)$ as follows:
    - Query z_j for $j \in S_t \setminus S_{t-1}$
    - Pick an arbitrary w from the set $\{w | w_j = z_j \text{ for all } j \in S_t\}$ [a]
    - Sample $x \leftarrow G^{-1}(w) \cap A(t-1)$ uniformly at random [b] [c]
    - Set $\beta(t) = \mathrm{PROJ}(x, S_t)$
  - Update $v(t)$ according to $\beta(t)$
- **Case 2:** T makes a type 2 query ℓ at $v(t-1)$.
  - Sample a bit $b \in \mathbb{F}_2$ uniformly at random.
  - Update $\text{LIN-QUERY}(t) = \text{LIN-QUERY}(t-1) \cup \{(\ell, b)\}$
  - Update $v(t)$ according to b

[a] All bits in S_t have been queried at this point.
[b] Since $A(t-1)$ fixes all bits of S_{t-1}, this automatically ensures x is consistent with $\beta(t-1)$
[c] Since S_t is $A(t-1)$-compatible, the subspace $A(t-1)[\backslash S]$ is safe, so the set in question is non-empty by the equidistribution lemma.

Since $|\tilde{\mathrm{Cl}}(A_{\mathrm{safe}})| \leq p$, the size of the amortized closure can increase by at most 1 at each query and the closure is a subset of the amortized closure, the final size of the closure of the affine space reached by T_{par} (and therefore that of $T = \mathrm{CANONIZE}(T_{\mathrm{par}}, A_{\mathrm{safe}})$) [4] is at most $p + q$. Thus, T_{ord} makes at most $p + q$ queries.

It remains to show that the resulting distribution is close to $\mathrm{TRANSCRIPT}(T, G^{-1}(z)) \cap A_{\mathrm{safe}})$. To do so, we shall employ a hybrid argument.

We define a procedure for generating $\mathrm{TRANSCRIPT}(T, x)$ when x is sampled uniformly at random from $G^{-1}(z) \cap A_{\mathrm{safe}}$. We call this procedure $\mathrm{GENERATE}(T, A, z)$ (this procedure assumes full knowledge of z). We shall be comparing the output distributions of $\mathrm{GENERATE}(T, A_{\mathrm{safe}}, z)$ and $\mathrm{SIMULATE}(T, A_{\mathrm{safe}}, z)$.

$\mathrm{GENERATE}(T, A_{\mathrm{safe}}, z)$

Input:

- Safe affine space $A_{\mathrm{safe}} \subseteq \mathbb{F}_2^{nb}$
- A_{safe}-canonical PDT T
- A point $z \in \mathbb{F}_2^n$ (with full access)

[4] To clarify, the affine space reached by T_{par} is $A_{\mathrm{safe}} \cap \{\text{intersection of all queries made by } T_{\mathrm{par}}\}$.

Algorithm

- Define TR_0 = initial transcript :
  - $A(0) = A_{\mathrm{safe}}$
  - $S_0 = \phi$
  - $\beta(0)$ = empty assignment
  - LIN-QUERY$(0) = \phi$
  - $v(0)$ = root of T
- For $t = 1, 2, \cdots, d = \mathrm{depth}(T)$,

$$\mathrm{TR}_t = \text{UPDATE-ACTUAL}(T, \mathrm{TR}_{t-1}, A_{\mathrm{safe}}, t, z)$$

- Return TR_d

Here UPDATE-ACTUAL$(T, \mathrm{TR}_{t-1}, A_{\mathrm{safe}}, t, z)$ is the function which updates the transcript according to the appropriate conditional distrbution. This is implemented as follows:

UPDATE-ACTUAL$(T, \mathrm{TR}_{t-1}, A_{\mathrm{safe}}, t, z)$

Input:

- $A_{\mathrm{safe}} \subseteq \mathbb{F}_2^{nb}$ safe affine space
- T: A_{safe}-canonical PDT
- TR_{t-1}: a transcript at phase $t-1$
- A point $z \in \mathbb{F}_2^n$ (with full access)

Output:

- TR_t: a transcript at phase t

Algorithm:

- **Case 0:** $v(t-1)$ is a leaf.
  - Do nothing. Return input transcript.
- **Case 1:** T makes a type 1 query at $v(t-1)$.
  In TR_t, S_{t-1} gets replaced by S_t.
  - Replace S_{t-1} by S_t. Sample the new closure assignment $\beta(t)$ as follows:
    - Sample x uniformly at random from $G^{-1}(z) \cap A(t-1)$
    - Set $\beta(t) = \mathrm{PROJ}(x, S_t)$ for all $j \in S_t$ [a]
  - Update $v(t)$ according to $\beta(t)$
- **Case 2:** T makes a type 2 query ℓ at $v(t-1)$.
  - Let $\lambda = \Pr_{x \in A(t-1) \cap G^{-1}(z)}[\langle \ell, x \rangle = 0]$.
  - Let b be a random binary variable which is 0 with probability λ, 1 with probability $1 - \lambda$.

– LIN-QUERY(t) = LIN-QUERY$(t-1) \cup \{(\ell, b)\}$

– Update $v(t)$ according to b

[a]Since $A(t-1)$ fixes all bits in S_{t-1} to $\beta(t-1)$, this automatically ensures $\beta(t)$ is consistent with $\beta(t-1)$. Also, by equidistribution lemma, $A(t-1) \cap G^{-1}(z)$ is non-empty (although that is not required because if we are following a valid execution transcript, this set is guaranteed to be non-empty regardless of the gadget).

Our goal is to show the output distributions of SIMULATE(T, A_{safe}, z) and GENERATE(T, A_{safe}, z) are close in statistical distance. A transcript from GENERATE(T, A_{safe}, z) is generated by the following sampling procedure.

- Start with the initial transcript TR_0
- For $t = 1, 2, \cdots, d = \text{depth}(T)$,

$$\text{TR}_t \leftarrow \text{UPDATE-ACTUAL}(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z)$$

- Output TR_d.

A transcript from SIMULATE(T, A_{safe}, z) is generated by the following sampling procedure.

- Start with the initial transcript TR_0
- For $t = 1, 2, \cdots, d = \text{depth}(T)$,

$$\text{TR}_t \leftarrow \text{UPDATE-SIMULATION}(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z)$$

- Output TR_d.

We define a sequence of hybrids $\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_d$ ($d = \text{depth}(T)$) such that $\mathcal{H}_d = \text{GENERATE}(T, A_{\text{safe}}, z)$ and $\mathcal{H}_0 = \text{SIMULATE}(T, A_{\text{safe}}, z)$.

Hybrid $\mathcal{H}_i$

- Start with the initial transcript TR_0
- For $t = 1, 2, \cdots, d$,

$$\text{TR}_t \leftarrow \begin{cases} \text{UPDATE-ACTUAL}(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z) & \text{if } t < i \\ \text{UPDATE-SIMULATION}(T, \text{TR}_{t-1}, A_{\text{safe}}, t, z) & \text{if } t \geq i \end{cases}$$

- Output TR_d.

To show the statistical distance between the outputs of $\mathcal{H}_0, \mathcal{H}_d$ is small, we show that the statistical distance between the outputs of $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ is small for all i. More specifically, we shall show that $d_{TV}(\mathcal{H}_i, \mathcal{H}_{i+1}) \leq o(n^{-1-\eta/20}))$. Since $d \leq nb$, this implies that the total simulation error is at most $o(b \cdot n^{-\eta/20})$, as claimed.

Since $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ differ only in a single update step, it suffices to show that changing just one update step from UPDATE-ACTUAL to UPDATE-SIMULATION does not significantly affect the distribution (in statistical distance). Since at phases $i+1, i+2, \cdots$, $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$

get identically updated, it suffices to show the distributions of the new data added (in case of a type 1 query, the partial assignment $\beta(t)$; in case of a type 2 query, the bit b) are almost identically distributed by UPDATE-ACTUAL and UPDATE-SIMULATION.

We look at both possible types of queries and analyze the resulting distributions.

When T makes a type 1 query

Common data

- Sets S_{t-1} (closure at phase $t-1$) and S_t (new closure)
- $\beta(t-1)$: assignment to variables in blocks of S_{t-1}
- $A(t-1)$: affine space seen by T at phase $t-1$

Output

- $\beta(t)$: assignments to variables in blocks of S_t

Sampling procedure of UPDATE-ACTUAL

- Sample $x \in A(t-1) \cap G^{-1}(z)$ uniformly at random.
- Set $\beta(t) = \mathrm{PROJ}(x, S_t)$

Sampling procedure of UPDATE-SIMULATION

- Query z_j for $j \in S_t$
- Pick an arbitrary $w \in \mathbb{F}_2^n$ from the set $\{w | w_i = z_i \forall i \in S_t\}$.
- Sample $x \in A(t-1) \cap G^{-1}(w)$ uniformly at random.
- Set $\beta(t) = \mathrm{PROJ}(x, S_t)$.

Showing these distributions are close

We shall show that for all $w \in \mathbb{F}_2^n$ such that $w_i = z_i \ \forall i \in S_t$, the distributions $\nu_1 = \mathrm{PROJ}(G^{-1}(w) \cap A(t-1), S_t)$ and $\nu_2 = \mathrm{PROJ}(G^{-1}(z) \cap A(t-1), S_t)$ are $n^{-1-\eta/8}$ close to each other. This implies the result.

We shall show that for all $y \in \{0,1\}^{S_t \times [b]}$ such that y is consistent with $\beta(t-1)$ and $g(y(i)) = z_i = w_i \ \forall i \in S_t$, it holds that $\Pr_{\nu_1}[y] \in [1 \pm o(n^{-1-\eta/8})]\Pr_{\nu_2}[y]$. (It is easy to see all $y \in \mathrm{supp}(\nu_1) \cup \mathrm{supp}(\nu_2)$ satisfy these two conditions.)

We have

$$\Pr_{\nu_1}[y] = \frac{|G^{-1}(w) \cap A(t-1) \cap C_y|}{|G^{-1}(w) \cap A(t-1) \cap C_{\beta(t-1)}|}$$

and a similar expression for $\Pr_{\nu_2}[y]$. Thus,

$$\frac{\Pr_{\nu_1}[y]}{\Pr_{\nu_2}[y]} = \frac{|G^{-1}(w) \cap A(t-1) \cap C_y|}{|G^{-1}(z) \cap A(t-1) \cap C_y|} \times \frac{|G^{-1}(z) \cap A(t-1) \cap C_{\beta(t-1)}|}{|G^{-1}(w) \cap A(t-1) \cap C_{\beta(t-1)}|}$$

Since S_t is an $A(t-1)$-compatible set, we get that $A(t-1)[\setminus S_t]$ is a safe space. Applying Lemma 8.3 to $A(t-1)$ and $\beta(t)$, the first term lies in $[1 - n^{-1-\eta/6}, 1 + n^{-1-\eta/6}]$. Since

$\beta(t-1)$ fixes precisely the lifted variables in $\mathrm{Cl}(A(t-1))$, we can also apply Lemma 8.3 on $A(t-1)$ and $\beta(t-1)$, which gives us that the second term lies in $[1-n^{-1-\eta/6}, 1+n^{-1-\eta/6}]$. Thus, $\Pr_{\nu_1}[y]/\Pr_{\nu_2}[y] \in [1-n^{-1-\eta/8}, 1+n^{-1-\eta/8}]$. Finally, we have

$$\begin{aligned} d_{TV}(\nu_1,\nu_2) &= \frac{1}{2}\sum_y |\Pr_{\nu_1}(y) - \Pr_{\nu_2}(y)| \\ &\leq \frac{1}{2} \times n^{-1-\eta/8} \sum_y \Pr_{\nu_1}(y) \\ &< n^{-1-\eta/8} \end{aligned}$$

as desired.

When T makes a type 2 query

Common data

- S_{t-1} (closure at phase $t-1$)
- $\beta(t-1)$: assignment to variables in blocks of S_{t-1}
- $A(t-1)$: affine space seen by T at phase $t-1$
- ℓ: the new linear form queried

Output

- A bit $b \in \{0,1\}$

Sampling procedure of UPDATE-ACTUAL

- Let $\lambda = \Pr_{x \leftarrow G^{-1}(z) \cap A(t-1)}[\langle \ell, x \rangle = 0]$
- $$\text{Output } b = \begin{cases} 0 & \text{with probability } \lambda \\ 1 & \text{with probability } 1-\lambda \end{cases}$$

Sampling procedure of UPDATE-SIMULATION

- $$\text{Output } b = \begin{cases} 0 & \text{with probability } 1/2 \\ 1 & \text{with probability } 1/2 \end{cases}$$

Showing these distributions are close

To show that the distributions of b are close in both cases, we need to establish that $\lambda = \Pr_{x \leftarrow G^{-1}(z) \cap A(t-1)}[\langle \ell, x \rangle = 0]$ is close to $1/2$. Let $A' = A(t-1) \cap \{x | \langle \ell, x \rangle = 0\}$. Since ℓ is a type 2 query, $\mathrm{Cl}(A') = \mathrm{Cl}(A(t-1)) = S_{t-1}$. Our goal is to show

$$\Pr_{x \leftarrow G^{-1}(z)}[x \in A' | x \in A(t-1)] \in \left[\frac{1}{2} \pm o(n^{-1-\eta/6})\right]$$

Let $z = (z_1, z_2)$ where $z_1 \in \mathbb{F}_2^{S_t}, z_2 \in \mathbb{F}_2^{[n]\setminus S_t}$. Rewrite the above expression as

$$\Pr_{u \leftarrow G^{-1}(z_2)}[u \in (A')_{\beta(t-1)} | u \in A(t-1)|_{\beta(t-1)}]$$

Since $\mathrm{Cl}(A') = \mathrm{Cl}(A(t-1)) = S_{t-1}$, both the affine spaces $(A')_{\beta(t-1)}$ and $(A(t-1))_{\beta(t-1)}$ are safe. Let $k = \mathrm{codim}\big((A(t-1))_{\beta(t-1)}\big)$. Note that $\mathrm{codim}((A')_{\beta(t-1)}) = k+1$ [a]

By Lemma 5.6, we have $\Pr_{u \in G^{-1}(z_2)}[u \in A(t-1)_{\beta(t-1)}] \in [1 \pm n^{-1-\eta/4}]2^{-k}$ and $\Pr_{u \in G^{-1}(z_2)}[u \in (A')_{\beta(t-1)}] \in [1 \pm n^{-1-\eta/4}]2^{-k-1}$. Thus, we have

$$\Pr_{x \leftarrow G^{-1}(z)}[x \in A' | x \in A(t-1)] = \frac{\Pr_{u \in G^{-1}(z_2)}[u \in (A')_{\beta(t-1)}]}{\Pr_{u \in G^{-1}(z_2)}[u \in A(t-1)_{\beta(t-1)}]} \tag{8.2}$$

$$\in \left[\frac{1}{2} - n^{-1-\eta/6}, \frac{1}{2} + n^{-1-\eta/6}\right], \tag{8.3}$$

so the statistical distance between the two distributions is at most $O(n^{-1-\eta/6})$.

[a] We have $\mathrm{codim}((A')_{\beta(t-1)}) \in \{k, k+1\}$. If $\mathrm{codim}((A')_{\beta(t-1)})$ were k instead, then ℓ would have been fixed by $A(t-1)$ (because $A(t-1)$ fixes all bits of $\mathrm{Cl}(A(t-1))$). Therefore, in the conversion from the original PDT to its canonical version, this query would not have been made.

□

Now we prove the main result of this section: if Φ is $(p, p+q)$-DT-hard then (Φ, g) is (p, q)-PDT-hard whenever $g : \mathbb{F}_2^b \to \mathbb{F}_2$ is a gadget with $||\hat{g}||_\infty \le n^{-2-\eta}$ (for some constant $\eta > 0$) and $b = O(\log n)$.

Proof. (of Theorem 8.2) Let $P \subseteq \{0, 1, *\}^n$ be a set of partial assignments and let μ_ρ be a DT-hard distribution for each $\rho \in P$ with $|\rho| \le p$. This same family of distributions will establish PDT-hardness.

Assume this family is not PDT-hard. Then, the following objects exist:

- An affine space $A \subseteq \mathbb{F}_2^{nb}$ with $\tilde{\mathrm{Cl}}(A) \le p$
- A closure assignment $y \in \mathbb{F}_2^{\mathrm{Cl}(A) \times [b]}$ such that $\rho = G(y) \in P$. Let μ_ρ denote the hard distribution for ρ guaranteed to exist by DT-hardness.
- A PDT T with $\mathrm{depth}(T) \le q$ such that, as $x \leftarrow G^{-1}(\mu_\rho) \cap C_y \cap A$, with probability $\ge 2/3$, the following holds:
  - Let $A(q)$ be the affine space seen by T after q queries, i.e.,

$$A(q) = C_y \cap A \cap \{\text{intersection of all queries made}\}$$

  Then, $G(x)|_{\mathrm{Cl}(A(q)} \notin P$.

WLOG assume T only queries linear forms supported on blocks in $[n] \setminus \mathrm{Cl}(A)$ (since the closure assignment y is fixed). Note that A_y is a safe affine subspace. Our goal is to contradict the original assumption of $(p, p+q)$-DT-hardness - so we shall construct an ordinary decision tree T_{ord} such that starting from the cube C_ρ, when z is sampled from μ_ρ, within $p+q$ queries T_{ord} sees a partial assignment not in P.

Let $\mathcal{T}$ be the randomized decision tree which simulates CANONIZE(T, A_y) by making at most $p+q$ queries, guaranteed to exist by Theorem 8.8. Let $z = (z_1, z_2)$ where $z_1 \in \mathbb{F}_2^{\mathrm{Cl}(A)}$, $z_2 \in \mathbb{F}_2^{[n]\setminus \mathrm{Cl}(A)}$. By Theorem 8.8, the output distribution of $\mathcal{T}(z_2)$ is $o(1)$-close to the transcript of CANONIZE(T, A_y) when run on a uniformly random preimage of z_2 conditioned with A_y. Let β be the closure assignment at the last step of the transcript. When z is sampled according to μ, with probability $\geq 2/3 - o(1)$ (over the randomness of z and $\mathcal{T}$), the partial assignment $G(\beta, y)$ does not lie in P. Consider the randomized ordinary decision tree T_{rand} which simply runs $\mathcal{T}$ and outputs $G(\beta, y)$. With probability $\geq 1/2$, T_{rand} sees a partial assignment not in P after $p+q$ queries. By fixing the coins of T_{rand}, we get a deterministic ordinary decision tree T_{det} which sees a partial assignment not in P with probability $\geq 1/2$. This contradicts the assumption of $(p, p+q)$ hardness.

□

8.5 DT-hardness implies lower bounds for depth-restricted Res($\oplus$)

Combining Theorem 7.2 and Theorem 8.2, and replacing η by $\eta/2$ (for later convenience), we obtain the following:

Theorem 8.9. *Let Φ be a $(p, p+q)$-DT-hard CNF. Let $g : \mathbb{F}_2^b \to \mathbb{F}_2$ be a gadget with $b = O(\log n)$ and $||\hat{g}||_\infty \leq n^{-2-\eta/2}$ for some constant $\eta > 0$. Then, any Res($\oplus$) refutation of $\Phi \circ g$ of size s must have depth $\Omega\left(\frac{pq}{\log(s)}\right)$.*

9 Proving DT Hardness

The main theorem of this subsection is that Tseitin contradiction over an expander is $(\Omega(n), \Omega(n))$-DT hard, where n is the number of variables.

For convenience of the reader, we recall the definition of DT-hardness here.

Definition 8.1: Let Φ be a CNF on n variables, call a set of partial assignments $P \subseteq \{0, 1, *\}^n$ to be (p, q)-DT-hard wrt Φ if the following hold:

- **Non-emptiness:** $P \neq \phi$
- **No falsification:** No partial assignment $\rho \in P$ falsifies any clause of Φ.
- **Downward closure:** For any $\rho \in P$ and any $j \in [n]$, if $\tilde{\rho}$ is obtained by setting $\rho(j) \leftarrow *$, then $\tilde{\rho} \in P$
- **Hard for decision trees:** For any $\rho \in P$ which fixes at most p variables, there exists a distribution μ_ρ on assignments to unfixed variables such that the following holds:
  - Let T be a decision tree of depth q querying the unfixed variables. If we sample an assignment to the unfixed variables from μ_ρ and run T for q steps, the partial assignment seen by the tree also lies in P with probability $\geq 1/2$.

The CNF Φ is (p, q)-DT hard if it admits a set of (p, q)-DT-hard partial assignments.

Theorem 9.1. *Let Φ be the Tseitin contradiction over an $(m, d, \lambda < 0.95)$ expander (with m odd). Then, Φ is $(m/2000, m/2000)$-DT-hard – i.e., there exists a non-empty $(m/2000, m/2000)$-DT-hard set of partial assignments for Φ.*

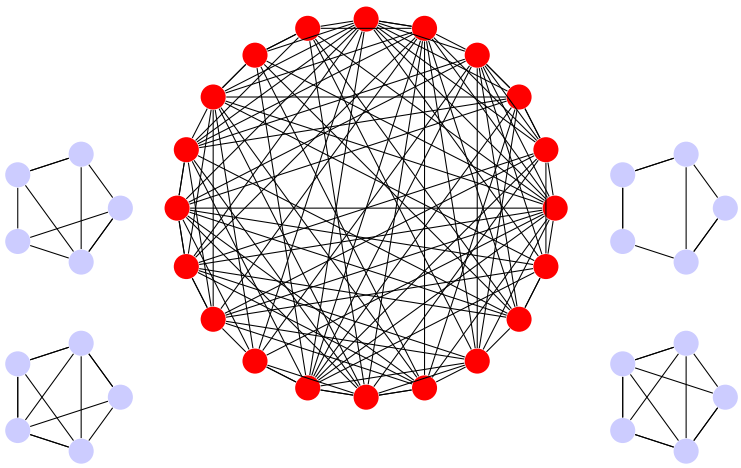
Remark 9.1. *We assume throughout that $d \geq 3$ as otherwise no expander can be constructed.*

9.1 Choosing the set of partial assignments

We define a partial assignment to the edges of our graph to be *valid* when it satisfies conditions given below. The set $P \subseteq \{0,1,*\}^n$ will be the set of valid partial assignments. Recall, $\text{fix}(\rho) \triangleq \{e | e \text{ has been fixed by } \rho\}$. For each $v \in V$, define $f_\rho(v) \triangleq 1 + \sum_{(v,w)\in \text{fix}(\rho)} \rho(v,w)$ (i.e., $f_\rho(v)$ denotes the parity of the unfixed edges incident to v in order to satisfy the original degree constraint for v).

Definition 9.2. Let $\rho \in \{0,1,*\}^E$ be a partial assignment. We say ρ is *valid* if it satisfies the following:

1. There exists exactly one connected component C in $(V, E \setminus \text{fix}(\rho))$ such that $\sum_{v\in C} f_\rho(v) \equiv 1 \pmod 2$. We call this component the *odd* component and every other component is called *even*.
2. The size of the odd connected component, $|C|$, is more than $m/2$.



Example of a $\rho \in P$: in $(V, E \setminus \text{fix}(\rho))$, exactly one component has $\sum f$ odd – and the size of that component is large

We now show that the set P of all valid partial assignments is indeed (p,q)-DT-hard. We begin by showing below that the first two properties for being (p,q)-DT-hard are satisfied.

Lemma 9.3. *The set of partial assignments P satisfies the conditions **Downward Closure** and **No falsification** for Φ (as defined in Definition 8.1).*

Proof. Both properties are straightforward to verify.

- **No falsification:** In order to falsify any clause, ρ has to fix all edges of some vertex. In that case, that vertex is an isolated connected component in $(V, E \setminus \text{fix}(\rho))$ and the total f_ρ in that component is 1 (mod 2). However, the first condition stipulates that there is exactly one connected component whose total f_ρ is odd, and that component has size more than $m/2$.
- **Downward closure:** Let $\rho \in P$, and let $\tilde{\rho}$ be obtained from ρ by setting $\rho(e) = *$ for some $e \in E$ that was fixed by ρ. There are three cases.

1. $e = (a, b)$ bridges the largest component C with some other component W. Wlog, $a \in C$ and $b \in W$. We have $\text{fix}(\tilde{\rho}) = \text{fix}(\rho) \setminus \{e\}$. Let the new expanded connected component be $C' = C \cup W$. Note that W forms a connected component in $(V, E \setminus \text{fix}(\rho))$ and therefore $\sum_{v \in W} f_\rho(v) = 0 \pmod 2$. For each $v \neq a, b$, $f_{\tilde{\rho}}(v) = f_\rho(v)$, and $f_{\tilde{\rho}}(a) = \rho(e) + f_\rho(a) \pmod 2$, $f_{\tilde{\rho}}(b) = \rho(e) + f_\rho(b) \pmod 2$. We have $|C'| \geq |C| \geq m/2$, so all we need to verify is that $\sum_{v \in C'} f_{\tilde{\rho}}(v) \equiv 1 \pmod 2$.

$$\sum_{v \in C \cup W} f_{\tilde{\rho}}(v) = \sum_{v \in C} f_\rho(v) + \rho(e) + \sum_{v \in W} f_\rho(v) + \rho(e) = 1 + 0 = 1 \pmod 2$$

2. $e = (a, b)$ bridges two components U and W, none of which is C. Very similar argument as above shows that

$$\sum_{v \in U \cup W} f_{\tilde{\rho}}(v) = \sum_{v \in U} f_\rho(v) + \rho(e) + \sum_{v \in W} f_\rho(v) + \rho(e) = 0 + 0 = 0 \pmod 2$$

   Thus, C remains the unique odd connected component.
3. $e = (a, b)$ does not bridge two different components. It is simple to verify in this case that the parity of all components remain unchanged.

□

Now we come to the final property: hardness for decision trees.
Before proceeding, we provide an informal proof overview. Readers interested in the actual proof can skip ahead to the end of this box.

Informal overview of proof of Theorem 9.9

We started with the CNF Φ being the Tseitin contradiction on an expander, with a parity constraint $\sum_{w \in N(v)} z(v, w) \equiv 1 \pmod 2$ for each $v \in V$. Given a partial assignment ρ, some of the variables get fixed. Now for each $v \in V$ we have a parity constraint over its unfixed incident edges:

$$\sum_{\substack{w \in N(v) \\ (w,v) \notin \text{fix}(\rho)}} z(v, w) \equiv f_\rho(v) \pmod 2$$

This is another Tseitin contradiction on the graph where the edges in $\text{fix}(\rho)$ are deleted (we call this graph G_ρ). For notational convenience, for a set of vertices $S \subseteq V$ call $f_\rho(S) = \sum_{v \in S} f_\rho(v)$. Let $C_1, C_2, \cdots, C_k$ be the connected components of G_ρ. The conditions for $\rho \in P$ are following: exactly one of $C_1, C_2, \cdots, C_k$ has $f_\rho(C_j) \equiv 1 \pmod 2$, and moreover, the size of that component is at least $m/2$.

Suppose C_1 is the unique odd component. There is no assignment which simultaneously satisfies all parity constraints of C_1 - so C_1 must contain a falsified clause. We have to design a hard distribution μ_ρ, Informally speaking, we must make it hard for a prover to locate a falsified clause.

It can be shown (Corollary 9.6) that there exists an assignment z_{sat} which simultaneously satisfies the parity constraint of every vertex that is not in C_1. Since our goal is to make locating a falsified clause harder, it makes sense to not have any falsified clause in $C_2, C_3, \cdots, C_k$ in the first place: we assign those edges according to z_{sat}. Now comes the interesting part: how do we define a distribution on the values of the edges in C_1?

Our procedure for this is simple: we pick a uniformly random vertex v. We shall call this vertex the *root*. One can show that there exists an assignment which satisfies all parity constraints other than v (Corollary 9.6). After picking v, we uniformly at random pick one such assignment. As we shall see soon, a decision tree that tries to home in on a partial assignment not in P is effectively trying to locate the root.

For simplicity let us assume the decision tree does not query any edges in $C_2, C_3, \cdots, C_k$ (there are no falsified clauses there, so the tree would just be wasting its budget). When it queries an edge in C_1, we delete that edge from the graph. Now, when the tree queries an edge e, the component C_1 might get split into two components: $C_{new}^{(0)}$ and $C_{new}^{(1)}$. Note that after getting the response to the query, f also gets updated (Recall that $f(v)$ is the RHS of the parity constraint at v. When a variable gets queried and determined, it goes to the RHS.) Out of $C_{new}^{(0)}$ and $C_{new}^{(1)}$, the component that contains the root will have total f odd, and the other one will have total f even. Reason: for the component that does not contain the root, there is an assignment which satisfies all its parity constraints, so its total f is even. The total f of the other component cannot be even, as otherwise there would be an assignment that satisfies the parity constraint at each vertex.

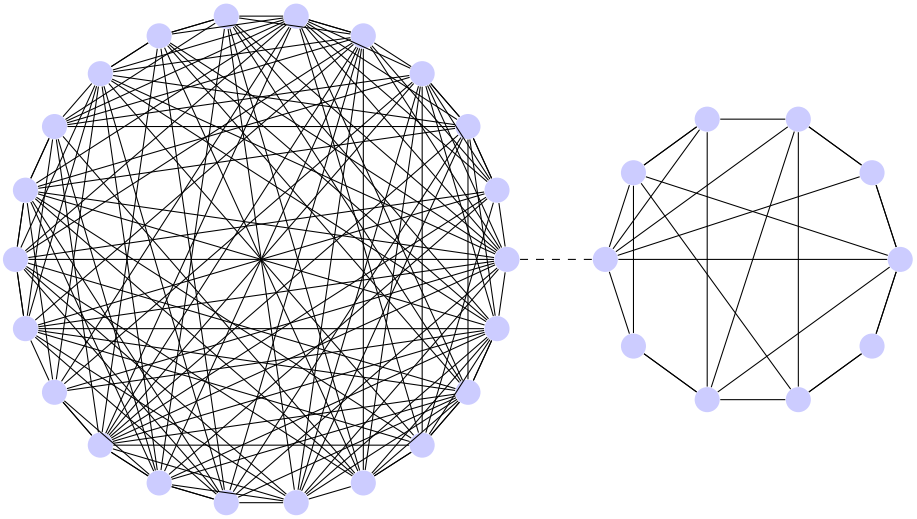
So at any point of execution of the decision tree, the graph (with all the queried edges deleted) will have exactly one component with f odd - and that will be the component containing the root. If the tree wants to see a partial assignment that does not lie in P, it needs to ensure the unique odd component has size $\leq m/2$. In other words, it needs to narrow down the possibility of the root to at most $m/2$ vertices.

Given this, our decision to pick the root uniformly at random looks wise - we want there to be as much uncertainty about its location as possible. But what about the situation when the tree has queried some edges, and therefore has some idea about the location of the root? We need to analyze exactly what information the tree has about the location of the root. For one, the tree knows that the root must lie in the current odd component. We shall show that this is all that the tree knows about the location of the root: when z is sampled from μ conditioned on [current information obtained by the tree], the conditional distribution of the root is uniform on the odd component (Lemma 9.8).

Now essentially the game looks as follows: at any point of time, the tree has seen some partial assignment - and it knows that the root lies in the unique odd component. As the game proceeds and more edges are queried (and then deleted), the unique odd component keeps shrinking. The goal of the decision tree is to ensure the size of the odd component goes below $m/2$.

To complete the proof, we need one last ingredient about expander graphs. Using the isoperimetric profile of the graph, one can show that if $\leq m/1000$ edges are deleted from G, one of its components must have large size: $\geq m\left(1 - \frac{1}{2d}\right)$. So if the decision tree is to put the root in a small component, it must actually put the root in a *very small* component.

Now consider a point of time when the large component shrinks. Before the bridge is queried, the conditional distribution of the root is uniform on this entire component.



Therefore, after the bridge is queried and it is revealed in which component the root lies, with overwhelmingly high probability the root is going to end up in the larger component. With high probability, this is going to happen every time. To formalize this intuition, we do an amortized induction where we count the amount by which the largest component has shrunk instead of total number of queries made.

First, we prove an easy but crucial lemma. This lemma is standard and has appeared in the literature before (for example in [Urq87]).

Lemma 9.4. *Let $G = (V, E)$ be a connected undirected graph and let $f : V \to \mathbb{F}_2$ a parity constraint for each vertex. Let $T \subseteq E$ be a spanning tree, and let $\tilde{z} \in \mathbb{F}_2^{E \setminus T}$ be an assignment to the edges not in T. Let $v \in V$ be a vertex. There exists a unique assignment $z \in \mathbb{F}_2^E$ such that z extends $\tilde{z}$ and $\sum_{w \in N(u)} z(u, w) = f(u)$ for all $u \neq v$.*

Proof. We construct z as follows: convert T to a rooted tree by making v the root and then process the vertices bottom up, starting at the leaves of T. When vertex u is being processed, all edges in the subtree of u have been assigned. Then, exactly one edge incident to u is kept unfixed (the edge $(u, \text{parent}[u])$ - assign it so that $\sum_{(u,w) \in E} z(u, w) = f(u)$ is satisfied.

It is also clear that this is the unique assignment to the edges in T which satisfies all these constraints and is consistent with the assignment to the edges of $E \setminus T$. This is because once the edges in the subtree of u has been fixed, there is a unique choice of $z(u, \text{parent}[u])$ that satisfies the parity constraint of u.

□

We call the procedure in the proof of Lemma 9.4 FIX$(G, T, v, \tilde{z}, f)$.

FIX$(G, T, v, \tilde{z}, f)$

Input:

- $G = (V, E)$: a connected graph
- T: a spanning tree of G
- v: a vertex in G

- $\tilde{z} \in \mathbb{F}_2^{E \setminus T}$: an assignment to the edges not in T
- $f : V \to \mathbb{F}_2$: a parity constraint for each vertex

---

Output:

- The unique assignment $z \in \mathbb{F}_2^E$ such that z extends $\tilde{z}$ and $\sum_{w \in N(u)} z(u,w) = f(u)$ for all $u \neq v$

Algorithm:

- Set $z(e) = \tilde{z}(e) \forall e \notin T$
- Root T at v. Let children(u) be the set of children of v.
- Execute RECURSIVE-FIX(v)
- Return z

Here RECURSIVE-FIX(u) is implemented as follows (with T, G, z, f being global variables and the vertex u being passed as a parameter):

RECURSIVE-FIX(u):

- For $w \in$ children(u):
  - Execute RECURSIVE-FIX(w).
  - Set $z(u,w) = f(w) + \sum_{u' \in N(w) \setminus \{u\}} z(u',w)$

We make the following observation now.

Observation 9.5. Let G be a connected graph, T a spanning tree of G, $v \in V$ any vertex, $\tilde{z} \in \mathbb{F}_2^{E(G) \setminus E(T)}$ an assignment to the non-tree edges, and $f : \mathbb{F}_2^V \to \mathbb{F}_2$ a parity constraint for each vertex.

- If $\sum_{u \in V} f(u) = 0 \pmod 2$ and $z = \text{FIX}(G, T, v, \tilde{z}, f)$, then $\sum_{w \in N(v)} z(v,w) = f(v) \pmod 2$
- If $\sum_{u \in V} f(u) = 1 \pmod 2$ and $z = \text{FIX}(G, T, v, \tilde{z}, f)$, then $\sum_{w \in N(v)} z(v,w) \neq f(v) \pmod 2$

Corollary 9.6. *Let G be a connected undirected graph and let $f : V \to \mathbb{F}_2$ be any map.*

1. *If $\sum_{v \in V} f(v) = 1 \pmod 2$, then for any $v \in V$ there exists an assignment $z \in \mathbb{F}_2^E$ such that $\sum_{w \in N(u)} z(u,w) = f(u) \pmod 2$ for each $u \neq v$, and $\sum_{w \in N(v)} z(v,w) \neq f(v) \pmod 2$.*
2. *If $\sum_{v \in V} f(v) = 0 \pmod 2$, there exists an assignment $z \in \mathbb{F}_2^E$ such that $\sum_{w \in N(u)} z(u,w) = f(u) \pmod 2$ is satisfied for all $u \in V$.*

Now we define the following hard distribution for each $\rho \in P$, when $|\rho| \leq \frac{m}{2000}$.

9.1.1 The hard distrbution

Our goal in this subsection is to define for each $\rho \in P$ a distribution $\mu = \mu_\rho$ on the unfixed variables so that the requirement in Definition 8.1 for DT-Hardness is satisfied.

Definition 9.7. Let $\rho \in P$ be a valid partial assignment which fixes at most $\frac{m}{1000}$ edges. Define the following:

1. Define f_ρ as before (i.e. for each $v \in V$, $f_\rho(v) = 1 + \sum_{(u,v)\in \mathrm{fix}(\rho)} \rho(u,v)$. After the edges in $\mathrm{fix}(\rho)$ are fixed according to ρ, $f_\rho(v)$ is the RHS of the modified parity constraint of v.
2. Let C_ρ be the unique connected component in $G_\rho = (V, E \setminus \mathrm{fix}(\rho))$ whose total f_ρ is odd.

Notation: For a subset $A \subseteq V$ we denote by $E(A)$ the set of edges in G both of whose endpoints lie in A.

Now we describe the procedure of sampling from μ. Note that the values of edges in $\mathrm{fix}(\rho)$ are fixed; we have to define a distribution on variables in $\mathrm{free}(\rho)$. We do this as follows.

DTFooling
Input:

- Graph $G = (V, E)$
- A valid partial assignment $\rho \in \{0, 1, *\}^E$, $\rho \in P$

Output: A sample $z \in \{0,1\}^E$ from μ_ρ

Sampling procedure

- **Assigning the edges of C_ρ**: Fix an arbitrary spanning tree T of C_ρ formed by edges from $\mathrm{free}(\rho)$. Uniformly at random pick a vertex $v \in C_\rho$. Let $\tilde{w} \in \mathbb{F}_2^{E(C_\rho)\setminus(\mathrm{fix}(\rho)\cup E(T))}$ be a uniformly random assignment to the unfixed edges in C_ρ not in T. Let $\tilde{z} \in \mathbb{F}_2^{E(C_\rho)\setminus E(T)}$ be the following assignment to the non-tree edges in C_ρ:

$$\tilde{z}(e) = \begin{cases} \rho(e) & \text{if } e \in E(C_\rho) \cap \mathrm{fix}(\rho) \\ \tilde{w}(e) & \text{if } e \in (E(C_\rho) \cap \mathrm{free}(\rho)) \setminus E(T) \end{cases}$$

  Assign the edges in C_ρ according to $\mathrm{FIX}(C_\rho, T, v, \tilde{z}, f_\rho)$.

- **Assigning the edges for every other component C':** Pick an arbitrary spanning tree T' of C' formed by edges from $\mathrm{free}(\rho)$. Let$\tilde{w}' \in \mathbb{F}_2^{E(C')\setminus\{\mathrm{fix}(\rho)\cup E(T')\}}$ be a uniformly random assignment to the free non-tree edges. Let $\tilde{z}' \in \mathbb{F}_2^{E(C')\setminus E(T')}$ be the following assignment to the non-tree edges in C':

$$\tilde{z}'(e) = \begin{cases} \rho(e) & \text{if } e \in E(C') \cap \mathrm{fix}(\rho) \\ \tilde{w}'(e) & \text{if } e \in (E(C') \cap \mathrm{free}(\rho)) \setminus E(T') \end{cases}$$

  Pick an arbitrary vertex v'. Assign the edges of C' according to $\mathrm{FIX}(C', T', v', \tilde{z}', f_\rho)$.

Let $A_{\rho,v}$ denote the set of all $z \in \mathbb{F}_2^E$ that are consistent with ρ and satisfy the following:

1. For all $u \neq v$, $\sum_{w\in N(u)} z(u,w) = f_\rho(u)$.

2. $\sum_{w \in N(v)} z(v,w) = 1 + f_\rho(v)$.

We make the following remark now.

Remark 9.2. *Let ρ be any valid (partial) assignment to edges of G. Then,*

1. *$A_{\rho,v}$ is an affine space in $\mathbb{F}_2^E$, for each $v \in C_\rho$.*
2. *DTFooling picks a random $v \in C_\rho$ and then samples a random point in $A_{\rho,v}$.*

For any $z \in \text{supp}(\mu)$, the parity constraint is violated for exactly one vertex (the vertex which was chosen as the root of the spanning tree of the odd connected component). Call this vertex $root(z)$.

Before proving the hardness, we note down some properties of the distribution.

9.1.2 Conditional Distribution of the Root is Uniform

We prove a useful property of the distribution sampled by DTFooling, given a valid partial assignment ρ. The idea is when a decision tree queries bits from an assignment z to the edges sampled according to μ_ρ, the graph G_ρ starts splitting into further smaller components. The decision tree knows at every instant in which component root(z) lies,as there is always a unique odd component. The lemma below ensures that conditioned on what the decision tree has observed so far, the distribution of root(z) remains uniform over all vertices in the odd component.

In the following let free$(\rho) \subseteq E$ be the set of edges free in ρ. For convenience of the reader, we re-state the definition of f_ρ in Definition 9.2 here.

- $f_\rho \in \mathbb{F}_2^V, f_\rho(v) = 1 + \sum_{(v,w) \in \text{fix}(\rho)} \rho(v,w)$

Lemma 9.8. *Let ρ be a valid partial assignment and $\mu = \mu_\rho$ be the distribution in Definition 9.7. Let $S \subseteq$ free(ρ) be a subset of free edges and let $\alpha \in \mathbb{F}_2^S$ be an assignment to S. Define $f_\alpha(v) = f_\rho(v) + \sum_{(v,w) \in S} \alpha(v,w)$. Suppose the connected components of $(V,$ free$(\rho) \setminus S)$ are $C_1, C_2, \cdots, C_k$, where $\sum_{v \in C_1} f_\alpha(v) = 1$, and for all $j \neq 1$, $\sum_{v \in C_j} f_\alpha(v) = 0$. Then, the following is true:*

- *The distribution of root(z) as z is sampled from $\mu = \mu_\rho$ conditioned on $z|_S = \alpha$, is uniform on C_1.* [5]

This lemma is essentially saying the following: suppose the input is sampled from μ and currently, the tree has seen a partial assignment. After fixing the edges seen by the tree, the graph splits into multiple components and there is a unique odd component C_1. Then, when z is sampled from μ conditioned on being consistent with current information obtained, the distribution of root(z) is uniform on C_1.

Proof. Conditioned on $z|_S = \alpha$, the root cannot lie in any of $C_2, C_3, \cdots, C_k$. (Reason: after we choose the root, only the parity constraint at the root is violated; other parity constraints are satisfied. But after fixing S to α, it is not possible to satisfy all parity constraints of C_1 simultaneously as the sum of the modified parity constraints of C_1 is odd.)

[5] In order for the statement to be valid, we need that some assignment in supp(μ) is consistent with α (otherwise we are conditioning on a probability 0 event). This follows from Corollary 9.6 applied to each component of $(G,$ free$(\rho) \setminus S)$ separately.

So we need to show that for all $u \in C_1$, $\Pr_\mu[\text{root}(z) = u | z_S = \alpha]$ is a non-zero quantity independent of u. Note that $C_1 \subseteq C_\rho$ (recall that C_ρ is the unique odd component of G_ρ). Since for all $u \in C_1$, $\Pr_\mu[\text{root}(z) = u] = \frac{1}{|C_\rho|}$ is a non-zero quantity independent of u, by Bayes' rule it suffices to show that for all $u \in C_1$, $\Pr_\mu[z_S = \alpha | \text{root}(z) = u]$ is a non-zero quantity independent of u.

Let $M \in \mathbb{F}_2^{V \times E}$ be the edge-vertex incidence matrix of G. Let $\gamma_v \in \mathbb{F}_2^V$ be the following vector:

$$\gamma_v(u) = \begin{cases} f_\rho(u) & \text{if } u \neq v \\ 1 + f_\rho(u) & \text{otherwise} \end{cases}$$

Once v is chosen as the root, the sampling procedure samples a uniformly random element of the affine space $\{z | Mz = \gamma_v\}$.

Let $S = \{r_1, r_2, \cdots, r_{|S|}\}$. Let $N \in F^{S \times E}$ be the matrix whose j-th row is the standard basis vector at coordinate r_j. Once u is chosen as the root, $z|_S = \alpha$ if and only if z satisfies the following equation:

$$\begin{bmatrix} M \\ \hline N \end{bmatrix} z = \begin{bmatrix} \gamma_u \\ \hline \alpha \end{bmatrix}$$

Let

$$J = \begin{bmatrix} M \\ \hline N \end{bmatrix}$$

Conditioned on satisfying $Mz = \gamma_u$, the probability of satisfying $z|_S = \alpha$ is either $2^{\text{rank}(M) - \text{rank}(J)}$ (if there is a solution) or 0 (if there is some inconsistency in the right hand sides of the system of equations). (Indeed, for $v \notin C_1$ there is an inconsistency in the right hand sides - as noted above.)

Now for all $v \in C_1$, we shall show there is a z such that $\text{root}(z) = v$ and $z|_S = \alpha$ – this will show that for $v \in C_1$, $\Pr[z|_S = \alpha | \text{root}(z) = v]$ is a non-zero quantity independent of v. To show this, we construct an assignment as follows:

- For each C_j, choose a spanning tree Q_j disjoint from $S \cup \text{free}(\rho)$.
- Let $\tilde{z}$ be any assignment to the edges not in $Q_1, Q_2, \cdots, Q_k$ such that $\tilde{z}$ agrees with α on S.
- Assign the values of the edges in C_1 according to $\text{FIX}(C_1, Q_1, v, \tilde{z}, f_\alpha)$
- For $j > 1$, pick an arbitrary vertex $v_j \in C_j$ and assign the edges of Q_j according to $\text{FIX}(C_j, T_j, v_j, \tilde{z}, f_\alpha)$

This assignment satisfies $z|_S = \alpha$ by construction; moreover, it also satisfies $Mz = \gamma_v$ because of Lemma 9.4 and Observation 9.5. This completes the proof. □

9.1.3 Proving Hardness

Now we prove $(p, q)-$hardness for ordinary decision trees.

Convention: Henceforth, given a partial assignment ρ and a decision tree T making queries to the variables in $\text{free}(\rho)$, we define *the partial assignment seen by the tree* to be the partial assignment formed by fixing the edges queried by the tree and ρ.

Theorem 9.9. *Let $G = (V, E)$ be an $(m, d, \lambda < 0.95)$-spectral expander with $|V| = m$ being odd. Let P be the set of partial assignments defined as in Definition 9.2. Let $\rho \in P$ be a valid partial assignment with $|\rho| = p \leq \frac{m}{2000}$. Let μ be the distribution defined as in Definition 9.7* [6]*. Let T be any decision tree making at most $q \leq \frac{m}{2000}$ queries. Sample $z \leftarrow \mu$. Then, with probability $\geq 1/2$, the partial assignment seen by the tree after $q = m/2000$ queries also lies in P.*

Notice the slightly unconventional notation here: m denotes the number of vertices in the graph, and n denotes the number of edges (i.e., the number of variables in the Tseitin contradiction), ($n = md/2 = \Theta(n)$).

Proof. We fix some notation that will be used in the rest of the proof.

1. At time-step j, the partial assignment seen by the tree is ρ_j (this includes the edges fixed by ρ and the edges queried by T).
2. $E_j \triangleq \text{fix}(\rho_j)$
3. $G_j \triangleq (V, E \setminus E_j)$
4. Define $f_j(v) = 1 + \sum_{(v,w)\in E_j} \rho_j(v, w)$.
5. Let C_j be the unique connected component of G_j such that $\sum_{v\in C_j} f_j(v) \equiv 1 \pmod 2$

Before proceeding, we make some remarks.

Remark 9.3. *After some vertex v is chosen to be the root, it is guaranteed that the parity constraint of all vertices other than v is satisfied. It is also known that all parity constraints are not satisfiable simultaneously (since sum of the right hand sides is odd). So, after we know the value of some edges (say given by the partial assignment σ), after removing those edges, exactly one connected component has odd $\sum f_\sigma$ - and the root lies in this component. This means that after z is sampled according to μ, condition (i) defining membership in P is always satisfied. Only condition (ii) (which stipulates that the odd component must have large size) can possibly be violated*

Remark 9.4. *Suppose after querying an edge, in G_{j+1} the component C_j splits into $C_j = A \cup B$. Initially $\sum_{v\in C_j} f_j(v)$ is odd. After querying the edge, exactly one of $\sum_{v\in A} f_{j+1}(v)$ and $\sum_{v\in B} f_{j+1}(v)$ is odd - and the root must lie in the component where the sum is odd.*

Once the value of the edge is revealed, the decision tree knows which one of A, B contains the root. Thus, the decision tree has made some progress in determining the location of the root.

We want to say that the decision tree can never make too much progress - our tool here is Lemma 9.8, which says that the decision tree does not know anything about the root other than the fact that its conditional distribution is uniform on the current odd component.

We start with a crucial lemma.

[6] μ is a distribution on $\mathbb{F}_2^n$ such that every $z \in \text{supp}(\mu)$ is consistent with ρ

Lemma 9.10. *At any time-step j, the largest connected component of G_j must have size $\geq m\left(1-\frac{1}{2d}\right)$*

Proof. Suppose not; let time-step j be a time step where all the connected components of G_j have size $< m\left(1-\frac{1}{2d}\right)$. We then greedily pick a subset of the connected components whose union T has size in the interval $\left[\frac{m}{4d}, m-\frac{m}{4d}\right]$. Cheeger's inequality (Lemma 6.2) then implies the cut $E(T, V\setminus T)$ has at least $\frac{m}{200}$ edges.

This means the current partial assignment fixes at least $m/200$ edges. However, the current partial assignment can only fix $p+q \leq m/1000$ edges. □

Now, every time the decision tree queries an edge, we make it pay us some coins as follows. Suppose the current partial assignment lies in P; the current graph is G_j and the current odd component is C_j, and the tree queries the edge e.

- If removing e keeps C_j connected, the tree does not have to pay anything.
- Suppose removing e splits C_j into two components: $C_j = A \cup B$. The value of e is revealed - and it determines in which component of A, B the root belongs to. Suppose the root lies in A. If $|A| \leq m/2$, the decision tree does not pay anything and wins the game. Otherwise, the decision tree has to pay $|B|$ coins.

(In other words: if, at any point of time, the largest component in G_j isn't the odd component, the decision tree wins the game. Otherwise, if the decision tree shrinks the size of the largest component by s, it must pay s coins.)

By Lemma 9.10, the decision tree only pays $\leq \frac{m}{2d}$ coins. So we start by awarding the decision tree a budget of $b = \frac{m}{2d}$ coins, and argue (by induction on number of coins remaining) that the decision tree loses the game with high probability. (The decision tree loses the game when it has to pay some coins but it is broke.)

At this point, we allow the decision tree to make as many as queries as it wants – as long as it maintains that the largest component has size $\geq m\left(1-\frac{1}{2d}\right)$ (and therefore it does not use more than b coins). We prove the following statement by inducting on number of coins remaining.

Lemma 9.11. *Suppose the decision tree has c coins remaining and has not won the game yet. Then, the probability it wins the game is $\leq \frac{6c}{5m} + \frac{1}{5}$.*

Proof. We induct on c. Consider the base case $c = 0$: the decision tree has no coins remaining. Let the current odd component be C. The first time the tree splits C, the root must lie in the smaller component for the decision tree to win. Suppose the tree queries an edge e and C splits into $C = A \cup B$ where $|A| \geq m\left(1-\frac{1}{2d}\right)$. Before querying e, the conditional distribution of the root was uniform on C by Lemma 9.8. Conditioned on the partial assignment revealed before querying e, the probability the root lies in B is $|B|/|C|$. The probability the tree wins the game is thus

$$\frac{|B|}{|C|} \leq \frac{\frac{m}{2d}}{m\left(1-\frac{1}{2d}\right)} \leq \frac{1}{5},$$

so the base case is true.
Now we handle the inductive step. Suppose the tree has c coins. Suppose the current odd component is C_j, with $|C_j| \geq m\left(1 - \frac{1}{2d}\right)$. Suppose the decision tree queries e and removing e splits C_j into $C_j = A \cup B$, where A is the larger component $\left(|A| \geq m\left(1 - \frac{1}{2d}\right)\right)$. The tree wins the game at this stage if the root lies in B, otherwise it pays $|B|$ coins and proceeds to the next stage. Before querying e the distribution of the root is uniform on C, so the probability it lies in B is $\frac{|B|}{|A|+|B|} \leq \frac{6|B|}{5m}$. If the root does not lie in B, the decision tree has $c - |B|$ coins remaining, so then it can win the came with probability at most $\frac{6(c-|B|)}{5m} + \frac{1}{5}$ by the inductive hypothesis. By union bound, the probability the tree wins the game is at most

$$\frac{6|B|}{5m} + \frac{6(c-|B|)}{5m} + \frac{1}{5} = \frac{6c}{5m} + \frac{1}{5}$$

□

Since the decision tree starts off with $\frac{m}{2d}$ coins, it can win with probability at most $\frac{6}{10d} + \frac{1}{5}$. Since $d > 2$, with probability $\geq \frac{3}{5}$, the partial assignment seen by the decision tree after q queries lies in P. □

Now we have all the ingredients require to prove $(m/2000, m/2000)$-DT-hardness of Φ.

Proof. (of Theorem 9.1) Choose the set of partial assignments P as defined in Section 9.1. We have already established that this set P satisfies all requirements in Definition 8.1 defining DT-hardness.

- **No falsification** and **Downward Closure**: Established in Lemma 9.3.
- **Hardness against decision trees:** Established in Theorem 9.9.

This completes the proof. □

pr

10 Putting everything together

With Theorem 8.9 and Theorem 9.1 in hand, we are now in a position to prove our main result, Theorem 1.1.

Proof. (of Theorem 1.1) Let Φ be the Tseitin contradiction on G. Recall that m is the number of vertices in G and n is the number of variables in the unlifted Tseitin contradiction (i.e., $n = |E|$). Theorem 9.1 shows that that Tseitin contradiction over such an expander is $(m/2000, m/2000)$-DT-hard. Applying Theorem 8.9 with $p = q = m/4000$ and denoting by IP the Inner Product gadget on $(4+\eta)\log(n)$ bits, we get the following result:

- Any size s refutation of $\Phi \circ \mathrm{IP}$ must require depth $\Omega\left(\frac{n^2}{\log(s)}\right)$

Note that the number of variables in $\Phi \circ \mathrm{IP}$ is $N = O(n\log(n))$. We can also interpret the result as follows:

- Any depth $N^{2-\epsilon}$ Res($\oplus$) refutation of $\Phi \circ \mathrm{IP}$ requires size $\exp(\tilde{\Omega}(N^{\epsilon}))$

This is what we wanted to show. □

References

[AI25] Yaroslav Alekseev and Dmitry Itsykson. "Lifting to Bounded-Depth and Regular Resolutions over Parities via Games". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*. Ed. by Michal Koucký and Nikhil Bansal. ACM, 2025, pp. 584–595. DOI: 10.1145/3717823.3718150.

[Ale+04] Michael Alekhnovich et al. "Pseudorandom Generators in Propositional Proof Complexity". In: *SIAM J. Comput.* 34.1 (2004), pp. 67–88. DOI: 10.1137/S0097539701389944.

[BCD24] Sreejata Kishor Bhattacharya, Arkadev Chattopadhyay, and Pavel Dvorák. "Exponential Separation Between Powers of Regular and General Resolution over Parities". In: *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*. Ed. by Rahul Santhanam. Vol. 300. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, 23:1–23:32. DOI: 10.4230/LIPICS.CCC.2024.23.

[Bea+92] Paul Beame et al. "Exponential Lower Bounds for the Pigeonhole Principle". In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*. Ed. by S. Rao Kosaraju et al. ACM, 1992, pp. 200–220.

[BI25a] Farzan Byramji and Russell Impagliazzo. "Lower bounds for the Bit Pigeonhole Principle in Bounded-Depth Resolution over Parities". In: *Electron. Colloquium Comput. Complex.* TR25-118 (2025).

[BI25b] Farzan Byramji and Russell Impagliazzo). "Lower bounds for the Bit Pigeonhole Principle in Bounded-Depth Resolution over Parities (Revision 1)". In: *Electron. Colloquium Comput. Complex.* TR25-118 (2025).

[BK23] Paul Beame and Sajin Koroth. "On Disperser/Lifting Properties of the Index and Inner-Product Functions". In: *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*. Ed. by Yael Tauman Kalai. Vol. 251. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 14:1–14:17. DOI: 10.4230/LIPICS.ITCS.2023.14.

[BW01] Eli Ben-Sasson and Avi Wigderson. "Short proofs are narrow - resolution made simple". In: *J. ACM* 48.2 (2001), pp. 149–169. DOI: 10.1145/375827.375835.

[CD25] Arkadev Chattopadhyay and Pavel Dvorák. "Super-Critical Trade-Offs in Resolution over Parities via Lifting". In: *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*. Ed. by Srikanth Srinivasan. Vol. 339. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 24:1–24:19. DOI: 10.4230/LIPICS.CCC.2025.24.

[Cha+17] Arkadev Chattopadhyay et al. "Lower Bounds for Elimination via Weak Regularity". In: *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*. Ed. by Heribert Vollmer and Brigitte Vallée. Vol. 66. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 21:1–21:14. DOI: 10.4230/LIPICS.STACS.2017.21.

[Cha+21] Arkadev Chattopadhyay et al. "Query-to-Communication Lifting Using Low-Discrepancy Gadgets". In: *SIAM J. Comput.* 50.1 (2021), pp. 171–210. DOI: 10.1137/19M1310153.

[Cha+23] Arkadev Chattopadhyay et al. "Lifting to Parity Decision Trees via Stifling". In: *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*. Ed. by Yael Tauman Kalai. Vol. 251. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 33:1–33:20. DOI: 10.4230/LIPICS.ITCS.2023.33.

[Che71] Jeff Cheeger. “A Lower Bound for the Smallest Eigenvalue of the Laplacian”. In: *Problems in Analysis*. Ed. by Robert C. Gunning. Princeton: Princeton University Press, 1971, pp. 195–200. ISBN: 9781400869312. DOI: doi:10.1515/9781400869312-013.

[EGI24] Klim Efremenko, Michal Garlıék, and Dmitry Itsykson. “Lower Bounds for Regular Resolution over Parities”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O’Donnell. ACM, 2024, pp. 640–651. DOI: 10.1145/3618260.3649652.

[EI25] Klim Efremenko and Dmitry Itsykson. “Amortized Closure and Its Applications in Lifting for Resolution over Parities”. In: *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*. Ed. by Srikanth Srinivasan. Vol. 339. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 8:1–8:24. DOI: 10.4230/LIPICS.CCC.2025.8.

[Göö+16] Mika Göös et al. “Rectangles Are Nonnegative Juntas”. In: *SIAM J. Comput.* 45.5 (2016), pp. 1835–1869. DOI: 10.1137/15M103145X.

[GPT22] Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. “Linear Branching Programs and Directional Affine Extractors”. In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 4:1–4:16. DOI: 10.4230/LIPICS.CCC.2022.4.

[GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. “Query-to-Communication Lifting for BPP”. In: *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. Ed. by Chris Umans. IEEE Computer Society, 2017, pp. 132–143. DOI: 10.1109/FOCS.2017.21.

[Hak85] Armin Haken. “The Intractability of Resolution”. In: *Theor. Comput. Sci.* 39 (1985), pp. 297–308. DOI: 10.1016/0304-3975(85)90144-6.

[IS14] Dmitry Itsykson and Dmitry Sokolov. “Lower Bounds for Splittings by Linear Combinations”. In: *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*. Ed. by Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik. Vol. 8635. Lecture Notes in Computer Science. Springer, 2014, pp. 372–383. DOI: 10.1007/978-3-662-44465-8_32.

[LPS88] A Lubotzky, R Phillips, and P Sarnak. “Ramanujan graphs”. In: *Combinatorica* 8.3 (Sept. 1988), pp. 261–277.

[Mar73] G.A. Margulis. “Explicit constructions of expanders.” In: *Problemy Pereda ˇci Informacii, 9(4):71–80,* (1973).

[MP97] Alexis Maciel and Toniann Pitassi. “On $\mathrm{ACC}^0[p^k]$ Frege Proofs”. In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*. Ed. by Frank Thomson Leighton and Peter W. Shor. ACM, 1997, pp. 720–729. DOI: 10.1145/258533.258669.

[Urq87] Alasdair Urquhart. “Hard examples for resolution”. In: *J. ACM* 34.1 (1987), pp. 209–219. DOI: 10.1145/7531.8928.

Appendices

A Facts about Amortized Closure

Proof. **(of Lemma 2.25)** $\mathrm{Cl}(V) = \mathrm{Cl}(W)$ follows from the definition of closure.

We shall show $\tilde{\mathrm{Cl}}(V) = \tilde{\mathrm{Cl}}(W)$. Since amortized closure of a set of vectors depends only on its span, we can choose a different basis for V. Define $V_{in} = \{v \in V | \mathrm{supp}(v) \subseteq \mathrm{Cl}(V)\}$. Let $B_{in} = \{a_1, a_2, \cdots, a_k\}$ be a basis for V_{in}. Complete this to a basis for V: $B = B_{in} \cup B_{out}$. Let $C = B \cup \{e_{j,k} | j \in \mathrm{Cl}(V), k \in [b]\}$. We have $\mathrm{span}(B) = \mathrm{span}(V)$ and $\mathrm{span}(C) = \mathrm{span}(W)$. So, it suffices to show that $\tilde{\mathrm{Cl}}(B) = \tilde{\mathrm{Cl}}(C)$.

Let $M_{in} \in \mathbb{F}_2^{k \times nb}$ be the matrix where the rows of B_{in} are stacked.

We shall use the following claim (proof is deferred after this proof).

Claim A.1. There exists a index $\text{col-closure}(i) \in [b]$ for each $i \in \mathrm{Cl}(B)$ such that the columns $(i, \text{col-closure}(i))$ of M_{in} are linearly independent.

We shall to show that any set of blocks which is acceptable for C is also acceptable for B - this implies $\tilde{\mathrm{Cl}}(B) = \tilde{\mathrm{Cl}}(C)$. Note that the matrix of B looks like the following.

$$\begin{bmatrix} 0 & M_{in} & 0 \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

where the entries marked $*$ are arbitrary. The matrix of C looks like the following:

$$\begin{bmatrix} 0 & M_{in} & 0 \\ * & * & * \\ * & * & * \\ * & * & * \\ 0 & I & 0 \end{bmatrix}$$

Let $S \subseteq [n]$ be a C-acceptable set of blocks. For each $i \in S$, there is an index $\mathrm{col}(i)$ such that the corresponding columns in C are linearly independent. We need to find another set of indices $\text{col-new}(i)$ for each $i \in S$ such that the corresponding columns in B are linearly independent. We choose these indices as follows:

$$\text{col-new}(i) = \begin{cases} \mathrm{col}(i) & \text{if } i \notin \mathrm{Cl}(B) \\ \text{col-closure}(i) & \text{if } i \in \mathrm{Cl}(B) \end{cases}$$

It is easy to see this choice works. Suppose some linear combination of these columns were 0. This linear combination cannot include any $i \in \mathrm{Cl}(B)$ - since the corresponding columns in M_{in} are linearly independent. Thus, this linear combination only includes columns not in $\mathrm{Cl}(B)$ - however, for any $i \in \mathrm{Cl}(B)$, the columns in B and C are identical (upto a fixed number of trailing 0's). □

It remains to prove Claim A.1.

Proof. (of Claim A.1).
We inductively prove the following statement.

Let V be a set of vectors with $\mathrm{Cl}(V) = S$. Let $V_{in} = \mathrm{span}\{v \in V, \mathrm{supp}(v) \in V\}$. There exists a choice of index col-clos(i) for each $i \in \mathrm{Cl}(V)$ such that the following statement is true.

Let $T = \{v_1, v_2, \cdots, v_k\} \in V_{in}$ be a set of vectors that span V_{in}. Let M_T be the matrix where the rows of T are stacked on top of one other:

$$M_T = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ \cdots \\ v_k \end{bmatrix}$$

Then, the columns $(i, \mathrm{col}(i))$ in M_T are linearly independent.

Note that this statement is true for one set of vectors that span V_{in} if and only if it is true for all sets of vectors that span V_{in} (the statement is saying $\tilde{\mathrm{Cl}}(V_{in}) = \mathrm{Cl}(V)$, and amortized closure and closure depend only on linear span, Fact 2.14).

We prove this statement by induction. The base case when $\mathrm{Cl}(V) = \phi$ is trivial. Consider the case when we a new vector u to V and the closure changes from S_{old} to S_{new}. Let $V_{extra} = \mathrm{span}\{v \in V \setminus V_{in}, \mathrm{supp}(v) \subseteq S_{new}\}$.Let $w_1, w_2, \cdots, w_l$ be a set of vectors in $V \cup \{u\}$ which span V_{extra}. We have $\mathrm{span}\{v \in V \cup \{u\} | \mathrm{supp}(v) \subseteq S_{new}\} = \mathrm{span}(v_1, v_2, \cdots, v_k, w_1, w_2, \cdots, w_l)$.

Let

$$M_{old} = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ \cdots \\ v_k \end{bmatrix}$$

$$M_{extra} = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ \cdots \\ w_l \end{bmatrix}$$

$$M_{new} = \begin{bmatrix} M_{old} & 0 & 0 \\ \hline & M_{extra} & \end{bmatrix}$$

By inductive hypothesis, we have a choice choice$(i) \in [b]$ for each $i \in S_{old}$ such that the columns $(i, \mathrm{choice}(i))$ of M_{old} are linearly independent.

We need to find a set of linearly independent columns of M_{new} from distinct blocks.

Since projecting out the closure keeps the rest of the subspace safe, the set $V_{extra}[\setminus S]$ is a safe set. By definition of extra, for every $i \in S_{new} \setminus S_{old}$, there is a choice of index ind$(i) \in [b]$ such that the columns $(i, \mathrm{ind}(i))$ of M_{extra} are linearly independent. To complete the induction step, we need to exhibit a choice new-choice$(i) \in [b]$ for each $i \in S_{new}$ so that the columns $(i, \text{new-choice}(i))$ in M_{new} are linearly independent. We do this as follows:

$$\text{new-choice}(i) = \begin{cases} \text{choice}(i) & \text{if } i \in S_{old} \\ \text{ind}(i) & \text{if } i \in S_{new} \setminus S_{old} \end{cases}$$

It is easy to see the corresponding columns are linearly independent. If any linear combination is zero, that cannot include any columns from S_{old} - because the corresponding columns of M_{old} are linearly independent, and the newly added columns are 0 on the first k entries. So the linear combination can only include columns from $S_{new} \setminus S_{old}$- and this set of columns is linearly independent by construction of ind.

□

Proof. **(of Corollary 2.27)** By Lemma 2.26 we have that $\text{Cl}(A) = \text{Cl}(B)$, so A_y, B_y are nice affine spaces. It remains to show that $\text{codim}(B_y) = \text{codim}(A_y) + 1$.

Let $A = \{x | Mx = b\}$ and let the set of rows of M be $v_1, v_2, \cdots, v_k$. Let $B = \{x | \tilde{M}x = \tilde{b}\}$ where $\tilde{M}$ has $k+1$ rows, the first k of which are $v_1, v_2, \cdots, v_k$. Let the last row be w.

Let $S = \text{Cl}(A) = \text{Cl}(B)$. The set of defining linear forms of A_y is $v_1[\setminus S], v_2[\setminus S], \cdots, v_k[\setminus S]$ and the set of defining linear forms of B_y is $v_1[\setminus S], \cdots, v_k[\setminus S], w[\setminus S]$. We wish to show $w[\setminus S]$ is linearly independent from $v_1[\setminus S], v_2[\setminus S], \cdots, v_k[\setminus S]$. This is equivalent to showing that w does not lie in $\text{span}(\{v_1, v_2, \cdots, v_k\} \cup \{e_{i,j} | i \in S, j \in [b]\})$. FTSOC assume w lies in $\text{span}(\{v_1, v_2, \cdots, v_k\} \cup \{e_{i,j} | i \in S, j \in [b]\})$. Thus, $w = r + s$ for some $r \in \text{span}(v_1, v_2, \cdots, v_k)$ and $s \in \mathbb{F}_2^n$ such that $\text{supp}(s) \subseteq S$. Since Cl and $\tilde{\text{Cl}}$ of a set depend only on its linear span, we can WLOG replace w by s. Hence, assume $\text{supp}(w) \subseteq S$.

By Lemma 2.25, we have that

$$\tilde{\text{Cl}}(\{v_1, v_2, \cdots, v_k, w\}) \subseteq \tilde{\text{Cl}}(\{v_1, v_2, \cdots, v_k\} \cup \{e_{i,j} | i \in S, j \in [b]\}) = \tilde{\text{Cl}}(\{v_1, v_2, \cdots, v_k\})$$

This is a contradiction, since we assumed that

$$|\tilde{\text{Cl}}(\{v_1, v_2, \cdots, v_k, w)| = |\tilde{\text{Cl}}(v_1, v_2, \cdots, v_k)| + 1$$

□

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il