# On bounded depth proofs for Tseitin formulas on the grid; revisited*

Johan Håstad and Kilian Risse

*KTH Royal Institute of Technology*

December 5, 2022

**Abstract**

We study Frege proofs using depth-$d$ Boolean formulas for the Tseitin contradiction on $n \times n$ grids. We prove that if each line in the proof is of size $M$ then the number of lines is exponential in $n/(\log M)^{O(d)}$. This strengthens a recent result of Pitassi et al. [PRT22]. The key technical step is a multi-switching lemma extending the switching lemma of Håstad [Hås20] for a space of restrictions related to the Tseitin contradiction.

The strengthened lemma also allows us to improve the lower bound for standard proof size of bounded depth Frege refutations from exponential in $\tilde{\Omega}(n^{1/59d})$ to exponential in $\tilde{\Omega}(n^{1/(2d-1)})$.

*Supported by the Approximability and Proof Complexity project funded by the Knut and Alice Wallenberg Foundation.

# 1 Introduction

Mathematicians like proofs, formal statements where each line follows by simple reasoning rules from previously derived lines. Each line derived in this manner, assuming that the reasoning steps are sound, can give us some insight into the initial assumptions of the proof. A particularly interesting consequence is contradiction. Deriving an obviously false statement allows us to conclude that the initial assumptions, also called axioms, are contradictory. We continue the study of Frege proofs of contradiction where each line in the proof is a Boolean formula of depth $d$. This subject has a long tradition, so let us start with a very brief history.

A very basic proof system is resolution: each line of such a proof simply consists of a disjunction of literals. The derivation rules of resolution are also easy to understand and simple to implement, but the proof system nevertheless gives rise to reasonably short proofs for some formulas. It is far from easy to give lower bounds for the size of proofs in resolution but it has been studied for a long time and by now many strong bounds are known. An early paper by Tseitin [Tse68] defined an important class of contradictions based on graphs that is central to this and many previous papers. For each edge there is a variable and the requirement is that the parity of the variables incident to any given node sum to a particular bit which is called the charge of that vertex. If the sum of the charges is one modulo two this is a contradiction. For a subsystem of resolution, called regular resolution, Tseitin proved exponential lower bounds on refutations of these formulas. After this initial lower bound it took almost another two decades before the first strong lower bound for general resolution was obtained by Haken [Hak85], whose lower bound applied to the pigeonhole principle (PHP). Many other resolution lower bounds followed, but as we are not so interested in resolution and rather intend to study the more powerful proof system with formulas of larger, though still bounded, depth $d$ on each line, let us turn to such proof systems.

The study of proofs with lines limited to depth $d$ dates back several decades. A pioneering result was obtained by Ajtai [Ajt94] who showed that the PHP cannot be proved in polynomial size for any constant depth $d$. Developments continued in the 1990s and polynomial size proof were ruled out for values of $d$ up to $O(\log\log n)$ for both the PHP [PBI93, KPW95] as well as the Tseitin contradiction defined over complete [UF96] and expander graphs [Ben02].

These developments followed previous work where the computational power of the class of circuits[1] of depth $d$ was studied [Sip83, FSS84, Yao85, Hås86, Raz88, Smo87]. It is not surprising that it is easier to understand the computational power of a single circuit rather than to reason about a sequence of formulas giving a proof. This manifested itself in that while the highest value of $d$ for which strong bounds were known for size of proofs remained at $O(\log\log n)$, the results for circuit size extended to almost logarithmic depth.

This gap was (essentially) closed in two steps. First Pitassi et al. [PRST16] proved super-polynomial lower bounds for $d$ up to $o(\sqrt{\log n})$ and then Håstad [Hås20] extended this to depth $\Theta(\frac{\log n}{\log\log n})$ which, up to constants, matches the result for circuits.

The key technique used in most of the described results is the use of restrictions. These set most of the variables to constants which simplifies the circuit or formulas studied. If done carefully one can at the same time preserve the contradiction refuted or the function computed. Of course one cannot exactly preserve the contradiction and to be more precise a contradiction with parameter $n$ before the restriction turns into a contradiction of the same type but with a smaller parameter, $n/T$, after the restriction.

The simplification under a restriction usually takes place in the form of a switching lemma. This makes it possible to convert depth $d$ formulas to formulas of depth $d-1$. A sequence of

[1] When the depth is small, there is no major difference between circuits and formulas so the reader should feel free to ignore this difference.

restrictions is applied to reduce the depth to (essentially) zero making the circuit or formula straightforward to analyze. The balance to be struck is to find a set of restrictions that leave a large resulting contradiction but at the same time allows a switching lemma to be proved with good parameters.

In proof complexity the most commonly studied measure is the total size of a proof. There are two components to this size, the number of reasoning steps needed and the size of each line of the proof. In some cases, such as resolution, each line is automatically bounded in size and hence any lower bound for proof size is closely related to the number of proof steps. In some other situation the line sizes may grow and an interesting question is whether this can be avoided.

This line of investigation for Frege proofs with bounded depth formulas was recently initiated by Pitassi et al. [PRT22]. They consider the Tseitin contradiction defined over the grid of size $n \times n$, a setting where strong total size lower bounds for Frege refutations of bounded depth had previously been given by Håstad [Hås20]. If each line of the refutation is limited to size $M$ and depth $d$, then Pitassi et al. [PRT22] showed that the Frege proof must consist of at least $\exp(n/2^{O(d\sqrt{\log M})})$ many lines. For most interesting values of $M$ this greatly improves the bounds implied by the results for total proof size. In particular if $M$ is a polynomial the lower bounds are of the form $\exp(n^{1-o(1)})$, as long as $d = o(\sqrt{\log n})$, in contrast to the total size lower bounds of the form $\exp(n^{\Omega(1/d)})$. Pitassi, Ramakrishnan, and Tan [PRT22] rely on the restrictions introduced by Håstad [Hås20] but analyze them using the methods of Pitassi et al. [PRST16].

We study the same Tseitin contradiction on the grid and improve the lower bounds to $\exp(n/(\log M)^{O(d)})$, a bound conjectured by Pitassi et al. [PRT22]. These bounds are the strongest bounds that can be proved by the present methods and even if we cannot match them by constructing actual proofs we can at least represent the intermediate results of a natural proof by such formulas. We discuss this in more detail below.

### 1.1 Overview of proof techniques

The structure of the proof of our main result follows the approach of [PRT22] but relies on proving much sharper variants of the switching lemma.

In a standard application of a switching lemma to proof complexity one picks a restriction and demands that switching happens to all depth two formulas in the entire proof. Each formula switches successfully with high probability and by an application of a union bound it is possible to find a restriction to get them all to switch simultaneously.

The key idea of [PRT22] is that one need not consider all formulas in the proof at the same time. Rather one can focus on the sub-formulas of a given line. It is sufficient to establish that these admit what is called an $\ell$ common partial decision tree of small depth. This is a decision tree with the property that at each leaf, each of the formulas can be described by a decision tree of depth $\ell$. It turns out that this is enough to analyze the proof and establish that it cannot derive contradiction. The key property is that it is sufficient to only look at the constant number of formulas involved in each derivation step and analyze each such step separately.

The possibility to compute a set of formulas by an $\ell$ common partial decision tree after having been hit by a restriction is exactly what is analyzed by what has become known as a "multi-switching lemma" as introduced by [Hås14, IMP12]. This concept was introduced in order to analyze the correlation of a small circuits of bounded depth with parity but turns out to also be very useful in the current context.

Even though there is no general method, it seems like when it is possible to prove a

standard switching lemma there is good hope to also prove a multi-switching lemma with similar parameters. This happens when going from [Hås86] to [Hås14] and when going from [PRST16] to [PRT22]. We follow the same approach here and this paper very much builds on [Hås20]. We need a slight modification of the space of restrictions and changes to some steps of the proof, but a large fraction of the proof remains untouched. Let us briefly touch on the necessary changes.

The switching lemma of Håstad [Hås20] has a failure probability to not switch to a decision tree of depth $s$ of the form $(As)^{\Omega(s)}$ where $A$ depends on other parameters. As a first step one needs to eliminate the factor $s$ in the base of the exponent. This triggers the above mentioned change in the space of restrictions. This change enables us to prove a standard switching lemma with stronger parameters and, as a warm-up, we give this proof in the current paper. This results in an improvement of the lower bound for total proof size from $\exp(\tilde{\Omega}(n^{1/58d}))$ to $\exp(\tilde{\Omega}(n^{1/(2d-1)}))$. Even though the exponent's exponent is probably still off by a factor of 2, this is a substantial improvement in the parameters.

The high level idea of the proof of the multi-switching lemma is that for each of the formulas analyzed we try to construct a decision tree of depth $\ell$. If this fails then we take the long branch in the resulting decision tree and instead query these variables in the common decision tree. A complication that arises is that the answers on the long path in the local decision tree and the answers on a potentially long branch in the common decision tree are different. This causes us to analyze a new combinatorial game on the grid, as defined in Section 3.1.

### 1.2 Constructing small proofs

Let us finally comment on a possible upper bound; how to construct efficient refutations. If we are allowed to reason with linear equations modulo two then the Tseitin contradiction has efficient refutations. In particular on the grid we can sum all equations in a single column giving an equation containing $O(n)$ variables that must be satisfied. Adding the corresponding equation for the adjacent column maintains an equation of the same size and we can keep adding equations from adjacent columns until we have covered the entire grid. We derive a contradiction and we never use an equation containing more than $O(n)$ variables.

If we consider resolution then it is possible to represent a parity of size $m$ as a set of clauses. Indeed, looking at the equation $\sum_{i=1}^{m} x_i = 0$ we can replace this by the $2^{m-1}$ clauses of full width where an odd number of variables appear in negative form. Now replace each parity in the above proof by its corresponding clauses. It is not difficult to check that Gaussian elimination can be simulated by resolution. Given linear equation $L_1 = b_1$ and $L_2 = b_2$ with $m_1$, and $m_2$ variables respectively, and both containing the variable $x$ we want to derive all clauses representing $L_1 \oplus L_2 = b_1 \oplus b_2$. We have $2^{m_1-1}$ clauses representing the first linear equation and the $2^{m_2-1}$ clauses representing the second linear equation. Now we can take each pair of clauses and resolve over $x$ and this produces a good set of clauses. If $L_1$ and $L_2$ do not have any other common variables we are done. If they do contain more common variables then additional resolution steps are needed but these are not difficult to find and we leave it to the reader to figure out this detail. We conclude that Tseitin on the grid allows resolution proofs of length $2^{O(n)}$.

Let us consider proofs that contain formulas of depth $d$ and let us see how to represent a parity. Given $\sum_{i=1}^{m} x_i = 0$ we can divide the variables in to groups of size $(\log M)^{d-1}$ and write down formulas of depth $d$ and size $M$ that represent the parity and the negation of the parity of each group. Assume that the output gate of each of these formulas is an or. We now use the above clause representation of the parity of the groups and get a set of $2^{m/(\log M)^{d-1}}$

formulas of size $mM/(\log M)^{d-1}$ that represent the linear equations This means that we can represent each line in the parity proof by about $2^{n/(\log M)^{d-1}}$ lines of size about $M$. We do not know how to syntactically translate a Gaussian elimination step to some proof steps in this representation and thus we do not actually get a proof, only a representation of the partial results.

### 1.3 Organization

Let us outline the contents of this paper. We start in Section 2 and Section 3 with some preliminaries. In Section 4 we define the set of restrictions used in the current paper which are almost the same as in [Hås20]. We give some details how decision trees should be modified using local consistency in Section 5. The important tool for turning switching lemmas to lower bounds for proofs is by something called $t$-evaluations and we recall this in Section 6. Next we show how to construct these evaluations and derive our two main theorems assuming the new switching lemmas in Section 7. The strengthened version of the standard switching lemma is given in Section 8 and the extension to a multi-switching lemma is presented in Section 9. Large portions of the proof for the standard switching lemma as well as many definitions are identical to the proof of [Hås20]. We end with some conclusions in Section 10.

## 2 Preliminaries

We have a graph $G$ which we call "the grid" but to avoid problems at the perimeter we in fact use the torus. In other words we have nodes indexed by $(i, j)$, for $0 \leq i, j \leq n-1$ where $n$ is an odd integer and a node $(i, j)$ is connected to the four nodes at distance 1, i.e. where one coordinate is identical and the other moves up or down by 1 modulo $n$. For each node $v$ we have a *charge* $\alpha_v$ and for each edge $e$ in the graph we have a variable $x_e$. A Tseitin formula is given by a set of linear equalities modulo 2. That is, for each vertex $v$ in $G$ we have

$$\sum_{e \ni v} x_e = \alpha_v.$$

The main case we consider, which we call "the Tseitin contradictions" is when $\alpha_v = 1$ for each $v$. We do use more general charges in intermediate steps and hence the following lemma from [Hås20] is useful for us.

**Lemma 2.1.** *Consider the Tseitin formulas with charges $\alpha_v$. If $\sum_v \alpha_v = 0$ this formula is satisfiable and has $2^{r_n}$ solutions where the positive integer $r_n$ depends only on $n$ and not on the value of $\alpha_v$.*

As a converse to the above lemma, when $\sum_v \alpha_v = 1$ it is easy to see, by summing all equations, that the system is contradictory. In particular the Tseitin contradictions with $\alpha_v = 1$ for all $v$ are indeed contradictions for graphs with an odd number of nodes. We note that each Tseitin formula for the grid graph can be written as a 4-CNF formula by having 8 clauses of length four for each node.

We are interested in proofs in the form of deriving the constant false from these axioms. The exact reasoning rules turn out not to be of central importance but are stated in Section 6. The important properties of these rules are that they are sound and of constant size.

The sub-formulas that appear in this proof are allowed to contain only $\vee$-gates and negations. We simulate $\wedge$ using $\wedge F_i = \neg \vee \neg F_i$ and we define the depth of a formula to be the number of alternations of $\vee$ and $\neg$.

## 3 Properties of assignments on the grid and some games

We are interested in solutions to subsystems of the Tseitin contradictions. It follows from Lemma 2.1 that as soon as we drop the constraints at a single node we have a consistent system and indeed many solutions.

On a set $X$ of nodes we say that a partial assignment is *complete* if it gives values to exactly all variables with at least an endpoint in $X$. The support of a partial assignment $\alpha$ is denoted by $\text{supp}(\alpha)$ and is the set of nodes adjacent to a variable given a value. Note that the support of a complete assignment on $X$ also includes the neighbors of $X$.

We consider partial assignments that give values to few variables and in particular we are interested in cases where the size of the set $X$ is at most $2n/3$ and hence cannot touch all rows or columns of the grid. Let $X^c$ denote the complement of $X$.

In this case, $X^c$ contains a giant component containing almost all nodes of the grid. This follows as there are at least $n/3$ complete rows and columns in $X^c$ and the nodes of these rows and columns are all connected. The other, small, components of $X^c$ are important to control as an assignment on $X$ might fail to extend in a consistent way to such a component. To avoid this problem, for a set $X$ we let the *closure of* $X$, $\text{cl}(X)$ denote all nodes either in $X$ or in small connected components of $X^c$. Note that $\text{cl}(X)^c$ is exactly the giant component of $X^c$.

**Definition 3.1.** An assignment $\alpha$ with $X = \text{supp}(\alpha)$ is *locally consistent* if it can be extended to a complete assignment on $\text{cl}(X)$ that satisfies all parity constraints on this set.

We extend this definition to say that two assignments are consistent with each other if they do not give different values to the same variable and when you look at the union of the two assignment this gives a locally consistent assignment. The following lemma from [Hås20] is many times useful.

**Lemma 3.2.** *Suppose $\alpha$ is a locally consistent assignment where $|\text{supp}(\alpha)| \leq n/2$ and $x_e$ a variable not in $\text{supp}(\alpha)$. Then there is a locally consistent assignment $\alpha'$ that extends $\alpha$ and gives a value to $x_e$.*

We are interested in complete assignments on some sets $X$ and the grid and in particular how it looks from the outside. Let a *border assignment* be an assignment to the variables with one end-point in $X$ and one outside $X$. Such an assignment $\alpha$ is *achievable* if and only if there is an assignment that has the border assignment $\alpha$ and satisfies the parity conditions on $X$.

**Lemma 3.3.** *Let $X$ be a connected set. The a border assignment $\alpha$ is achievable if and only if the parity of the bits $\alpha$ equals the parity of the size of $X$.*

*Proof.* By induction on $|X|$, and the base case when $X$ is a singleton is obvious. For the induction step take any $v$ such that removing $v$ keeps $X$ connected. Of the variables next to $v$ some are forced by the border assignment. Fix the rest of the variables next to $v$ that satisfies the parity constraint at $v$. Apply induction to $X$ with $v$ removed and the border assignment including the just made assignment to the variables next to $v$. □

By a simple extension we have the following.

**Lemma 3.4.** *Let $X$ be a connected set. The a border assignment $\alpha$ is achievable by a locally consistent assignment if and only if the parity of the bits $\alpha$ equals the parity of the size of $X$ and this is true also for the border assignment of any small connected component of the complement of $X$.*

A process that is useful is the following dynamic matching game. We have two players, one adversarial player that supplies nodes while the other, matching player $P_M$, is supposed to dynamically create a matching that contains the nodes given by the adversarial player. Our strengthened lower bound for the size of a proof uses the same combinatorial lemma as the proof in [Hås20] namely the following.

**Lemma 3.5.** *[Hås20] When the dynamic matching game is played on the $n \times n$ grid, $P_M$ can survive for at least $n/2$ moves.*

The purpose of this lemma is to find which variables to include in the extended decision tree used. As discussed in the introduction our new lower bound for the number of lines of a proof with short lines needs a multi-switching lemma and it turns out that the decision which variables to include is described by a more complicated combinatorial game. We now discuss this game. The reader that wants motivation for this game is encouraged to first read the proof of the standard switching lemma to find the reason for Lemma 3.5 and then start reading the proof of the multi-switching lemma.

### 3.1 Another game on the grid

The game is played on the grid between an adversary $A$ and a player $P$. They take turns picking vertices and edges on the $n \times n$ grid. Once a vertex is picked it can never be picked again. The set of picked vertices is called $S$. The vertices outside $S$ are called "free". The total number of picked nodes always remains less than $n/2$ and hence there is always a large connected component in the complement of $S$. Other connected components in the complement are called "small". Some of the picked elements are called "active".

The task of $P$ is to pick as few vertices as possible such that the following properties hold.

1. The number of picked nodes has the correct parity in some special components of $S$ described below.
2. The size of any small component in the complement of $S$ is even.

The game starts with an empty grid, and takes place in rounds where $A$ decides when to start the next round. $A$ can do two types of moves.

1. Pick an arbitrary new vertex $v$ and make it active. This is called a "simple" move.
2. Declare that a round is over. In this case $A$ can make any edge between an active vertex and a free vertex active. Each connected component must have an even number of activated edges leading in to it.

   This second type of move is called a *completion* move. When this move is completed all vertices become inactive and the next round starts.

After a simple move $P$ must pick some vertices to form a connected component of even size jointly with the just placed vertex. $P$ must also make sure that each connected component of the complement is of even size. Any vertex picked by $P$ in response to a simple move becomes active. Note that in this situation $P$ picks an odd number of vertices and hence at least one.

After a completion move $P$ must pick the free vertices with at least one adjacent active edge. It may pick some more vertices to achieve the following.

1. The parity of the size of each connected component of the just picked vertices must equal to the parity of active edges adjacent to it.

2. The number of vertices in any small connected component in the complement is even.

Although this looks complicated please note that if the there is only one active edge going in to the nodes $P$ must pick and these do not split any connected component of the complement, then these forced nodes is all that $P$ needs to select.

What forces $P$ to act in general is the creation of small odd size components in the complement of $S$ due to making the "obvious" choices. For any such component $C$, $P$ needs to add vertices to $S$ to make it of even size. It is also restricted to only adding vertices adjacent to a supplied starting vertex. This vertex is in $S$ but connected to at least some vertex in $C$. We call this "evenizing" with starting point $w$. All connected components of the complement created in this process must be made of to be of even size. It is simple to see that this can always be done, simply add any vertex adjacent to $w$. If this does not split $C$ in to at least two components then $P$ is done. Otherwise $P$ can simply recurse on any created component of odd size with the chosen vertex as the starting point. We must prove that, over the course of the entire game, $A$ cannot force $P$ to add too many vertices. To get some understanding of the problems, let us first give an example where $P$ is forced to make many moves.

**Example.** Suppose $C$ consists of the vertices $(1, x)$ for $1 \leq x \leq t$ jointly with $(2, x)$ for even $x$ at most $t - 1$ and $(0, x)$ for odd $x$ at most $t - 1$. This has an odd number of vertices (in fact $2t - 1$) and suppose the starting vertex is $(1, 0)$. $P$ needs to add $(1, 1)$ since this is the only vertex in $C$ connected to $(1, 0)$. This creates the isolated vertex $(0, 1)$ and a component of size $2t - 3$ that is very similar to the starting component. It is easy to see that $P$ ends up picking all vertices of this component.

We set up a potential function to prove that such massive responses as in the example can only happen rarely. For each connected component of the complement consider its edges to elements of $S$. For each edge to an active vertex we assign four points and for each other edge one point. Suppose the total number of points for component $C_i$ is $f_i$ and this number is called the *score* of $C_i$. We have a parameter $T$ and we say that each component of size at most $T$ is *ultra small*. We later fix $T$ to a suitable constant. A component that is not ultra small is called *sizeable*. This includes the large component. We now define the potential as

$$\sum_i f_i + G - D(F - 1)$$

where the sum is over components that are sizeable, $F$ the number of components that are sizeable, $D$ is a constant to be chosen suitably, and $G$ is the number of ultra small components. For $G$ we only count a component the first time it becomes ultra small. Further splitting of an ultra small component is ignored. The reason for using $(F - 1)$ is that we want to start the potential at 0 and hence not count the large component in this number.

We want to prove that this potential increases by at most a constant for each simple move and decreases by at least one half for other moves. By setting $T$ large enough (after we have chosen $D$) we make sure that $f_i \geq 2D$ for any component of size at least $T$.

Let us first analyze simple moves. When $A$ chooses a vertex it might increase $\sum_i f_i$ by at most 16. This might also cause a component of the complement of $S$ to split. To analyze the cost of such a split we first pay the increase by the addition of the extra vertex to $S$ in the form of increase to $f_i$. We then see how the splitting of a component of the complement affects the potential. First note that splitting an ultra small component does not affect the potential (remember that we do not count this as an increase in $G$) and thus we are interested in splitting a sizeable component. We have sequence of simple lemmas.

**Lemma 3.6.** *If a sizeable component splits into ultra small components then the potential decreases by at least $D - 4$.*

*Proof.* Suppose $C_i$ splits. This means that the term $f_i - D$ disappears in the potential. By construction this is at least $D$. We might have an increase of $G$ by 4 but no other increase. The lemma follows. □

Next we have.

**Lemma 3.7.** *If a sizeable component splits and the result contains at least two sizeable components then the potential decreases by at least $D$.*

*Proof.* The creation of a sizeable component increases $F$. Any ultra small component created increases $G$ by one but at the same time its score is removed from the sum causing a decrease of that sum by at least 4. □

Finally we analyze the third possibility.

**Lemma 3.8.** *If a sizeable component splits into a sizeable component and one or more ultra small components the potential decreases by at least three for each component split off.*

*Proof.* The value of $F$ does not change. Any ultra small component created increases $G$ by one but its score of at least 4 is removed from the sum $\sum_i f_i$. □

The above lemmas imply that the splitting of components only decreases the potential. What remains is to analyze the cost when $P$ is forced to evenize an odd size component. By "cost" we here mean increase in potential. We might have a negative cost which is a decrease in potential.

**Lemma 3.9.** *The cost of evenizing a component with an active starting point is at most $11 - m/2$ where $m$ is the number of moves made by $P$ in sizeable components. The cost of evenizing an ultra small component is 0.*

*Proof.* We prove the lemma by induction over the size of the component. If the component is ultra small then no term of the potential can change so there is no cost.

As a first attempt let $P$ pick an arbitrary vertex, $v$, next to the starting which we call $w$. If this does not result in any new odd size component we are done. We have added at most three more edges at cost four each while eliminating the cost of $(v, w)$. As a result $f_i$ might have increased by at most 8 giving the same increase in the potential and $P$ has made one move in sizeable component.

Now suppose that choosing $v$ creates some new odd size components that have to be evenized. We know that this number must be even and since any component has to be adjacent to $v$ and since $v$ has at most three neighbors other than $w$ there must be exactly two such components and call them $C_1$ and $C_2$. Let $v_i$ be an element in $C_i$ that is a neighbor of $v$. Suppose the scores of these two components are $f_1$ and $f_2$ and the score of the component that splits is $f$. Note that $f$ is measured before $v$ is placed in $S$ while $f_1$ and $f_2$ are measured after this has happened and thus we need to keep track of what happens to edges next to $v$. One fact to our advantage is that while $(v, w)$ was counted in $f$ its four points do not appear in neither $f_1$ or $f_2$.

There are a number of cases depending on the status of the fourth neighbor of $v$ (on top of $w$, $v_1$ and $v_2$). It can be in a third, new, component, be an element of $S$, or belong to $C_1$ or $C_2$. In the first case that third component is of even size and hence need not be evenized. If it is sizeable we get a decrease in potential of at least $D$ and if it is ultra small by at least

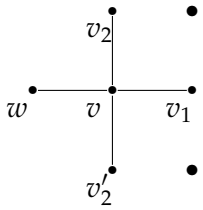


Figure 1: The larger circles are elements of $S$

three. In either case we are doing strictly better then if this node belongs to $S$ and hence we can ignore this case and we may assume that we get exactly two new components.

If neither of these two new components is ultra small, then the potential decreases by Lemma 3.7. The total increase in the score is bounded by 8 as we eliminate the score of $(v, w)$ and add at most 3 new edges with three points each. The cost, by induction, to evenize $C_1$ and $C_2$ is at most $22 - (m_1 + m_2)/2$ where $m_i$ is the number of moves of $P$ made in sizeable components when evenizing $C_i$. The total change to the potential is thus bounded from above by $30 - (m_1 + m_2)/2 - D$ and making sure that $D \geq 20$ the lemma follows in this case.

Now suppose $C_1$ and $C_2$ are both ultra small. Then $G$ increases by two but we have a decrease of $D$ in potential by Lemma 3.6 and in this case we in fact have a total decrease in the potential and no more moves in sizeable components. This establishes the lemma in this case.

Finally assume that $C_1$ is ultra small while $C_2$ is not. In this case we get an increase of $G$ by one while $F$ does not change. We need to analyze the change in scores and the cost of possible recursive calls.

If the fourth neighbor of $v$ (on top of $v_1$, $w$ and $v_2$) does not belong to $C_2$ then $f_2 \leq f - 3$. This follows as the only new edge in $C_2$ that was not present in $C$ is $(v_2, v)$ but this is compensated by $(v, w)$ being present in $C$ but not in $C_2$. On top of this at least three points have disappeared from $f$ when forming $C_1$. For the recursive costs we have that, by induction, the cost to evenize $C_2$ is at most $11 - m_2/2$. As $C_1$ is ultra small we have no cost for its recursive call. The net cost is thus bounded from above by $1 - 3 + 11 - m_2/2 \leq 10 - m/2$ and the lemma follows also in this case.

Finally consider the case when the missing neighbor of $v$, call it $v_2'$, belongs to $C_2$. This causes the potential addition of 4 to $f_2$ by the edge $(v, v_2')$ and this needs to be addressed. Unfortunately this leads to a rather tedious case analysis.

Suppose without loss of generality that $w$ is to the left of $v$. We first have three cases whether $v_1$ is to right, above, or below $v$. The cases above and below are symmetric so in fact we can drop the case of $v_1$ being below $v$.

Let us assume that $v_1$ is to the right of $v$ and $v_2$ above and $v_2'$ is below. The situation looks like in Figure 1, where we note that the vertices to the right of $v_2$ and $v_2'$ must be in $S$ to make removing $v$ disconnect $C_1$ and $C_2$.

Now suppose that we can remove $v_2$ from $C_2$ and keep it connected. Then $P$ can pick $v$ and $v_2$ and the remaining part of $C_2$ is even and there is no recursive call. Let us compare $f_2$ and $f$. We have lost at least 3 points from $f$ that now belongs to $C_1$. We also lost 4 points from $(w, v)$ becoming internal of $S$. We gain 4 points from $(v, v_2')$. Finally we can have two new edges next to $v_2$ (going left and up). There is a net gain in potential of at most 5 and the lemma follows also in this case as $P$ only made two moves.

The case when we can remove $v_2'$ and keep $C_2$ is connected is symmetric and hence we need to consider the case when both create new components and thus we can assume that

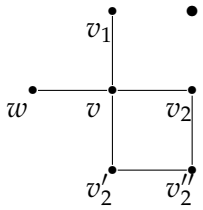


Figure 2: The larger circle is an elements of $S$

both removing $v_2$ and $v_2'$ splits $C_2$. The two components that $C_2$ splits to when $v_2$ is removed must then be connected to $v_2$ from top and from the left and for $v_2'$ the two components attach from left and below.

Put both of $v_2$ and $v_2'$ into $S$. Then $C_2$ splits in to three components, two of which might have to be evenized. If two of these are sizeable then we have increased $F$ by one. The analysis is very much as before and the extra decrease of $D$ provided by Lemma 3.7 well compensates for the two recursive evenizing calls.

The case when $C_2$ splits into three ultra small components is also very similar to previous cases. There is no recursive call and Lemma 3.6 provides a large decrease. The case that remains to analyze is that we have exactly two ultra small components.

We have (remember we also have $C_1$) created three ultra small components. Each decreases the score by at least three and increase $G$ by three for a net decrease of 6. The edge $(v, w)$ is now interior to $S$ while it is counted in $f$. The only new edges in $f_2$ are one from each of $v_2$ and $v_2'$. Thus we have net decrease of 2. We still have one recursive call on the remaining component that is sizeable but this costs, by induction, at most $11 - m/2$ where $m$ is the number of nodes chosen by $P$ in this recursive call. The lemma follows in the case when both $v_2$ and $v_2'$ disconnect $C_2$.

We have the final case when $v_2$ is to the right of $v$ and $v_2'$ is below. Suppose first that adding either $v_2$ or $v_2'$ to $S$ does not disconnect $C_2$. Then if $P$ removes this vertex and $v$ and there is no recursive call. Suppose it removes $v_2'$ (the case of $v_2$ being similar). Then we might get three new edges costing four points next to $v_2'$ and the edge $(v, v_2)$ is of the same cost while the cost of $(v, w)$ disappears. At least three points disappear with the creation of $C_1$ while there is an increase of one for $G$. This implies that there is an increase of at most 10 and as $P$ has picked two vertices the lemma follows in this case.

We need to analyze the situation when both removing either $v_2$ and $v_2'$ disconnects $C_2$. Let us first observe that the vertex $v_2''$ in the picture belong to $C_2$ since otherwise removing $v_2$ does not disconnect $C_2$. The situation is like in Figure 2.

Now, consider putting all four vertices $v$, $v_2$ $v_2'$, and $v_2''$ in $S$. This splits $C_2$ into a number of components as we may have one component hanging off each side of the square. If at least two are sizeable we get an increase in $F$ and Lemma 3.7 takes care of of the local costs and we can apply induction. Similarly if all components are ultra small Lemma 3.6 tells us that there is a decrease in potential. We need to analyze what happens when exactly one component is sizeable.

In fact we must have three ultra small components hanging off the square each giving a net decrease of at least 2. Indeed we have $C_1$ and the ultra small components created when $v_2$ and $v_2'$ are removed. Since we have a sizeable component we must have one component hanging off each of the four sides of the cube, as we cannot have two components attaching to the same side.

We only have one recursive call with a cost of $11 - m/2$, and we have net decrease in 6

from the ultra small components. Finally for edges, we do not any more count the cost of $(v, w)$ and we can only have two new edges entering the component of the recursive call. The new edges to the ultra small components do not count. Thus apart from the recursive call we have a net decrease of 2 and this compensates for the four points added by $P$. □

The above takes care of all simple moves. Let us look at completion moves.

**Lemma 3.10.** *A completion decreases the potential by at least the number of active edges chosen by A. This includes the forced response by P*

*Proof.* The first that happens is that an edge which costs 4 is replaced by an inactive vertex next to it. This results in at most three edges of cost one and is hence a decrease of at least one in potential. If several active edges go to the same vertex $P$ has to add two vertices but this gives a decrease of at least two. Now unless this causes a split of a component we are done.

If it splits an ultra small component then there is no further change in the potential. If it splits a sizeable component then we might have to evenize two components and the following lemma is what we need.

**Lemma 3.11.** *The cost of evenizing a component with an inactive starting point is at most* $3 - m$ *where* $m$ *is the number of vertices added by* $P$ *to sizeable components. The cost of evenizing an ultra small component is at most 0.*

Let us assume this lemma then finish the proof of Lemma 3.10. As many times previously unless we get exactly one non ultra small component it is easy to prove that there is a decrease so assume that this is the case. Each ultra small component decreases the potential by a least three and this is sufficient to pay for the evenizing of the component and this is demonstrated by Lemma 3.11. □

The proof of Lemma 3.11 is surprisingly much simpler than the proof of Lemma 3.9. The key difference is that new edges added only cost one and not four. This makes it much easier to compensate the cost of new edges by the loss in potential due to the appearance of ultra small components.

*Proof of Lemma 3.11.* If the response of putting a vertex, $v$, next to the starting vertex is sufficient then we have $m = 1$ and the potential increases by at most 2 as three edges are added and one is removed. The lemma is thus true in this case and let us analyze what happens to the potential if $v$ causes the component of the complement to split. As before, unless it is a sizeable component that splits and the result is exactly one sizeable component and one or more ultra small components, we do have a substantial decrease in the potential due to the loss of a term $D$.

As in the previous proof the worst case is when $v$ splits the component into two components $C_1$ and $C_2$ where the first is sizeable and the the second is ultra small and the third neighbor of $v$ belongs to $C_1$. In this case we have added two more edges of $v$ into $C_1$. We have removed one edge (between $v$ and the starting point) and lost the cost of at least three edges that are now part of $C_2$. This is a net loss of two to the potential. We need to evenize $C_1$ and this cost by induction at most $3 - m_1$ if $P$ picks $m_1$ vertices in this process. Finally we have one more ultra small component and thus the total cost is at most $2 - m_1$. Since $P$ picks $m_1 + 1$ vertices in total, the lemma follows. □

We finally state the conclusion of this section.

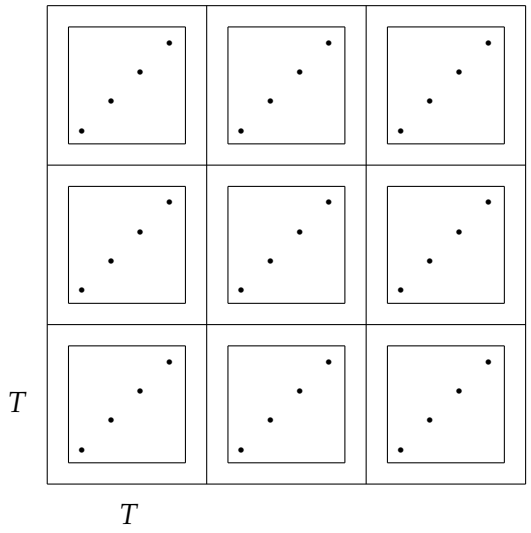


Figure 3: The centers and central areas

**Lemma 3.12.** *If $A$ makes $s$ simple moves in the game, then the total number of moves is bounded by $O(s)$.*

*Proof.* The potential increases by $O(1)$ for each simple move of $A$. The evenizing of any odd component created costs at most $O(1)$ but is decreased by $1/2$ for any vertex chosen by $P$ is a sizeable component. We conclude that the total number of moves in sizeable components is at most $O(s)$.

As the number of ultra small component created is bounded by the potential, their number is $O(s)$. In each such component there are $O(1)$ moves. □

# 4 Restrictions

We use (essentially) the same space of random restriction as [Hås20]. The only difference is the choice in the number of live centers in the partial restrictions. This is the parameter $k$ below which changes its value from $Cs(n/T)^2$ to $C \log n(n/T)^2$. For completeness we repeat all definitions from [Hås20] but we keep the description brief and for intuition and motivation we refer to [Hås20].

## 4.1 Full restrictions

In an $n \times n$ grid we make sub-squares of size $T \times T$ where $T$ is odd. In each sub-square we choose[2] $\Delta = \sqrt{T}/2$ of the nodes and call them *centers*. These are located evenly spaced on the diagonal of the $3T/4 \times 3T/4$ central sub-square. This implies that they have separation $3\sqrt{T}/2 = 3\Delta$ in both dimensions. A schematic picture of this is given in Figure 3.

The centers in neighboring sub-squares are connected by paths that are edge-disjoint except close to the endpoints. Let us describe how to connect a given center to a center in the sub-square on top. As there are $T/4 = \Delta^2$ rows between the two central areas, for each pair

[2]For simplicity we assume that some arithmetical expressions that are supposed to be integers are in fact exact as integers. By a careful choice of parameters this can be achieved but we leave this detail to the reader.

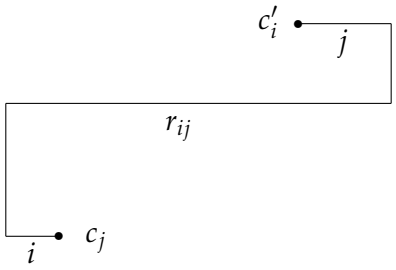


Figure 4: A path

of centers (the $j$th center, $c_j$ in the bottom sub-square and $i$th center $c'_i$ in the top sub-square) we can designate a unique row, $r_{ij}$ in this middle area.

To connect $c_j$ to $c'_i$ we first go $i$ steps to the left and then straight up to the designated row $r_{ij}$. This is completed by starting at $c'_i$ and then going $j$ steps to the right and down to the designated row. We finally use the appropriate segment from the designated row to complete the path (which might be in either direction). A picture of this is given in Figure 4. We index the centers from 1 to $\Delta$ and hence each path consists of 5 non-empty segments. The first and last segments are totally within the central area while the middle segment is totally in the area between the central areas. Segments two and four go from the central areas to the area in-between.

Connecting $c_j$ to a center $c'_i$ in a sub-square to the left is done in an analogous way. There is a unique column $c_{ij}$ reserved for the pair and the path again consists of five non-empty segments. The first and last segments consist of $i$ vertical edges up from $c_j$, and $j$ vertical edges down from $c'_i$. We add horizontal segments connecting to the designated column $c_{ij}$ the and middle segment is along this column. The below lemma is proved in [Hås20].

**Lemma 4.1.** *The described paths are edge-disjoint except for the at most $\Delta$ edges closest to an endpoint. For each edge $e$, if there is more than one path containing $e$, these paths all have the same endpoint closest to $e$.*

We let the term *closest endpoint* of an edge denote the closest endpoint breaking ties in an arbitrary way. The key property we need is that the "closest endpoint" of a path through an edge is uniquely defined by the edge.

We define the *direction* of a path to be the relative positions of the sub-squares of its two endpoints. It is true that the paths are undirected but at times when we consider paths from a fixed center $v$ it is convenient to think of such paths as starting at $v$ and thus speak of paths going left or right from $v$ rather than sideways. We note that apart from having the same closest endpoint, all paths through one fixed edge $e$ have the same direction.

A restriction is defined by first choosing one center in each $T \times T$ sub-square and then the paths described above connecting these centers. Note that these paths are edge-disjoint. The chosen centers naturally form a $n/T \times n/T$ grid if we interpret the paths between the chosen centers as edges. We proceed to make the correspondence more complete by assigning values to variables.

We choose a solution to the Tseitin formula with charges 0 at the chosen centers and 1 at other nodes. As the number of chosen centers is odd, by Lemma 2.1, there are many such solutions. For variables not on the chosen paths these are the final values while for variables on the chosen paths we call them *suggested* values.

For each path $P$ between two chosen centers we have a new variable $x_P$ and for each variable $x_e$ on $P$ it is replaced by $x_P$ if the suggested value of $x_e$ is 0 and otherwise it is

replaced by $\bar{x}_P$.

We claim that with these substitutions we have reduced the Tseitin problem on an $n \times n$ grid to the same problem on an $n/T \times n/T$ grid. This is true in the sense that we have an induced grid when we interpret paths as new edges and we need to see what happens to the axioms.

Given a formula $F$ we can apply a restriction $\sigma$ to it in the natural way resulting in a formula denoted by $F\lceil_\sigma$. Variables given constant values are replaced by constants while surviving variables are replaced by the appropriate negation of the corresponding path-variable. A restriction has a natural effect on the Tseitin contradiction as follows.

- The axioms for nodes not on a chosen paths are all reduced to true as all variables occurring in them are fixed in such a way that the axioms are true.
- The axioms for interior nodes of a chosen path are reduced to tautologies as the axiom is true independent of the value of the involved variable(s) $x_P$. This is true as flipping a single $x_P$ changes the value of two variables next to any such node.
- The axioms at the chosen centers turn into the axioms of the smaller instance.

These just defined restrictions are called *full restrictions* and a typical full restriction is denoted by $\sigma$. Note that these full restrictions are really "affine restrictions" in the vocabulary of [RST15] as they do not only assign values to variables but also identify several old variables with the same new variable that might also be negated. For simplicity, however, we keep the simpler term "restrictions".

### 4.2 Partial restrictions and pairings

A typical partial restriction is called $\rho$ and as we mostly discuss partial restrictions we simply call them "restrictions" while we use the term "full restrictions" when that is what we have in mind. At the same time as describing partial restrictions we give a probability distribution on such restrictions.

Let $k$ be an odd integer of the form $C \log n(n/T)^2$ for a constant $C$ to be determined. The first step of constructing $\rho$ is picking $k$ centers uniformly at random from the set of all $\Delta(n/T)^2$ centers defined in the previous section. These are the *alive* centers. In the future we only consider the case when the number of live centers in each sub-square is between a factor .99 and 1.01 of its expected value $C \log n$. By choosing $C$ appropriately the probability of this being false is can be made to be $1/n$.

We define charges that are 0 for all live centers and 1 for dead centers. As the number of live centers is odd we can apply Lemma 2.1 and pick a random solution with these charges to the Tseitin formula. For edges not on paths between live centers these are final values while for variables on such paths we call them *preferred* values.

The choice of the centers together with the fixed and preferred variables is denoted by $\rho$. The choice of $\rho$ is the main probabilistic event. Note that by Lemma 2.1 the number of possible values for fixed and preferred values is independent of which centers are alive and even of $k$ as long as it is odd.

We now describe how to turn a partial restriction $\rho$ into a full restriction $\sigma$. Choose one center to survive in each sub-square[3]. These are called the *chosen centers* and paths between such centers correspond to the variables that remain and are called *chosen paths*. Centers that

[3]This choice can be done in an arbitrary way but to be definite let us choose the center from the lowest numbered row.

were alive through the first part of the process but are not chosen are called *non-chosen*. The centers killed already by $\rho$ are simply called dead. We proceed to define a pairing.

**Definition 4.2** (pairing)**.** A *pairing* $\pi$ is a graph supported on the non-chosen centers. Each component of $\pi$ is either a single edge or a star of size four with one center and three nodes of degree one. Connected centers are located in adjacent sub-squares.

The following lemma follows from the proof of the corresponding lemma from [Hås20] which had the parameter $s$ instead of $\log n$.

**Lemma 4.3.** *If each sub-square has between* $.99C \log n$ *and* $1.01C \log n$ *non-chosen centers, a pairing* $\pi$ *exists.*

Let us establish some notation. As the original grid is also a graph with edges we from now on use the term "grid-edges" to refer to edges in the original grid. The chosen centers form a smaller grid and this also has edges and we call these "new grid-edges". We only consider paths in the original grid and we keep the shorter term "path" for these. In other words, from now on an "edge" is a connection between two live centers and corresponds to a path in the grid-graph. A "new grid-edge" corresponds to a chosen path and is thus also an edge in the graph of the live centers. We say that two chosen centers are neighbors if they are in adjacent sub-squares.

Some edges are conflicting in that we do not allow them to be present in the graph at the same time. More precisely we allow at most one path in each of the four directions from a center. As picking a path corresponds to changing the variables on this path this is the same as saying that the variables can only change values at most once.

As stated above $\pi$ makes it possible to turn $\rho$ into $\sigma$. Variables not on live paths take their fixed values. Variables on live paths but not on chosen paths take their preferred values unless they are on a path chosen by $\pi$ in which case these values are negated. On the chosen paths, the preferred values now becomes suggested and this completes the description of $\sigma$. Thus $\sigma$ is obtained deterministically from $\rho$ and $\pi$ and when we want to stress this dependence we sometimes write $\sigma(\rho, \pi)$.

We use the term "preferred values" as a vast majority of the variables will eventually be fixed to these values as very few variables appear on the paths of $\pi$ or turn into suggested values. On the other hand "suggested values" are much less certain as the path-variables should be thought of as equally likely to be 0 and 1 and thus these variables are equally likely to take also the non-suggested value.

As an intermediate between $\rho$ and the full restriction $\sigma$ we have $\rho$ and some information in the form of existence or non-existence of edges. We have the following definition.

**Definition 4.4** (information piece)**.** An *information piece* is either in form of an edge $(v, w)$ for two centers $v$ and $w$ or $(v, \delta, \perp)$ where $v$ is a center and $\delta$ is a direction (i.e. "left", "right" "up" or "down"). The former says that there is an edge from $v$ to $w$ while the latter says that there is no edge from $v$ in the direction $\delta$.

We note that, as edges are undirected, $(v, w)$ and $(w, v)$ denote the same information. We also use sets of information pieces.

**Definition 4.5** (consistent information set)**.** An *information set* $I$ is a collection of information pieces. Its *support*, denoted by $\text{supp}(I)$, is the set of centers mentioned in these pieces. An information set $I$ is *consistent* if

1. it does not have two different pieces of information from the same center in one fixed direction, and

2. if $I$ has information in all four directions from a center $v$ then it has an odd number of edges touching $v$.

A partial assignment to some path-variables naturally corresponds to a set of information pieces. An assignment of 0 to a path-variable gives two non-edges, in the appropriate directions, with closest end-points at the two chosen centers connected by this path. An assignment of 1 gives an information piece in the form of an edge between the two chosen centers. We use the term "consistent" both for sets of information pieces and partial assignments. Consistency for assignments requires an odd number of ones adjacent to any center that has all its variable assigned and this exactly corresponds to the property of information pieces in all four directions in the definition above. This makes the two notions close and hence using "consistent" for both should hopefully not confuse the reader.

Jointly with $\rho$ an information set fixes the values of some more variables as follows.

**Definition 4.6** (forcing)**.** Let $\rho$ be a restriction and $I$ an information set. A variable $x_e$ is considered *forced by* $(\rho, I)$ if and only if either its closest endpoint, $v$, is not live in $\rho$ or if the information of $v$ in the direction of $e$ is contained in $I$. It is forced to its preferred value in $\rho$ unless the relevant information piece states that there is an edge from $v$ in the direction of $e$ that corresponds to a path that passes through $e$ in which case it takes the opposite value. Variables not on live paths take the value given by $\rho$.

There are other situations where the value of a variable might be determined by $\rho$ and $I$, such as the lack, or scarcity, of live centers in a sub-square but we do not use this information in the reasoning below. We need the notion of a closed information set.

**Definition 4.7** (closed information set)**.** An information set $I$ is *closed* if $I$ is consistent and for each $v \in \mathrm{supp}(I)$, the set $I$ contains the information in all four directions.

The definition implies that for any $v \in \mathrm{supp}(I)$, in any direction $\delta$ where there is not an element of $\mathrm{supp}(I)$, we have a non-edge $(v, \delta, \perp)$. When considered as a graph such an information set is an odd-degree graph (with degrees one and three) on the centers of $\mathrm{supp}(I)$.

Note that if we have a closed information set $I$ then if we consider all variables forced by $(\rho, I)$ this can be described by a restriction where the centers in the $\mathrm{supp}(I)$ are killed. We simply negate the values of any preferred variable on any path in $I$ and then forget that the centers in $\mathrm{supp}(I)$ were alive.

Thus, if we let such a closed information set operate on a restriction $\rho$ we get a restriction with fewer live centers where the number of killed centers is exactly the number of centers in the support of the corresponding graph.

### 4.3 Generalized restrictions

In our proof we allow generalized (partial) restrictions. These are like standard restrictions but we allow the violation of the Tseitin condition at some dead centers. Such centers are called *bad* and we keep a close track on their number. These generalized restrictions are only used for book-keeping reasons.

## 5 Decision trees

We have decision trees where each internal node is marked with a variable and the outgoing edges are marked with 0 and 1. The leaves of a decision tree are labeled by 0 and 1. We allow decision trees of depth 0 which are constants 0 or 1.

All decision trees considered in this paper have a depth that is smaller than half the dimension of the grid we are currently considering. For each branch in a decision tree there is minimal partial assignment, $\tau$ such that any extension of this partial assignment creates an assignment that follows this path. We use this $\tau$ to identify that branch and we call it *consistent* if $\tau$ is consistent in the sense of Definition 3.1.

We trim decision trees to maintain the property that all branches of a decision tree are consistent. When a decision tree is created this is not a problem but trimming takes place when we consider what happens under a partial assignment $\tau$ or a full restriction $\sigma$. In that latter case, the initial decision tree uses the variables $x_e$ while the resulting decision tree uses the new variables $x_P$.

We sometimes think of a decision tree $T$ as the set of all branches leading from the root to the leaves. These have labels and fit together in a tree structure and each corresponds to a partial assignment $\tau'$ as discussed above. When creating the decision tree after $\tau$ or $\sigma$ the idea is to keep all branches that are consistent with the new information.

In the case of a partial assignment $\tau$ we keep all branches corresponding to $\tau'$ such that $\tau$ and $\tau'$ are consistent as discussed after Definition 3.1. In the case of a full restriction $\sigma$ the situation is not difficult but slightly more complicated so let us define this explicitly.

The assignment $\tau'$ assigns values to some variables $x_e$. Some of these are given values by $\sigma$ while the rest are now on chosen paths. To be consistent we require that for the variables given values by both $\sigma$ and $\tau'$, the two values agree. For each variable $x_e$ given a value by $\tau'$ we get a value for the corresponding path-variable $x_P$. For $\sigma$ and $\tau'$ to be consistent we require that no $x_P$ gets two conflicting values and that the values $x_P$ are consistent in the sense of Definition 3.1 when considered as an assignment on the smaller grid.

The key property that we need is that if the depth of $T$ is small enough then at least some branch of $T$ is consistent with $\tau$ or $\sigma$. In the former case we make sure that the total number of assigned variables under $\tau$ and $\tau'$ is at most half the dimension of the grid and in the latter case that the depth is a most half the dimension of the grid after $\sigma$. This together with the fact for each internal node of $T$ has out-degree two and Lemma 3.2 makes sure that some branch is consistent.

Once we have identified which branches remain it is easy to see that they form a decision tree. In fact it is also possible to define the new decision tree by a dynamic process where we start at the root of $T$ and consider each node in the tree. As we walk down the tree we can, for each node, check whether both values of the current variable are consistent with the partial assignment of the branch so far jointly with $\tau$ or $\sigma$. For a full restriction $\sigma$ we of course take into account that once we have determined the value on one variable on a path, all the other variables on the same path are determined. If only one value is consistent we eliminate the other sub-tree while if both values are consistent we have found a node in the new tree. In some situations we might get a tree which has a single branch consistent with $\tau$ or $\sigma$. This is considered a depth-0 tree with only one leaf. For a decision tree $T$ we let $T\lceil_\tau$ we the decision tree after we have applied $\tau$.

We let a *1-tree* be a decision tree where all leaves are labeled 1 and define a *0-tree* analogously. Special cases of such trees are trees of depth 0. We say $T\lceil_\pi = b$ if the decision tree given by $T_i\lceil_\pi$ is a $b$-tree.

We say that a decision tree $\mathcal{T}$ is an $\ell$ common partial decision tree for $T_1, \ldots, T_m$ of depth $t$ if

1. $\mathcal{T}$ is of depth $t$, and
2. for every $T_i$ and branch $\pi$ in $\mathcal{T}$ there are decision trees $T(i, \pi)$ of depth $\ell$ such that the following holds. Let $\mathcal{T}_i$ be the decision tree obtained from $\mathcal{T}$ by appending the trees

$T(i, \pi)$ at the corresponding leaf $\pi$ of $\mathcal{T}$. Then, if a branch $\pi'$ in $\mathcal{T}_i$ ends in a leaf labeled $b$, it holds that $T_i\lceil_{\pi'} = b$.

Next we turn to a procedure of representing formulas by decision trees of small depth.

## 6 Basics for $t$-evaluations

The concept of $t$-evaluations was introduced by Krajíček et al. [KPW95] and is a very convenient tool for proving lower bounds on proof size. The content of this section is standard and we follow the presentation of Urquhart and Fu [UF96] while using the notation of Håstad [Hås20]. We need a generalization of previous notions essentially as introduced by Pitassi et al. [PRT22].

A tree $T$ represents $T_1 \vee \ldots \vee T_s$ if for every branch $\pi$ of $T$ ending in a leaf labeled 1 it holds that there is an $i \in [s]$ such that $T_i\lceil_\pi = 1$, and if $\pi$ ends in a leaf labeled 0, then for all $i \in [s]$ it holds that $T_i\lceil_\pi = 0$. The set of formulas $\Gamma$ has a $t$-evaluation $\varphi$, mapping formulas from $\Gamma$ to decision trees of depth at most $t$, if the following holds.

1. $\varphi$ maps constants to the appropriate decision tree of depth 0,
2. axioms are mapped by $\varphi$ to 1-trees,
3. if $\varphi(F) = T$ then $\varphi(\neg F)$ is a decision tree with the same topology as $T$ but where the value at each leaf is negated, and
4. if $F = \vee_{i\in[s]} F_i$, then $\varphi(F)$ represents $\vee_{i\in[s]}\varphi(F_i)$.

Each line of a proof has its own $t$-evaluation. In order to argue about the proof we need that these different $t$-evaluations are consistent, as explained next.

Let us first define what it means for decision trees to be consistent. Two decision trees $T_1, T_2$ are consistent if for every branch $\pi$ of $T_1$ ending in a leaf labeled $b$ it holds that $T_2\lceil_\pi = b$ and vice-versa. Further, $T_1$ and $T_2$ are $\neg$-consistent, if for every branch $\pi$ of $T_1$ ending in a leaf labeled $b$, it holds that $T_2\lceil_\pi = \neg b$ and vice-versa.

Let us say that two formulas are isomorphic if they only differ in the order of the or's, and let us say that two formulas $F, G = \neg G'$ are $\neg$-isomorphic if $F$ and $G'$ are isomorphic.

Consider a $t$-evaluation $\varphi$ defined over a set of formulas $\Gamma$ and similarly let $\varphi'$ be a $t$-evaluation defined over the set of formulas $\Gamma'$. The two $t$-evaluations $\varphi$ and $\varphi'$ are consistent if

1. for all isomorphic formulas $F \in \Gamma$ and $F' \in \Gamma'$ it holds that $\varphi(F)$ and $\varphi'(F')$ are consistent, and
2. for all $\neg$-isomorphic formulas $F \in \Gamma$ and $F' \in \Gamma'$ it holds that $\varphi(F)$ and $\varphi'(F')$ are $\neg$-consistent.

We say that a Frege proof has a $t$-evaluation if for every line $\nu$ in the proof we have a $t$-evaluation $\varphi^\nu$ and for all lines $\nu, \nu'$ it holds that $\varphi^\nu$ and $\varphi^{\nu'}$ are consistent.

Let us consider a Frege proof of depth $d$ and for a line $\nu$ in the proof let $\Gamma^\nu$ be the set of subformulas occurring on line $\nu$. In the following we construct a sequence of restrictions $\sigma_1, \sigma_2, \ldots, \sigma_d$ such that for every line and all formulas of depth at most $k$ we have consistent $t(k)$-evaluations if the formulas are hit by the concatenation $\sigma_k^*$ of the first $k$ restrictions in the sequence. When considering proof size we in fact have that all $t(k)$ are equal to the same value $t$, while in the proof when we lower bound the number of small lines, the value $t(k)$

grows as a function of $k$. In fact, in the latter situation, each line has a common part to all decision trees of that line and this common part increases in size with $k$.

Getting back to $t(k)$-evaluations, put different we build by induction on $k$ for every line $\nu$ a $t(k)$-evaluation for all formulas in

$$\Gamma_k^\nu = \{F\lceil_{\sigma_k^*} \mid F \in \Gamma^\nu \wedge \mathsf{depth}(F) \leq k\}$$

that are pairwise consistent and we look to extend these $t(k)$-evaluations to $\Gamma_{k+1}^\nu$. To make sure that the domain of the $t$-evaluations does not decrease when we apply a restriction we use the lemma below from [Hås20]. The fact that we allow consistent $t(k)$-evaluations, instead of a single $t(k)$-evaluation for the entire proof, does not change the proof which is a simple and fairly formal verification and hence omitted.

**Lemma 6.1.** *Let $\varphi$ and $\varphi'$ be two consistent $t$-evaluations respectively defined on the set of formulas $\Gamma$ and $\Gamma'$, and let $\sigma$ be a full restriction whose output is a grid of size $n$. Then, provided that $t < n/4$, $\varphi(F)\lceil_\sigma$ and $\varphi'(F)\lceil_\sigma$ are consistent $t$-evaluations whose domain includes $\Gamma\lceil_\sigma$, and $\Gamma'\lceil_\sigma$ respectively.*

The important step of the argument is to use a switching lemma to extend the domain of the $t(k)$-evaluation from $\Gamma_k^\nu$ to $\Gamma_{k+1}^\nu$. We give that argument in the next section and here we turn to formulating the punch line once we have a $t(k)$-evaluation for a small Frege proof, where we think of $t(k)$ as small.

It turns out that under these assumptions all lines in the proof are represented by 1-trees. As the the constant false is represented by a 0-tree we can thus not derive the desired contradiction. Hence in order to obtain the desired contradiction the Frege proof must be large, respectively long in the case of Frege proofs of bounded line size.

In order to formalize this argument we need to fix a Frege system so we can argue about the derivation rules. By a result of Cook and Reckhow [CR79] the precise choice of the Frege system is not important and we choose the same system as [PRST16, Hås20, PRT22]. This system consists of the following rules.

- (Excluded middle) $(p \vee \neg p)$
- (Expansion rule) $p \to (p \vee q)$
- (Contraction rule) $(p \vee p) \to p$
- (Association rule) $p \vee (q \vee r) \to (p \vee q) \vee r$
- (Cut rule) $p \vee q, \neg p \vee r \to q \vee r$

These rules should be understood in the following manner: a depth $d$ Frege proof can at any time, by excluded middle, write down a line of the form $(p \vee \neg p)$ for any formula $p$ if the line is of depth at most $d$. Similarly the expansion rule says that if we have derived the formula $p$, then we can write down the line $(p \vee q)$ for any formula $q$ such that the line is of depth at most $d$. The crucial lemma is as follows.

**Lemma 6.2.** *Suppose we have a derivation using the above rules starting from the Tseitin axioms defined on the $n \times n$ grid, that also has a $t$-evaluation. Then, if $t \leq n/8$, each line in the derivation is mapped to a 1-tree. This, in particular, implies that we cannot derive contradiction.*

The proof in the standard case of this lemma is again a tedious and formal verification and can be found in full in [Hås20]. The proof is by induction over the number of derivation steps and the key property is to take any path that leads to 0 in the derived formula and find

a path that leads to a 0 in one of the assumptions. The fact that all decision trees are of depth less than $n/8$ ensures that it is possible to find a branch of any decision tree that is consistent with the given 0-branch.

In the current case, where each line has its own $t$-evaluation, due to consistency, not much is different. We can again take any 0-branch in the decision tree of a derived formula and find a 0-branch in one of the assumptions. Instead of repeating all cases let us do only the most interesting one: the cut rule.

We have $F = (q \vee r)$ derived on line $\nu$ and suppose $\varphi^\nu(F)$ is not a 1-tree. Take a supposed leaf with label 0 in $\varphi^\nu(F)$ and let $\tau$ be the assignment leading to this leaf. We know that $\varphi^\nu(q)\lceil_\tau$ and $\varphi^\nu(r)\lceil_\tau$ are both 0-trees by the definition of a $t$-evaluation.

Now suppose $(p \vee q)$ was derived on line $\nu' < \nu$ and $(\neg p \vee r)$ was derived on line $\nu'' < \nu$. By consistency of $\nu$ and $\nu'$ we know that $\varphi^{\nu'}(q)\lceil_\tau$ is a 0-tree and, as also $\nu$ and $\nu''$ are consistent, so is $\varphi^{\nu''}(r)\lceil_\tau$.

Now, if any branch in $\varphi^{\nu'}(p)\lceil_\tau$ ends in a leaf labeled 0, then $\varphi^{\nu'}(p \vee q)\lceil_\tau$ can be extended to reach a 0-leaf. This is in contradiction to the inductive assumption. For similar reasons $\varphi^{\nu''}(\neg p)\lceil_\tau$ is a 1-tree. This contradicts the assumed consistency of $\nu'$ and $\nu''$.

# 7 Proofs of the main theorems

We first reprove the main theorem of [Hås20] with improved parameters.

**Theorem 7.1.** *For $d \leq O(\frac{\log n}{\log\log n})$ the following holds. Any depth-$d$ Frege refutation of the Tseitin contradiction defined on the $n \times n$ grid requires size*

$$\exp\left(\Omega(n^{1/(2d-1)}(\log n)^{O(1)})\right) \ .$$

As outlined in the previous section, we construct a $t$-evaluation for all sub-formulas occurring in a short and shallow Frege proof. By Lemma 6.2 we then conclude that all shallow Frege proofs of the Tseitin contradiction must be long. For the total size lower bound we in fact do not create distinct $t$-evaluations per line but rather a single one, used on each line. Such a $t$-evaluation is clearly consistent and hence satisfies our needs. Let $\Gamma$ denote the set of sub-formulas occurring in the alleged proof. Our plan is to proceed as follows for $i = 0, 1, 2, \ldots, d$.

- We have a $t$-evaluation for all formulas of $\Gamma$ that were originally of depth $i$.
- Pick a random full restriction $\sigma_i$ and extend the $t$-evaluation to all formulas of $\Gamma\lceil_{\sigma_i}$ of original depth at most $i+1$.

At the starting point, $i = 0$, each formula is a literal which is represented by a natural decision tree of depth 1. In order to extend the $t$-evaluation to larger depth we use the following lemma, central to the argument.

**Lemma 7.2** (Switching Lemma)**.** *There is a constant $A$ such that the following holds. Suppose there is a $t$-evaluation that includes $F_i, 1 \leq i \leq m$ in its domain and let $F = \vee_{i=1}^m F_i$. Let $\sigma$ be a random full restriction from the space of restrictions defined in Section 4. Then the probability that $F\lceil_\sigma$ cannot be represented by a decision tree of depth at most $s \geq t$ and the number of live variables in each center is in the interval $[.99C\log n, 1.01C\log n]$ is at most*

$$\left(A(\log n)^{27} t\Delta^{-1}\right)^{s/108} \ .$$

We postpone the proof of this lemma to Section 8 and see how to use it when studying a refutation of size $N$. We start with a $t_1$-evaluation with $t_1 = 1$ for single literals and apply the lemma with $s = \Omega(\log N)$ in the first step, while we choose $t_i = s$ in later steps. We set $\Delta_i = \Omega(t_i(\log n)^{27})$ and hence have that $T_i = 4\Delta_i^2$ for each step.

We start with the original Tseitin contradiction on the $n \times n$ grid. Start with $n_0 = n$ and set $n_{i+1} = n/T_i$ for $i = 0, 1, \ldots, d-1$. We are going to choose a sequence of full restrictions $\sigma_i$ mapping a grid of size $n_i$ to a grid of size $n_{i+1}$ randomly. Let $\sigma_i^*$ be the composition of $\sigma_0, \sigma_1, \ldots, \sigma_i$. Let $\Gamma$ be the set of sub-formulas that appear in an alleged proof and we let

$$\Gamma_i = \{F\restriction_{\sigma_i^*} \mid F \in \Gamma \wedge depth(F) \leq i\} \ .$$

Let $f_i$ be the number of sub-formulas of depth at most $i$ in $\Gamma$.

**Lemma 7.3.** *With probability $1 - f_i 2^{-\Omega(s)}$ there is a $t$-evaluation $\varphi_i$ whose domain includes $\Gamma_i$.*

*Proof.* This is essentially collecting the pieces. We prove the lemma by induction over $i$. For $i = 0$ we have the $t$-evaluation that maps each literal to its natural decision tree of depth 1.

When going from depth $i$ to depth $i+1$ we need to define $\varphi_{i+1}$ on all formulas originally of depth at most $i+1$ and consider any such $F$.

1. Each $F$ of depth at most $i$ is, by induction, in the domain of $\varphi_i$ and we can appeal to Lemma 6.1.
2. If $F$ is of depth $i$ then $\varphi_{i+1}(\neg F)$ is defined from $\varphi_{i+1}(F)$ negating the labels at the leaves.
3. For $F = \vee F_i$ where each $F_i$ is of at most depth $i$ we apply Lemma 7.2.

The only place where the extension might fail is under step three but, by Lemma 7.2, the probability of failure for any individual formula is at most $2^{-\Omega(s)}$ and as we have at most $f_i - f_{i-1}$ formulas of depth exactly $i$ the induction is complete. □

Fixing parameters we reprove the main theorem from [Hås20] with stronger parameters.

*Proof of Theorem 7.1.* Suppose we have a refutation of size $N \leq \exp(c_1(n^{1/(2d-1)}(\log n)^{-c_2}))$ for suitable positive constants $c_1$ and $c_2$. In the first iteration we use Lemma 7.2 with $t = 1$ and $\Delta = (2tA(\log n)^{27})^{-1}$ and $s = 110 \log N$. In later applications we use $t = s$. It is easy to see that with these numbers we have successful switching at each round with high probability. The number of live centers are in the desired interval and we are always able to construct the new $t$-evaluation.

Up to poly-logarithmic factors we have that the final side length of the grid after all the restrictions is $n(\log N)^{-2(d-1)}$ and it is a $t$-evaluation with $t = O(\log N)$. Thus if $\log N$ is a poly-logarithmic factor smaller than $n^{1/(2d-1)}$ we get a contradiction to Lemma 6.2. □

Let us turn our attention to the main result of the present paper.

**Theorem 7.4.** *For any Frege proof of the Tseitin principle defined over the $n \times n$ grid graph the following holds. If each line of the proof is of size $M$ and depth $d$, then the number of lines in the proof is*

$$\exp\left(\Omega\left(\frac{n}{\left((\log n)^{O(1)} \log M\right)^{2d}}\right)\right) \ .$$

The strategy of the proof is similar to the proof of Theorem 7.1: we again build a $t$-evaluation for a supposed Frege proof. The main difference is that instead of creating a single $t$-evaluation for the entire proof we in fact independently create $t$-evaluations for each line. These $t$-evaluations turn out to be consistent, as defined in Section 6, and we thus obtain the claimed bounds.

Suppose we are given a Frege refutation of the Tseitin principle defined over the $n \times n$ grid consisting of $N$ lines, where each line is a formula of size $M$ and depth $d$. We denote by $\Gamma^\nu$ the set of sub-formulas of line $\nu$ in the proof and continue to construct a sequence of restrictions $\sigma_1, \sigma_2, \ldots, \sigma_d$ such that all formulas of depth at most $k$ have consistent $t(k)$-evaluations if hit by the concatenation $\sigma_k^*$ of the first $k$ restrictions in the sequence, where $t(k)$ is some function dependent on $k$ to be fixed later. That is, for every line $\nu$ we have a $t(k)$-evaluation $\varphi_k^\nu$ for all formulas in the set

$$\Gamma_k^\nu = \{F\restriction_{\sigma_k^*} \mid F \in \Gamma^\nu \wedge \mathsf{depth}(F) \leq k\},$$

and all these $t(k)$-evaluations are consistent. In addition to these $t(k)$-evaluations, for each line $\nu$ we also maintain a decision tree $\mathcal{T}_k(\nu)$. We maintain the property that $\mathcal{T}_k(\nu)$ is a $t$ common partial decision tree for all $t(k)$-evaluations $\varphi_k^\nu(\Gamma_k^\nu)$ of bounded depth.

These common partial decision trees $\mathcal{T}_k(\nu)$ are useful to extend the $t(k)$-evaluations $\varphi_k^\nu$ to larger depths. In each such step, increasing $k$, we apply for each branch $\pi$ from $\mathcal{T}_k(\nu)$ the following multi-switching lemma to the set of decision trees $\varphi_k^\nu(\Gamma_k^\nu)\restriction_\pi$ of depth at most $t$. We then extend $\mathcal{T}_k(\nu)$ in each leaf $\pi$ by the the common partial decision tree from the lemma to obtain $\mathcal{T}_{k+1}(\nu)$ of slightly larger depth.

**Lemma 7.5** (Multi-switching Lemma)**.** *There are constants $A$, $c_1$, and $c_2$ such that the following holds. Consider formulas $F_i^j$, for $j \in [M]$ and $i \in [m_j]$, each associated with a decision tree of depth at most $t$ and let $F^j = \vee_{i=1}^{m_j} F_i^j$. Let $\sigma$ be a random full restriction from the space of restrictions defined in Section 4. Then the probability that the number of live variables in each center is in the interval $[.99C \log n, 1.01C \log n]$ and $(F^j\restriction_\sigma)_{j=1}^M$ cannot be represented by an $\ell$ common partial decision tree of depth at most $s$ is at most*

$$M^{s/\ell}\left(A(\log n)^{c_1} t \Delta^{-1}\right)^{s/c_2} .$$

We defer the proof of this lemma to Section 9. We apply Lemma 7.5 with mostly the same parameters so let us fix these. We choose $\ell = t = \log M$ and $\Delta = D \cdot t \cdot (\log n)^{c_1}$, for a sufficiently large constant $D$. The parameter $s$ depends on $k$ and is fixed to $s = s_k = 2^{k-1} \log N$. With these parameters in place we can finally also fix $t(k) = \sum_{i \leq k} s_i + \log M \leq 2^k \log N + \log M$.

**Lemma 7.6.** *Suppose that for every line $\nu \in [N]$ we have consistent $t(k-1)$-evaluations $\varphi_{k-1}^\nu$ for formulas in $\Gamma_{k-1}^\nu$ along with a $t$ common partial decision tree $\mathcal{T}_{k-1}(\nu)$ for $\varphi_{k-1}^\nu(\Gamma_{k-1}^\nu)$ of depth $\sum_{i<k} s_i$. Then, with probability $1 - N^{-1}$, there is a full restriction $\sigma_k$ whose output grid is of dimension $n$ and, assuming that $t(k) \leq n/8$, for every line $\nu \in [N]$ there is a consistent $t(k)$-evaluation $\varphi_k^\nu$ for formulas in $\Gamma_k^\nu$ and a $t$ common partial decision tree $\mathcal{T}_k(\nu)$ for $\varphi_k^\nu(\Gamma_k^\nu)$ of depth $\sum_{i \leq k} s_i$.*

*Proof.* Let us first extend the common partial decision trees and then explain how to obtain $\varphi_k^\nu$ for different lines $\nu \in [N]$.

The interesting formulas of original depth $k$ to consider are the ones with a top $\vee$ gate. Let us fix a line $\nu \in [N]$ and consider all sub-formulas $\{F^j = \vee_{i=1}^{m_j} F_i^j\}_{j=1}^{M_\nu}$ of line $\nu$ of original depth $k$ with a top $\vee$ gate under the restriction $\sigma_{k-1}^*$. As the original depth of every $F_i^j$ is at most $k-1$, all these formulas are in the domain of $\varphi_{k-1}^\nu$. Let us further fix a path $\pi$ in $\mathcal{T}_{k-1}(\nu)$ and recall that all decision trees $\varphi_{k-1}^\nu(F_i^j)\restriction_\pi$ are of depth at most $t$.

For every $\nu \in [N]$ and branch $\pi$ of $\mathcal{T}_{k-1}(\nu)$ we apply Lemma 7.5 to the set of formulas $F_i^j \restriction_\pi$ with associated trees $\varphi_{k-1}^\nu(F_i^j)\restriction_\pi$ of depth at most $t$. The probability of failure of a single application is bounded by $N^{-2^k-1}$, assuming an appropriate choice of the constant $D$. As we invoke Lemma 7.5 at most $N \cdot 2^{\sum_{i<k} s_i} \leq N^{2^k}$ times, by a union bound, with probability at least $1 - N^{-1}$, there is a full restriction $\sigma_k$ such that for every line $\nu \in [N]$ and every branch $\pi \in \mathcal{T}_{k-1}(\nu)$ we get a $t$ common partial decision tree of depth at most $s_k$ for the formulas $(F^j \restriction_{\pi\sigma_k})_{j=1}^{M_\nu}$. Let us denote this common decision tree by $\mathcal{T}(\nu, \pi)$ and attach it to $\mathcal{T}_{k-1}(\nu)$ at the leaf $\pi$ to obtain $\mathcal{T}_k(\nu)$. The trees $\mathcal{T}_k(\nu)$ are of depth at most $\sum_{i \leq k} s_k$ as required.

Let us explain how to define $\varphi_k^\nu$ for a fixed line $\nu \in [N]$. Consider any formula $F$ in $\Gamma_k^\nu$.

- If $F$ is of depth less than $k$, then $F$ is in the domain of $\varphi_{k-1}^\nu$ and we can appeal to Lemma 6.1.
- If $F$ is of depth $k-1$ then $\varphi_k^\nu(\neg F)$ is defined from $\varphi_k^\nu(F)$ negating the labels at the leaves.
- For $F = \vee_i F_i$ of depth $k$ we use the previously constructed common partial decision trees. We define $\varphi_k^\nu(F)$ to be the decision tree whose first $\sum_{i \leq k} s_i$ levels are equivalent to $\mathcal{T}_k(\nu)$ followed by $t$ levels unique to $F$ obtained from the multi-switching lemma.

Let us check that the decision trees $\mathcal{T}_k(\nu)$ are indeed $t$ common partial decision trees for $\varphi_k^\nu(\Gamma_k^\nu)$. By construction this clearly holds for formulas of depth $k$ with a top $\vee$ gate. As $\mathcal{T}_k(\nu)$ is equivalent to $\mathcal{T}_{k-1}(\nu)$ on the upper levels, and restrictions only decrease the depth of decision trees, by the initial assumptions this also holds for formulas of depth less than $k$. As the $t(k)$-evaluations of formulas of depth $k$ with a top $\neg$-gate are defined in terms of formulas of depth less than $k$, we also see that $\mathcal{T}_k(\nu)$ is a $t$ common partial decision tree for such formulas.

Last we need to check that each $\varphi_k^\nu$ is a $t(k)$-evaluation plus that these are pairwise consistent.

By Lemma 6.1 all the properties hold for formulas of depth less than $k$. Let us verify the $t(k)$-evaluation properties for formulas of depth $k$.

Property 1 is immediate, as $k > 0$. As we only consider consistent decision trees, property 2 also follows. Further, property 3 is satisfied by construction. Property 4 can be established by checking the property for each branch $\pi$ in $\mathcal{T}_{k-1}(\nu)$ separately; for a fixed $\pi$ we see by Lemma 7.5 that this indeed holds.

Finally we need to establish that two $t(k)$-evaluations $\varphi_k^\nu$ and $\varphi_k^{\nu'}$ are consistent for formulas of depth $k$. By the inductive hypothesis we clearly have that $\neg$-isomorphic formulas are $\neg$-consistent. Further, isomorphic formulas with a top $\neg$ gate are consistent. Hence we are only left with checking consistency for isomorphic formulas of depth $k$ with a top $\vee$ gate.

Let $F = \vee_i F_i$ and $F' = \vee_i F_i'$ be two isomorphic formulas from $\Gamma_k^\nu$ and $\Gamma_k^{\nu'}$ respectively. For the sake of contradiction suppose $\varphi_k^\nu(F)\restriction_\pi = 1$ but $\varphi_k^{\nu'}(F')\restriction_\pi = 0$ for some assignment $\pi$. In the following we use that $t(k) \leq n/8$ and hence there are consistent branches as claimed. By property 2 we know that for some $F_i$ it holds that $\varphi_k^\nu(F_i)\restriction_\pi = 1$. As $F$ and $F'$ are isomorphic formulas we know that there is an $F_j'$ such that $F_i$ and $F_j'$ are isomorphic formulas. As such formulas have consistent decision trees (by induction and Lemma 6.1) we get that $\varphi_k^{\nu'}(F_j')\restriction_\pi = 1$. But this cannot be as by property 4 of a $t(k)$-evaluation this implies that $\varphi_k^{\nu'}(F')\restriction_\pi = 1$. This establishes that the different $t(k)$-evaluations are consistent, as required. □

With all pieces in place we are ready to prove Theorem 7.4.

*Proof of Theorem 7.4.* Suppose we are give a proof of length $N = \exp(n/((\log n)^c \log M)^{2d})$, for some constant $c$. We may assume that $M \leq \exp(n^{1/2d-1/2d(2d-1)})$, as otherwise we can apply Theorem 7.1.

In order to create the consistent $t(k)$-evaluations $\varphi^\nu$ for each line $\nu \in [N]$ we consecutively apply Lemma 7.6 $d$ times. We start with $\varphi_0^\nu$ which maps constants to the appropriate depth 0 decision tree and literals to the corresponding depth 1 decision trees. The common partial decision trees $\mathcal{T}_0(\nu)$ are all empty.

After applying Lemma 7.6 $d$ times we are left with a $t(d)$-evaluation for the proof. We need to ensure that $t(d)$ is upper bounded by the dimension of the final grid: $t(d) \leq 2^d \log N + \log M$, while the final side length of the grid is $n \cdot (4\Delta^2)^{-d} = n \cdot (2D(\log n)^{c_1} \log M)^{-2d}$. For our choice of $N$ and the assumption on $M$ this indeed holds and by Lemma 6.2 the theorem follows. □

# 8 The improved standard switching lemma

This section is dedicated to the proof of the switching lemma, restated here for convenience.

**Lemma 7.2** (Switching Lemma)**.** *There is a constant $A$ such that the following holds. Suppose there is a $t$-evaluation that includes $F_i, 1 \leq i \leq m$ in its domain and let $F = \vee_{i=1}^m F_i$. Let $\sigma$ be a random full restriction from the space of restrictions defined in Section 4. Then the probability that $F\lceil_\sigma$ cannot be represented by a decision tree of depth at most $s \geq t$ and the number of live variables in each center is in the interval $[.99C \log n, 1.01C \log n]$ is at most*

$$\left(A(\log n)^{27} t \Delta^{-1}\right)^{s/108} .$$

The proof very much follows the proof of [Hås20]. In fact large parts of the proof are the same. We repeat the proofs to make it possible for a reader not familiar with the mentioned proof to follow the argument. To make the argument slightly shorter we do not repeat all proofs of the various lemmas.

For the benefit of the reader completely on top of [Hås20] let us outline the differences in the following section. This section can be safely skipped by the less experienced reader.

## 8.1 Changes in the Argument

The key number that has changed is the parameter $k$, the total number of centers that are alive. In the definition of a partial restriction this parameter $k$ has changed from $Cs(n/T)^2$ to $C \log n(n/T)^2$. The fact that we had $\Omega(s)$ live centers in each square was crucial in finding live centers to extend the information sets $J_j$ to closed sets $\gamma_j$. This process needed $O(s)$ fresh centers from specific squares and there is nothing that prevents these from all being required to be in the same square. In the current proof we allow $\gamma_j$ to be not closed and this implies that the restriction $\rho^*$ is a generalized restriction where the Tseitin condition is violated at some vertices. This only happens when we have $\Omega(\log n)$ exposed non-chosen centers in a sub-square and results in a single violating vertex. As the there are at most $O(s)$ exposed centers over all we can have at most $O(s/\log n)$ violating centers. The number of generalized restrictions with $B$ violating centers is at most a factor $n^{2B}$ more than the the number of ordinary restrictions. This number is $2^{O(s)}$ and this factor can be absorbed in the constant $A$ in the statement of the switching lemma.

## 8.2 Proof Overview

Let us recall the setup. We have a full restriction $\sigma = \sigma(\rho, \pi)$ as defined in Section 4 that is made up of a restriction $\rho$ and a pairing $\pi$. The restriction $\rho$ has $(1 \pm 0.01)C \log n$ many live centers in each sub-square, for a large enough constant $C$. We have a formula $F = \vee_{i=1}^{m} F_i$ and a $t$-evaluation $\varphi$ that includes each $F_i$ in its domain and let $T_i = \varphi(F_i)$. As $\varphi$ is a $t$-evaluation each such tree $T_i$ is of depth at most $t$.

In the following we construct a decision tree $\mathcal{T}$ for $F\lceil_\sigma$ which is with high probability, over the choice of $\rho$, of depth at most $s$. The decision tree $\mathcal{T}$ is created in a similar manner as the canonical decision tree is usually constructed: we proceed in stages, where in each stage the current branch $\tau$ is extended by querying variables related to the first 1-branch $\psi$ in the trees $T_1\lceil_{\sigma\tau}, T_2\lceil_{\sigma\tau}, \ldots, T_m\lceil_{\sigma\tau}$. For now it is not so important what the related variables of $\psi$ precisely are and we can simply think of these as the variables on the branch $\psi$. Once all these variables have been queried, we check in each new leaf of the tree whether we traversed the path $\psi$. If so, then we label the leaf with a 1 and otherwise we continue with the next stage. If there are no 1-branches left, we label the leaf with a 0.

It is not so hard to see that this process indeed results in a tree $\mathcal{T}$ that represents $\vee_{i=1}^{m} T_i\lceil_\sigma$: for each leaf $\tau$ of $\mathcal{T}$ that is labeled 1 it holds that there is an $i \in [m]$ such that $T_i\lceil_{\sigma\tau} = 1$ and if $\tau$ is labeled 0, then for all $i \in [m]$ we have that $T_i\lceil_{\sigma\tau} = 0$, as required. It remains to argue that $\mathcal{T}$ is with high probability of depth at most $s$.

We analyze this event using the labeling technique of Razborov [Raz95]. The idea of this technique is to come up with an (almost) bijection from restrictions $\rho$ that give rise to a decision tree $\mathcal{T}$ of depth larger than $s$ to a set of restrictions that is much smaller than the set of all restrictions. In a bit more detail, given such a bad $\rho$, we create a restriction $\rho^*$ with fewer live centers such that with a bit of extra information we can recover $\rho$ from $\rho^*$. As the restriction $\rho^*$ has roughly $s$ fewer live centers than $\rho$, and the inversion requires little extra information, we obtain our statement.

Let us explain how to obtain $\rho^*$ from a $\rho$ that gives rise to a decision tree $\mathcal{T}$ of depth larger than $s$. To this end, we first need to slightly refine the construction process of $\mathcal{T}$. Namely, we need to discuss what the related variables of a branch $\psi$ are. Instead of thinking of this as a set of variables we rather want to think of it as an information set $J$, as introduced in Section 4. The information set $J$ is a minimal set that forces, along with the already collected information set on the branch $\tau$, the branch $\psi$. Once we identified such a set $J$, we then query all necessary variables to see whether we agree with $J$ (along with some further variables).

Recall that we are trying to explain how to construct $\rho^*$ from a $\rho$ that gives rise to a decision tree $\mathcal{T}$ of large depth. Fix a long branch $\tau$ in $\mathcal{T}$ and consider all the sets $J_1, J_2, \ldots, J_g$ identified on $\tau$. For this proof overview, let us assume that each $J_j$ is closed and the support of these information sets are pairwise disjoint. Let us stress that this is a simplification and does not hold in general. Assuming this holds, note that the union $J^* = \cup_{i=1}^{g} J_j$ is also closed and recall from Section 4 that all variables forced by $(\rho, J^*)$ can be described by a restriction where the centers in $\mathrm{supp}(J^*)$ are killed. This defines the restriction $\rho^*$: it is the restriction that forces all variables forced by $(\rho, J^*)$. Assuming that the support of $J^*$ is large, we see that $\rho^*$ has much fewer centers that are alive.

What remains is to argue that we can cheaply recover $\rho$ from $\rho^*$. The idea is to remove the set $J_j$, starting with $j = 1$, one-by-one from $\rho^*$. To do this cheaply we use the decision trees $T_1, \ldots, T_m$. Recall that the information set $J_1$ determines all variables on the first 1-branch $\psi_1$. This implies in particular that $\rho^*$ traverses the branch $\psi_1$. Hence identifying $\psi_1$ is for free: it is the first 1-branch in $T_1, \ldots, T_m$ traversed by $\rho^*$ (assuming that the set $J_1$ is pairwise disjoint from all later sets $J_j$). Once we identified the branch $\psi_1$, we want to recover the first part of

the long branch $\tau$ so that we can repeat this argument with $J_2$. As $\psi_1$ is of length at most $t$, using only $\log t$ bits per variable, we indicate which variables are different on $\tau$ from $J_1$. This lets us cheaply recover $\tau$ along with the centers killed by $J_1$. Repeating this argument $g$ times lets us recover $\rho$.

This completes the proof overview. We allowed ourselves several simplifications and left out a fair number of details. The most significant simplification is the assumption that all the information sets $J_j$ are closed. In the actual proof we extend each set $J_j$ into a closed set $\gamma_j$ and then take the union of these to define $\rho^*$. The process of closing a set $J_j$ may even fail at times and therefore $\rho^*$ has to slightly bend the rules of being a restriction. It turns out that $\rho^*$ is a generalized restriction as mentioned in Section 4.3. The step of closing up the $J_j$ is the main source of technical difficulty in the full proof.

The proof is split into four separate sections. In Section 8.3 we define the extended canonical decision tree $\mathcal{T}$ and in the subsequent Section 8.4 we prove some crucial properties of these decision trees. Section 8.5 explains how to extend the sets $J_j$ into closed information sets $\gamma_j$ in order to construct the restriction $\rho^*$. Finally, in Section 8.6 we show how to cheaply recover $\rho$ from $\rho^*$ and thereby prove Lemma 7.2.

### 8.3 Extended Canonical Decision Trees

Let us construct an *extended canonical* decision tree $\mathcal{T}$ for $F\lceil_\sigma$. We start with $\mathcal{T}$ the empty tree and extend it for each branch $\tau$ separately. For every branch $\tau$ we maintain the following objects throughout the creation of $\mathcal{T}$:

1. a set $S = S(\tau, \sigma)$ of centers, called the *exposed centers*,
2. a set $I = I(\tau, \sigma)$ of information pieces as defined in Definition 4.5, and
3. a (state of a) matching game $\mathcal{G} = \mathcal{G}(\tau, \sigma)$, as described in Section 3, played on the chosen centers of $\sigma$.

Initially the sets $S$ and $I$ are empty, and the matching game $\mathcal{G}$ is a new game with no vertices matched. We require that $S$, $I$ and $\mathcal{G}$ satisfy the following invariants.

1. No element is ever removed from $S$ or $I$. In other words, the sets $S$ and $I$ only become larger throughout the creation of a branch $\tau$.
2. The matched nodes in the game $\mathcal{G}$ are precisely the chosen centers in $S$.
3. The information set $I$ does not contain a path between a chosen center and a non-chosen center.
4. For non-chosen centers in $S$, the set $I$ consists of the closed information pieces corresponding to their component in $\pi$ (both edges and non-edges). If one center of such a connected component belongs to $S$, then so does the entire component. Thus for non-chosen centers in $S$ we have information pieces in all four directions.
5. For every chosen center in $S$ we have queried all incident variables $x_P$ in $\tau$ and this is the information that is present as information pieces in $I$. The one-answers are recorded in the form of a path while the zero answers as two non-edges, one at the neighboring chosen center in the appropriate direction which may or may not be an element of $S$. Observe that the value of $x_P$ jointly with $\rho$ determines the value of all variables $x_e$ on the chosen path $P$.

Let us stress the fact that information about $\pi$ comes from the restriction $\sigma$ and hence in Invariant 4 we do not query a variable in $\mathcal{T}$. However, querying a variable $x_P$, as done in Invariant 5, causes a query in the decision tree $\mathcal{T}$.

Further, observe that there is a crucial difference between Invariant 4 and Invariant 5: on the non-chosen centers we have information pieces in $I$ only on the centers in $S$. In contrast $I$ may contain information pieces from chosen centers that are not in $S$.

Let us discuss the creation of $\mathcal{T}$. We proceed in stages. In each stage we fix a branch $\tau$ in $\mathcal{T}$. We go over the decision trees $T_i = \varphi(F_i)$ one by one. Suppose we consider $T_i$. Take the first (in some fixed order) branch $\psi$ in $T_i$ that leads to a leaf labeled 1 which is consistent with $\tau$ and $\sigma$. If there is no such branch, then we continue with $T_{i+1}$ and if there is no such branch $\psi$ for any $T_i$, then we label the $\tau$ leaf of $\mathcal{T}$ by 0 and continue with a different branch $\tau'$ of $\mathcal{T}$ until all leaves of $\mathcal{T}$ are labelled. But for now let us assume that there is a branch $\psi$ as described.

For the variables appearing on $\psi$ we have unique values required to reach this leaf. We let a *possible forcing information* $J$ be an information set that jointly with $I$ and $\rho$ forces[4] all variables on $\psi$ to take these unique values. Let us call $\psi$ the *forceable branch*. The intuition is that if the information set $J$ agrees with the actual input, then indeed $\psi$ is followed and we can safely end with a 1-leaf. In most cases, however, the actual input does not agree with $J$ and we need to continue evaluating the extended canonical decision tree $\mathcal{T}$. We require the following properties of $J$.

1. If $J$ contains a non-edge from a chosen center it also contains a non-edge in the "reverse direction". As an example if it contains a non-edge going left from a chosen center $v$ then it contains a non-edge going right from the chosen center in the sub-square to the left of $v$.

2. The information set $J$ does not contain a path between a chosen center and a non-chosen center.

3. The information sets $I$ and $J$ are consistent and disjoint.

4. The part of $J$ on the non-chosen centers is closed and consistent with $\pi$, that is, $J$ contains a subset of the components of $\pi$ in the form of a closed set of information pieces.

5. $J$ is minimal given the above properties and the fact that, along with $I$, it should determine the values of all the variables on the forceable branch $\psi$.

Note that a set $J$ may not be unique for a given path $\psi$. If there are several sets as described above, choose one in a fixed but otherwise arbitrary manner. While the choice is not essential for what follows, we do need to establish that whenever some $T_i$ can still reach a 1-leaf, then there is a possible forcing information $J$. We postpone this to the following section (see Lemma 8.1) and for now assume that such a set $J$ exists whenever we have a branch $\psi$ as described.

Denote by $U$ the set of closest endpoints of variables on $\psi$ that are chosen centers but not contained in $S$. A somewhat subtle point to note is that $U$ may contain a closest endpoint of a variable that is determined by $I$: the set $I$ may contain information pieces about chosen centers outside the set of exposed centers $S$. The set $U$ is needed to ensure that we treat such centers correctly.

Let us continue the construction of the extended canonical decision tree $\mathcal{T}$ at $\tau$. Add $U$ and all centers in $\mathrm{supp}(J)$ to $S$ along with the centers described next. Let the adversary in the

[4]Recall from Definition 4.6 that a variable is forced if we have the relevant information at its closest endpoint.

game $\mathcal{G}$ supply $U$ along with all chosen centers in $\mathrm{supp}(J)$. We apply Lemma 3.5 and add all nodes provided by $P_M$ to $S$ (we tacitly assume throughout that $|S| \leq n/2$). Observe that this game is played on nodes of the grid and does not take into account any other information from $I$ or $J$.

Finally we need to update $I$ and extend $\mathcal{T}$. This is straightforward for the non-chosen centers added to $S$: for every such non-chosen center $v$ we add the information from $v$'s connected component in $\pi$ to $I$ (in the form of edges and non-edges).

For every chosen center added to $S$ we query all the incident variables, thereby extending $\mathcal{T}$. For every newly created consistent extension $\tau'$ of $\tau$ we need to update the set $I$. Record one-answers as an edge and zero-answers as two non-edges including the other endpoint of a potential chosen path, i.e., the chosen center in the adjacent sub-square in the given direction. Recall that we only consider consistent branches $\tau'$ (as assignments) and hence we create consistent information sets.

Finally, for every consistent $\tau'$ extending $\tau$, we check whether the information set $I(\tau', \sigma)$ traversed the forceable branch $\psi$ of $T_i$. This can clearly be done: all variables on $\psi$ have their closest endpoint in $S$ and each exposed center has information pieces in all four directions. If $\psi$ is indeed followed, we label the leaf $\tau'$ with a 1. Otherwise, if the forceable branch is not followed, then we proceed with the next stage.

This completes the description of the creation of the extended canonical decision tree $\mathcal{T}$ for $F\restriction_\sigma$. It is straightforward to check that the invariants hold after every completed stage.

### 8.4 Some Properties of Extended Canonical Decision Trees

In this section we prove two important properties of extended canonical decision trees, along with some auxiliary lemmas. The first important property is that the decision tree $\mathcal{T}$ does indeed represent $\vee_{i=1}^{m} T_i\restriction_\sigma$. Secondly, we show that the construction process of $\mathcal{T}$ is independent of the choice of the negations of the preferred values along the paths between chosen centers. This allows us to focus on long branches that have well-behaved information sets $I$.

Before proving these two statements, recall that we postponed the proof of the claim that if it is possible to each a 1-leaf of $T_i$, then there is a possible forcing information $J$. Let us establish this fact. Observe that at any point when forming the extended canonical decision tree, the information $I$ comes from information in $\pi$ and from queries already done in the decision tree $\mathcal{T}$ with answers $\tau$. Remember that $\sigma$ includes all the information from $\pi$.

**Lemma 8.1.** *If there is a* 1*-branch* $\psi$ *in* $T_i\restriction_\sigma$ *that is consistent with* $\tau$*, then there is a possible forcing information* $J$ *for* $\psi$*.*

*Proof.* Let $\psi'$ be the branch in $T_i$ that gives rise to $\psi$. Consider the assignment $\tau'$ to the path variables $x_P$ such that the 1-leaf of $T_i\restriction_\sigma$ is reached. Let us find a possible $J$ such that $\psi'$ is followed.

The information pieces next to chosen centers are simply those given by $\tau'$. These are, by definition, consistent with $\tau$ and can hence be included in $J$.

The information pieces next to non-chosen centers are the relevant information pieces from $\pi$. As all information pieces from $\pi$ are consistent, consistency is automatically satisfied for these pieces.

Dropping any non-required piece and all the pieces already in $I$ makes $J$ disjoint from $I$ and minimal. Clearly $J$ forces $\psi'$ to be followed. This completes the proof of the lemma. □

As an immediate corollary we have that the decision tree $\mathcal{T}$ is indeed a legitimate choice for $\varphi(F\restriction_\sigma)$.

**Corollary 8.2.** *The extended canonical decision tree $\mathcal{T}$ represents $\vee_{i=1}^{m} T_i\lceil_\sigma$.*

The creation of the extended canonical decision tree depends on $\rho$ and $\pi$ but not, in a serious way, on the negations of the preferred values along the paths between the chosen centers. The following lemma makes this intuition precise.

**Lemma 8.3.** *Let $\sigma_1$ be obtained from $\rho_1$ and $\pi$ and $\sigma_2$ from $\rho_2$ and $\pi$ where $\rho_1$ and $\rho_2$ pick the same set of centers and fixed values. Assume furthermore that the only difference between $\rho_1$ and $\rho_2$ is that for each chosen path $P$ there is a bit $c_P$ such that for each grid-edge $e$ on $P$ the preferred values of $x_e$ differ by $c_P$ in $\rho_1$ and $\rho_2$. Then the only difference between the extended canonical decision trees of $F\lceil_{\sigma_1}$ and $F\lceil_{\sigma_2}$ is the labeling of the internal edges.*

*Proof.* This follows by inspection of the procedure for forming the extended canonical decision tree. The only difference is that variables on chosen paths in one case are forced by a path and in the other case by two non-edges. This does not cause any difference in the construction of $\mathcal{T}$ as the supports of the two corresponding sets $J_1$ and $J_2$ are identical by Property 1 of a possible forcing information. □

This lemma is crucial in our analysis. It allows us to focus on long branches whose information set $I$ is well-behaved in the following sense.

**Definition 8.4** (closed branch)**.** Let $\mathcal{T}$ be an extended canonical decision tree. A branch $\tau$ in $\mathcal{T}$ is *closed* if the information set $I(\tau, \sigma)$ contains a path between two chosen centers $u, v$ if and only if the matching game $\mathcal{G}(\tau, \sigma)$ matched $u$ to $v$.

This slightly overloads the notion "closed" but as the information pieces given by the answers on a closed branch $\tau$ is (essentially) a closed information set we hope that this causes no confusion. The following lemma is an immediate consequence of Lemma 8.3.

**Lemma 8.5.** *If the probability that $F\lceil_\sigma$ needs a decision tree of depth $s$ is at least $q$, then the probability that the extended canonical decision tree of $F\lceil_\sigma$ contains a closed branch of length at least $s$ is at least $2^{-s}q$.*

This lemma allows us to only analyze closed branches. The main advantage of considering closed branches is that the information sets $I$ have a nice structure. We use the following property throughout the proof.

**Lemma 8.6.** *On a closed branch, after the completion of a stage, $I$ consists of a closed part on the exposed vertices $S$ jointly with a set of non-edges from chosen centers not in $S$ towards chosen centers in $S$.*

*Proof.* The information in $I$ about non-chosen centers in $S$ is from $\pi$ and thus by definition closed. Further, because we are on a closed branch, the set $I$ is also closed on the chosen centers in $S$. The only other information pieces in $I$ are non-edges from chosen centers not in $S$ towards chosen centers in $S$. □

Lastly we have an auxiliary lemma regarding the size of the set of exposed centers $S$.

**Lemma 8.7.** *In each stage at most $8t$ vertices are added to the set of exposed vertices $S$.*

*Proof.* A forceable branch $\psi$ is of length at most $t$ as the trees $T_i$ are of depth at most $t$. For each variable $x_e$ on $\psi$ there are at most 2 chosen centers in $\mathrm{supp}(J) \cup U$ if the closest endpoint of $x_e$ is chosen and at most 4 non-chosen centers if the closest endpoint is non-chosen.

When adding $\mathrm{supp}(J) \cup U$ to $S$ we add at most 1 extra center per chosen center in $\mathrm{supp}(J) \cup U$ to $S$. We conclude that at most $8t$ vertices are added to $S$ in a given stage. □

### 8.5 From $\rho$ to $\rho^*$

We want to bound the number of restrictions $\rho$ (as defined in Section 4) that give rise to an extended canonical decision tree $\mathcal{T}$ of depth at least $s$. In light of Lemma 8.5 we can focus on $\mathcal{T}$ that contain a *closed* branch $\tau$ of length at least $s$. Let us fix such a $\rho$ along with the extended canonical decision tree $\mathcal{T}$ and the closed branch $\tau$ of length at least $s$.

The goal of this section is to construct a restriction $\rho^*$ that is related to $\rho$ but has fewer live variables. In the following section we then show how to recover $\rho$ from $\rho^*$ with a bit of extra information. As $\rho^*$ has fewer live variables there are fewer such restrictions and, assuming we require only little extra information to recover $\rho$, we thus establish that there are very few $\rho$ that cause the extended canonical decision trees to be of depth at least $s$.

Recall that an information set $\gamma^*$ is closed if for every center $v$ in $\mathrm{supp}(\gamma^*)$ the set $\gamma^*$ contains information in all four directions of $v$ and, furthermore, $\gamma^*$ has an odd number of edges incident to every such $v$. The idea is to reduce the number of live variables with the help of a closed information set $\gamma^*$. Consider all variables forced by $(\rho, \gamma^*)$. Observe that $(\rho, \gamma^*)$ can be described by a restriction $\rho^*$ where all the centers in $\mathrm{supp}(\gamma^*)$ are killed: negate the values of any preferred variable on any path in $\gamma^*$. In the following we are going to construct a closed information set $\gamma^*$ with large support (linear in $s$) such that $\rho$ can be recovered from the resulting $\rho^*$ with a bit of extra information.

As suggested in the proof outline, we would like to choose $\gamma^*$ to be the union of all the possible forcing information sets used when creating the long branch $\tau$. Unfortunately this does not work: a possible forcing information is not always closed and insisting on a possible forcing information to be closed creates a dependence between $\mathcal{T}$ and the negations of the preferred values along paths between chosen centers. As such it becomes difficult to prove the crucial Lemma 8.5.

So it is not obvious how to guarantee that the possible forcing information is closed. What we can do, however, is to close these information sets *after* we have found a long closed branch $\tau$. We can then take $\gamma^*$ to be the union of these newly closed possible forcing information sets. Let us proceed by explaining how to close the possible forcing information sets.

As $\tau$ is a branch of length $s$, there is a first stage $g$ such that at the end of stage $g$ at least $s/4$ centers are exposed: only variables incident to exposed centers are queried and each exposed center causes at most 4 queries on the branch $\tau$. Put different, if we let $\tau_g \subseteq \tau$ be the closed path constructed by the end of stage $g$, then the set of exposed centers $S^*_g = S(\tau_g, \sigma)$ is for the first time of size at least $s/4$. We analyze the event of ever reaching such a stage $g$.

Note that $|S^*_g| < s/4 + 8t$ by Lemma 8.7 and $g \leq s/4$ as in each stage at least one center is added to the exposed centers $S$. For $j \in [g]$ we let the forceable branch of stage $j$ in the decision tree $T_{i_j}$ be denoted by $\psi_j$, let $J_j$ be the corresponding possible forcing information and $\tau_j \subseteq \tau_g$ be the branch in $\mathcal{T}$ created by the end of stage $j$. Denote the information set added at stage $j$ by $I_j$ and let $I^*_j = I^*(\tau_j, \sigma)$, or equivalently $I^*_j = \cup_{i=1}^{j} I_i$, be the information set gathered during the first $j$ stages. In the following we explain how to extend the information sets $J_j$ into (usually) closed sets $\gamma_j$. Sometimes this extension may fail to produce a closed set $\gamma_j$ but this happens rarely and hence enough centers are killed in $\rho^*$ to finish the argument.

Consider the sets $J_1, \ldots, J_g$ in order. Initially we set $\gamma_j = J_j$ and extend it as follows. Recall that when we create the extended canonical decision tree $\mathcal{T}$, in stage $j$, we add a set $U_j$ of chosen centers to $S^*_{j-1} = S(\tau_{j-1}, \sigma)$ that are closest endpoints of variables on $\psi_j$. Add all information pieces in $I^*_{j-1}$ incident to a chosen center in $U_j$ to $\gamma_j$. Note that because $\tau_g$ is a closed branch and $U_j$ is disjoint from $S^*_{j-1}$, by Lemma 8.6, all these added information pieces are non-edges towards chosen centers in $S^*_{j-1}$.

We need to close the set $\gamma_j$. Let us consider each center $v \in \mathrm{supp}(\gamma_j)$ separately. We

want to close $\gamma_j$ at $v$, meaning that (1) there are information pieces in all directions next to $v$ and (2) an odd number of these edges are present. Note that the non-chosen part in $\gamma_j$ is already closed as this part is closed in $J_j$. Hence we only need to add information pieces next to chosen centers and we thus focus on the case when $v$ is a chosen center. We claim that if $v$ has information pieces in all four directions in $\gamma_j$, then $\gamma_j$ is closed at $v$: since $I^*_{j-1}$ and $J_j$ are consistent (by Property 3) there is an odd number of edges next to $v$.

Otherwise, if $v$ has no information piece in some direction(s), add a non-edge in all but one such direction to $\gamma_j$. In case $v$ already has an odd number of edges next to $v$, add another non-edge in the final direction. Else we need to add an edge to an appropriately selected center in the suitable sub-square $R$. At this point we slightly bend the rules and allow to connect the chosen center $v$ to a non-chosen center in $R$.

Namely, we add an edge from $v$ to a so-called *fresh center* in $R$, unless there are no fresh centers available. A fresh center is a non-chosen but alive center that is not a member of $S^*_g$ and is not an element of any of the sets $\operatorname{supp}(\gamma_1), \ldots, \operatorname{supp}(\gamma_{j-1})$. If we add a fresh center we also add non-edges from the fresh center in the other three directions, ensuring that $\gamma_j$ is closed. Let us emphasize that we choose which fresh centers to add to $\gamma_j$ *after* the long branch $\tau_g$ has been constructed. This allows us to ensure that these centers do *not* appear in $S^*_g$.

If there is no such fresh center available in $R$, then we do not add anything and let $\gamma_j$ have a center of even degree. Let us call these centers *bad*. This completes the description of the construction of the sets $\gamma_1, \ldots, \gamma_g$.

In the following we want to argue that the union of the different $\gamma_j$ is closed if we disregard the bad centers. We establish this by arguing that the $\gamma_j$ have pairwise disjoint supports.

**Lemma 8.8.** *For $j \neq j'$ it holds that $\operatorname{supp}(\gamma_j) \cap \operatorname{supp}(\gamma_{j'}) = \emptyset$.*

*Proof.* Let us assume that $j' < j$. By definition (Property 3) $J_j$ and $I^*_{j-1}$ are disjoint but their supports may intersect. As $I^*_{j-1}$ contains information pieces in all directions of every center in $S^*_{j-1}$ (Invariants 4 and 5), the supports of $J_j$ and $I^*_{j-1}$ can only intersect in centers that are not in $S^*_{j-1}$. Because the support of $J_{j'}$ was added to the set of exposed centers at the end of stage $j'$ we have that $\operatorname{supp}(J_{j'}) \subseteq S^*_{j-1}$. This implies that $\operatorname{supp}(J_{j'})$ does not intersect $\operatorname{supp}(J_j) \cup U_j$ as $U_j$ is disjoint from $S^*_{j-1}$ by definition.

Further, because $U_{j'}$ is a subset of $S^*_{j-1}$ and $\operatorname{supp}(J_j) \cup U_j$ is disjoint from $S^*_{j-1}$, we have that $U_{j'}$ and $\operatorname{supp}(J_j) \cup U_j$ are disjoint. As the support of $\gamma_j$ consist of the support of $J_j$ along with $U_j$ and the added fresh centers, we conclude that the support of $\gamma_j$ and the support of $\gamma_{j'}$ are disjoint. □

The bad centers are the reason that $\rho^*$ is a generalized restriction as defined in Section 4.3. Before formally defining $\rho^*$ let us bound the number of bad centers in the information sets $\gamma_1, \ldots, \gamma_g$.

**Lemma 8.9.** *The number of bad centers in the information sets $\gamma_1, \ldots, \gamma_g$ is at most $O(s/\log n)$.*

*Proof.* Only chosen centers can become bad. For a chosen center in $\gamma_j$ to become bad, each non-chosen center in a neighboring square either occurs in $S^*_g$ or in one of the supports of $\gamma_i$, for $i < j$.

We claim that $\sum_{j \leq g} |\operatorname{supp}(\gamma_j)| = O(|S^*_g|)$. This is readily verified: when defining $\gamma_j$ we start out with the support being $\operatorname{supp}(J_j) \cup U_j$ and then enlarge it by at most a single center per element in the support. Lemma 8.8 implies in particular that

$$\sum_{j \leq g} |\operatorname{supp}(J_j) \cup U_j| = \Big| \bigcup_{j \leq g} \operatorname{supp}(J_j) \cup U_j \Big| \leq |S^*_g| \ , \tag{1}$$

and thus the claim follows.

By definition of $g$ and Lemma 8.7 we have that $|S_g^*| < s/4 + 8t$. Further, by assumption it holds that $t \leq s$ and thus $|S_g^*| + \sum_{j \leq g} |\operatorname{supp}(\gamma_j)| = O(s)$. Finally, every square contains $\Omega(\log n)$ non-chosen centers and hence there are at most $O(s/\log n)$ many bad centers. □

As mentioned before, closed graphs can be used to define restrictions with fewer live centers. Let $B$ denote the number of bad centers in the support of the different $\gamma_j$ and let $\gamma^* = \cup_{j=1}^{g} \gamma_j$. As each $\gamma_j$ is closed (except at the bad centers) and, by Lemma 8.8, they have pairwise disjoint supports we conclude that $\gamma^*$ has at most $B$ bad centers. We define $\rho^*$ to be the restriction defined by $\rho$ composed with the information $\gamma^*$. As previously explained, the bad centers cause $\rho^*$ to be a generalized restriction where all centers in $\operatorname{supp}(\gamma^*)$ are now dead. We call these the *disappearing* centers. By Lemma 8.9 we have at most $B \leq O(s/\log n)$ bad centers, while at least $|S_g^*| \geq s/4$ centers disappear.

## 8.6 Encoding $\rho$

We first need to introduce some more notation. Recall that $U_j$ is the set of closest endpoints of variables on the $j$th forceable branch $\psi_j$ that are chosen centers but not contained in $S_{j-1}^*$. We let $a_j$ be the number of closest endpoints of variables on $\psi_j$ that are also in $\operatorname{supp}(J_j) \cup U_j$ and let $b_j$ be the number of additional centers in $\gamma_j$, i.e., $b_j = |\operatorname{supp}(\gamma_j)| - a_j$. We let $a = \sum_{j=1}^{g} a_j$, define $b$ similarly and let $c = \left|\operatorname{supp}(I_g^*) \setminus \operatorname{supp}(\gamma^*)\right|$ be the number of centers in the support of $I_g^*$ that do not appear in the support of $\gamma^*$. The main goal of this section is to prove the following lemma stating that a restriction $\rho$ that causes the extended canonical decision tree to have a closed path of length at least $s$ can be encoded using few bits, given $\rho^*$ and $T_1, \ldots, T_m$. Put different, the mapping from $\rho$ to $\rho^*$ can be inverted with a bit of extra information. Recall that $\Delta$ is the number of centers in each sub-square.

**Lemma 8.10.** *Suppose we are given $\rho^*$ as well as the decision trees $T_1, \ldots, T_m$ each of depth at most $t$. Then*

$$a \log t + b \log \Delta + c \log \log n + O(a + b + c)$$

*many bits are needed to encode $\rho$.*

Before diving into the proof of this lemma let us show how the switching lemma follows from Lemma 8.10. For the proof of the switching lemma we need one further lemma that relates the parameters $a, b$ and $c$: it is not so hard to convince oneself that $b + c$ is of order $O(a)$. Indeed, it was shown by Håstad [Hås20] that $b + c$ is bounded by $25a$.

**Lemma 8.11** ([Hås20])**.** *It holds that $b + c \leq 25a$.*

*Proof of Lemma 7.2.* Let us analyze the probability that a random $\rho$ gives rise to a closed branch of length at least $s$. Let $m = \Delta(n/T)^2$ be the total number of centers and recall that $k = C \log n (n/T)^2$ is the total number of live centers.

Let us first count the number of restrictions $\rho$ that give rise to a closed branch of length at least $s$. By Lemma 8.10 this is upper bounded by the number of ways to choose $\rho^*$ times $t^a \Delta^b (\log n)^c A^{a+b+c}$, for some absolute constant $A$. The number of ways to choose $\rho^*$ is[5] at most $2^{1+r_n} \binom{m}{k-(b+a)} n^{2B}$, where $2^{r_n}$ is the number of possibilities for the choice of the fixed and preferred variables once the choice of centers is fixed, and $B$ is the number of bad centers.

[5] We sum the binomial coefficient over possible values of $a + b$ but this sequence is exponentially increasing and thus dominated by twice the maximal term.

In order to bound the probability that a restriction gives rise to a closed branch of length at least $s$ we also need to count the number of restrictions $\rho$. We can count these restrictions in a similar manner as we counted the restrictions $\rho^*$: there are $2^{r_n}\binom{m}{k}$ many such restrictions. Thus the probability of having a closed branch of length at least $s$ is bounded by

$$\frac{t^a \Delta^b (\log n)^c A^{a+b+c} 2^{1+r_n} \binom{m}{k-(a+b)} n^{2B}}{2^{r_n}\binom{m}{k}} \ . \tag{2}$$

The quotient of the the binomial coefficients can be bounded by

$$\begin{aligned}\prod_{i=0}^{a+b-1} \frac{k-i}{m+i-k} &\leq \left(\frac{k}{m-k}\right)^{a+b} \\ &= \left(\frac{C \log n}{\Delta - C \log n}\right)^{a+b} \\ &\leq \Delta^{-(a+b)} (\log n)^{a+b} A^{a+b} \ , \end{aligned} \tag{3}$$

for some different constant $A$. We conclude that the probability of a closed branch of length at least $s$ appearing in the extended canonical decision tree is at most

$$\Delta^{-a} (\log n)^{a+b+c} t^a A^{a+b+c} n^{2B} \ , \tag{4}$$

for a new constant $A$. Applying Lemma 8.11 and modifying $A$ again we can bound this by

$$\Delta^{-a} (\log n)^{26a} t^a A^a = (A (\log n)^{26} t \Delta^{-1})^a n^{2B} \ . \tag{5}$$

Finally, as the number of exposed centers is at most $a+b+c$ and the number of queried variables is at most four times the number of exposed centers we have $a+b+c \geq s/4$ and hence $a \geq s/104$ by Lemma 8.11. By Lemma 8.9 we have that $n^{2B} \leq 2^{O(s)}$ and we can thus incorporate this factor into the constant $A$. This concludes the analysis of the probability of the event that a closed branch of length at least $s$ appears in the extended canonical decision tree. Lemma 7.2 now follows from Lemma 8.5 and a final modification of the constant $A$. □

The rest of this section is dedicated to the proof of Lemma 8.10. On a very high level, we want to remove $\gamma^*$ from $\rho^*$. We do this in stages, where in each stage we remove a single $\gamma_j$ from $\rho^*$ by utilizing the decision trees $T_1, \ldots, T_m$ and reading a bit of extra information. Let us introduce some notation and note a simple observation in order to give a bit more detailed proof outline. For convenience let $I^*_0 = \emptyset$, let $\gamma^*_{\geq j} = \cup_{i=j}^{g} \gamma_i$, and let $\rho^*_{\geq j}$ be the restriction obtained from composing $\rho$ with the information $\gamma^*_{\geq j}$, i.e., $\rho^*_{\geq j}$ forces the same variables as $(\rho, \gamma^*_{\geq j})$ forces.

Recall that the possible forcing information $J_j$ along with $I^*_{j-1}$ determines all variables on the forceable branch $\psi_j$ of stage $j$. As $\gamma_j$ extends $J_j$ we observe that $(\rho, I^*_{j-1} \cup \gamma_j)$ traverses $\psi_j$. Further, as $\gamma^*_{\geq j}$ extends $\gamma_j$ and is consistent with $I^*_{j-1}$, it also holds that $(\rho, I^*_{j-1} \cup \gamma^*_{\geq j})$, or equivalently $(\rho^*_{\geq j}, I^*_{j-1})$, traverses $\psi_j$. This observation allows us to pursue the following high level plan.

We proceed in stages $j = 1, \ldots, g$. At the beginning of each stage $j$ we assume that we know the restriction $(\rho^*_{\geq j}, I^*_{j-1})$. Note that because $I^*_0 = \emptyset$ and $\rho^*_{\geq 1} = \rho^*$, we have that $(\rho^*_{\geq 1}, I^*_0)$ forces the same variables as $\rho^*$ and we hence have the necessary information to start at stage $j = 1$. By above observation the restriction $(\rho^*_{\geq j}, I^*_{j-1})$ traverses the forceable branch $\psi_j$. Let us assume for now that $\psi_j$ is the first 1-branch traversed, which allows us to identify $\psi_j$ for free.

This branch can in turn be used to identify a good fraction of $\gamma_j$: as $\psi_j$ is of length at most $t$ we only need to spend $\log t$ bits per variable on $\psi_j$ forced by $J_j$ to identify the corresponding closest center that disappeared. To find the remaining elements of $\gamma_j$, along with its graph structure, we use some additional external information. This lets us "remove" $\gamma_j$ from $\rho^*_{\geq j}$ to obtain $\rho^*_{\geq j+1}$. Before we can proceed with stage $j+1$ we also need to recover $I_j$. As a good fraction of the support of $I_j$ is already identified by $\gamma_j$ we can again use some external information to obtain the final missing pieces.

Unfortunately there are some complications. Recall that when we closed up the information sets $\gamma_j$ we potentially added information pieces to $\gamma_j$ that correspond to paths between chosen and non-chosen centers. Such information pieces are *not* allowed in a potential forcing information $J_j$. So it may well be that the first 1-branch traversed by $(\rho^*_{\geq j}, I^*_{j-1})$ is different from $\psi_j$. In order to find the correct forceable branch we introduce signatures.

**Definition 8.12** (signature)**.** Let $v$ be a center in the support of $\gamma_j$. The *signature* of any disappearing center $v$ consists of 9 bits. The first bit is 1 if and only if $v$ is a chosen center. For each of the four directions there is a bit indicating whether $v$ is the closest endpoint of a variable in this direction on the forceable branch $\psi_j$. For each of the four directions there is also a bit indicating whether there is an information piece in this direction in $J_j$.

**Remark:** Note that the stage $j$, although mentioned in the definition, is *not* part of the signature (as it was in [Hås20]). This change is mandated by our desire to get a tighter bound which requires a smaller signature.

By Lemma 8.8 the supports of two distinct information sets $\gamma_j$ and $\gamma_{j'}$ are disjoint and hence each center in the support of $\gamma^*$ has a unique signature. Also, recall that $J_j$ determines all variables on the forceable branch $\psi_j$ that are not determined by $I^*_{j-1}$. Hence every variable on $\psi_j$ that is not determined by $I^*_{j-1}$ has a closest endpoint with a signature.

As elaborated previously we use signatures to rule out that a candidate 1-branch is equal to the forced branch $\psi_j$. Let us define what it means for a signature to be in conflict with a 1-branch and an information set $I$. To this end observe that a chosen center $v$ along with its signature defines a partial assignment to the incident path variables: the variables in the domain of the partial assignment are all variables in the directions in which $v$ is the closest endpoint of some variable on the forceable branch (according to the first set of four bits) and these variables take values as indicated by the second set of four bits.

**Definition 8.13** (conflict)**.** Let $I$ be an information set, $\psi$ be a branch and $E$ be a set of tuples $(v, \mathsf{sign})$ each consisting of a center $v$ along with the signature sign of $v$. The set $E$ is in *conflict* with $\psi$ and $I$ if and only if either

1. there is a tuple $(v, \mathsf{sign}) \in E$ such that the directions in which $\psi$ has variables whose closest endpoint is $v$ do not agree with sign, or
2. the partial assignment on chosen path variables obtained from $I$ jointly with the assignments defined by the signatures $(v, \mathsf{sign}) \in E$, where $v$ is a chosen center and there is a variable on $\psi$ whose closest endpoint is $v$, is not consistent.

The following lemma states that if a set of signatures is not in conflict, then we have indeed identified the $j$th forceable branch $\psi_j$. This is the central lemma of the reconstruction process.

**Lemma 8.14.** *Let $E$ be the set of tuples $(v, \mathsf{sign})$ where $v \in \mathrm{supp}(\gamma^*_{\geq j})$ and sign is the signature of $v$. If $\psi$ is the first 1-branch traversed by $(\rho^*_{\geq j}, I^*_{j-1})$ such that $E$ is* not *in conflict with $I^*_{j-1}$ and $\psi$, then $\psi$ is the $j$th forceable branch $\psi_j$.*

*Proof.* We need to establish that $E$ is in conflict with $I^*_{j-1}$ and all branches $\psi$ before $\psi_j$. Suppose otherwise and let us construct a possible forcing information $J_j$ that could have been used in stage $j$ of the construction of the extended canonical decision tree to force the branch $\psi$.

On the non-chosen centers the set $J_j$ contains the pieces of $\pi$ needed to force all variables on $\psi$.

On the chosen centers the set $J_j$ consists of information pieces as given by the partial assignments defined by signatures $(v, \mathsf{sign}) \in E$ such that there is a variable on $\psi$ whose closest endpoint is $v$. These information pieces are consistent with $I^*_{j-1}$ as $E$ is not in conflict with $I^*_{j-1}$ and $\psi$. Furthermore, these force the input to traverse $\psi$ as these information pieces are the same as used in $\gamma^*_{\geq j}$. □

Before we describe the reconstruction procedure in detail we need a technical definition. Let $I^{*-}_{j-1}$ be $I^*_{j-1}$ except that we remove the information pieces that have at least one of their endpoints in $\mathrm{supp}(\gamma^*_{\geq j})$. Furthermore, let $I^-_j$ be $I_j$ with the same type of pieces taken away. The removed pieces are simple to describe. Recall that $S^*_j = \cup^j_{i=1} S(\tau_i, \sigma)$ is the set of exposed centers at the end of stage $j$.

**Lemma 8.15.** *An information piece in $I^*_{j-1}$ that is from a center in $\mathrm{supp}(\gamma^*_{\geq j})$ is in the form of a non-edge from a chosen center not in $S^*_{j-1}$ in the direction of a chosen center in $S^*_{j-1}$.*

*Proof.* According to Lemma 8.6 the information set $I^*_{j-1}$ consists of a closed graph on $S^*_{j-1}$ jointly with some non-edges from chosen centers not in $S^*_{j-1}$. Also, by Property 3, all information sets $J_{j'}$ with $j' \geq j$ are pairwise disjoint with $I^*_{j-1}$.

When extending $J_{j'}$ to $\gamma_{j'}$ we add the set $U_{j'}$ to the support, which may intersect with the support of $I^*_{j-1}$ but is disjoint from $S^*_{j-1}$. Also, we add the fresh centers to the support but these are by definition disjoint from $S^*_g \supseteq S^*_{j-1}$. Hence no $\gamma_{j'}$ with $j' \geq j$ can intersect the closed part of $I^*_{j-1}$. The statement follows. □

Hence very few information pieces are in $I^*_{j-1} \setminus I^{*-}_{j-1}$. Furthermore, these information pieces are in some sense redundant – the set $\gamma^*_{\geq j}$ contains the removed information pieces from $I^*_{j-1}$.

**Lemma 8.16.** *Any variable forced by $(\rho^*_{\geq j}, I^*_{j-1})$ is also forced by $(\rho^*_{\geq j}, I^{*-}_{j-1})$.*

*Proof.* By Lemma 8.15 the pieces removed from $I^*_{j-1}$ are next to centers that disappear in $\rho^*_{\geq j}$. As the information piece is a non-edge in both $I^*_{j-1}$ and $\gamma^*_{\geq j}$ it is forced to the same value. □

Furthermore when considered as assignments on the path variables, even though we do not know the other endpoint we know that a particular path variables is 0. This implies that $I^*_{j-1}$ and $I^{*-}_{j-1}$ are equally powerful when considering consistent values of path variables.

We can finally explain the reconstruction procedure. Throughout the procedure we maintain the following objects. A counter $j$ of the current stage to be reconstructed, the restriction $\rho^*_{\geq j}$, the information set $I^{*-}_{j-1}$, the exposed centers $S^*_{j-1}$, and a set $E$ of (prematurely identified) disappearing centers along with their signatures. Initially we set $j = 1$, $\rho^*_{\geq 1} = \rho^*$, and $S^*_0 = I^{*-}_0 = E = \emptyset$. Let us formally define the reconstruction process.

1. Find the next 1-branch $\psi$ traversed by the information $(\rho^*_{\geq j}, I^{*-}_{j-1})$.
2. If $\psi$ and $I^{*-}_{j-1}$ is in conflict with $E$, then go to Step 1.

3. Read a bit $b$ to determine if there are more disappearing centers to be found as the closest endpoint of a variable on $\psi$.

4. If $b = 1$, then we read an integer $i$ of magnitude at most $t$. This identifies the closest endpoint $v$ of the $i$th variable on $\psi$ as a disappearing center. Read the signature sign of $v$ and add $(v, \mathsf{sign})$ to $E$. If $E$ is in conflict with $\psi$ and $I^{*-}_{j-1}$, then go to Step 1. Otherwise repeat Step 3.

5. If $b = 0$, then we have found the forceable branch. Read some external information to determine $\gamma_j$ and $I^-_j$ (details below). Update $\rho^*_{\geq j}$ to $\rho^*_{\geq j+1}$, $I^{*-}_{j-1}$ to $I^{*-}_j$ and $S^*_{j-1}$ to $S^*_j$, remove all closest endpoints of $\psi$ from $E$, and set $j = j+1$. If $|S^*_j| \geq s/4$, then terminate. Otherwise go to Step 1.

Recall that each exposed center leads to at most 4 queries in the extended canonical decision tree and thus if there is a branch of length $s$, then this gives rise to a set of exposed centers $S^*_g$ of size at least $s/4$.

Let us note that for each variable identified on the forceable branch we have the signature of its closest endpoint as each such center belongs to $E$. Also, once we identified $I^{*-}_j$ it is straightforward to recover the set of exposed centers $S^*_j$.

By Lemma 8.14 and Lemma 8.16 we indeed identify the $j$th forceable branch $\psi_j$ in stage $j$. All that is left is to explain how to recover $\gamma_j$ and $I^-_j$.

We start with the reconstruction of $\gamma_j$. We identified all the closest endpoints of variables on $\psi_j$ and we know, by their signature, in which directions they need another center as the other endpoint of an edge. We read the identity of these other endpoints at a cost[6] of at most $\log \Delta$ for each center. This identifies $J_j$ along with some non-edge information pieces next to chosen centers in $U_j$ that are not contained in $\mathrm{supp}(J_j)$. To finalize the description of $\gamma_j$ we, unless a center is bad, read the identity of the unique fresh centers used to make $\gamma_j$ closed. This is done at a cost of $\log \Delta$ for each such center. Having identified $\gamma_j$ we turn to $I^-_j$. We first have a bit for each element in $\gamma_j$ to indicate whether it is also an element of $I^-_j$.

Recall that, by definition, any additional center in $\mathrm{supp}(I^-_j)$ does not belong to $\mathrm{supp}(\gamma^*_{\geq j+1})$. Thus any such center is still alive in $\rho^*_{\geq j}$ and can hence be identified using at most $\log\log n + \log 1.01C$ many bits as we know the sub-square to which it belongs.

What remains is to reconstruct the structure of $I^-_j$. Let us first reconstruct the non-chosen centers. For each non-chosen center in $J_j$, using $O(1)$ bits, we find out the size of the connected component in $\pi$ and the directions of each edge. Then we identify the other endpoint of each such edge using $\log\log n + \log 1.01C$ many bits.

For the chosen centers we can again discover the graph part with $O(1)$ bits per center for structure and an integer of magnitude $1.01C \log n$ for the identity. The non-edges not in $\mathrm{supp}(\gamma^*_{\geq j})$ are also reconstructed using $\log\log n + \log 1.01C$ bits for the identity and $O(1)$ bits per center for direction.

Finally, for any center in $\gamma_j$ we have 4 bits to describe whether the piece of information in the form of a non-edge in any direction should be added to $I^{*-}_j$.

This concludes the description of the reconstruction and we need to sum up the external information needed.

Recall that $a_j$ is the number of disappearing centers that are discovered through being the closest endpoint of a discovered variable and are part of $\psi_j$ and that $b_j$ is the number of

[6] It might be the case that some of these centers were found previously and are part of $E$ or that also the other endpoint is uniquely defined by occurring variable. In either case the cost, including the signature is $O(\log t)$ which is bounded by $\log \Delta$.

additional centers in $\gamma_j$. Furthermore let $c_j$ be the number of centers needed to be discovered in $I_j^-$ after $\gamma_j$ was discovered. As before we let $a = \sum_{j=1}^{g} a_j$ and define $b$ and $c$ similarly. The following summarizes the amount of external information needed.

- The disappearing centers that are discovered as closest endpoints contribute $a \log t$ many bits.
- The other disappearing centers contribute at most $b \log \Delta$ bits (or less as discussed in Footnote 6).
- The signatures contribute at most $(a+b)\log(A)$ many bits for a constant $A$: signatures are only needed for disappearing centers.
- The centers discovered to be part of $I$ contribute $c\,(\log 1.01C + \log\log n)$ bits.
- The graph structure of $\gamma^*$ and $I$ as well as the information which elements of $\gamma_j$ are included in $I_j$ contributes a factor $(a+b+c)\log B$, for some constant $B$.
- Throughout reconstruction at most $s + 8t + s/4$ bits $b$ are read. This follows as we can have at most $s + 8t$ bits that are 1 (as each time a disappearing variable is discovered, and this is bounded by Lemma 8.7) and at most $s$ bits that are 0 (as a stage is ended each time and $g \leq s/4$).

As the number of exposed centers is at most $a+b+c$ and the number of queried variables is at most 4 times the number of exposed centers, we have that $a+b+c \geq s/4$. As $t = O(s)$, we see that the final point requires at most $O(a+b+c)$ bits. The lemma follows.

# 9 The multi-switching lemma

The purpose of this section is to prove the multi-switching lemma, restated here for convenience.

**Lemma 7.5** (Multi-switching Lemma)**.** *There are constants $A$, $c_1$, and $c_2$ such that the following holds. Consider formulas $F_i^j$, for $j \in [M]$ and $i \in [m_j]$, each associated with a decision tree of depth at most $t$ and let $F^j = \vee_{i=1}^{m_j} F_i^j$. Let $\sigma$ be a random full restriction from the space of restrictions defined in Section 4. Then the probability that the number of live variables in each center is in the interval $[.99C\log n, 1.01C\log n]$ and $(F^j\lceil_\sigma)_{j=1}^M$ cannot be represented by an $\ell$ common partial decision tree of depth at most $s$ is at most*

$$M^{s/\ell}\left(A(\log n)^{c_1} t \Delta^{-1}\right)^{s/c_2} .$$

The proof of Lemma 7.5 very much follows the proof of Lemma 7.2. The first section gives a short proof overview, followed by Section 9.2 that formally explains how to construct a common partial decision tree. The following section then establishes some important properties about these decision trees and in Section 9.4 we finally prove Lemma 7.5.

## 9.1 Proof Overview

The high level proof outline is as follows. Consider the formulas $F^1, \ldots, F^M$ in order. If a formula $F^j$ is not turned in to a decision tree of depth $\ell$, find a branch in the extended canonical decision tree of $F^j$ of length at least $\ell$. Put the variables on this branch in the common partial decision tree, query those variables as well as some extra variables and recurse.

As in the proof of the switching lemma we consider any full restriction $\sigma = \sigma(\rho, \pi)$ for which Lemma 7.5 fails. With the aid of the formulas $F^1, \ldots, F^M$ we then turn $\rho$ into a $\rho^*$ such that the mapping can be inverted with little extra information to argue that there are few full restrictions for which Lemma 7.5 fails.

One complication to handle is that the answers to the variables found on a long branch in the extended canonical decision tree of $F^j$ and the answers to the same variables on the long branch in the common partial decision tree may differ. This leads to the more complicated game analyzed in Section 3.1.

## 9.2 Common Partial Decision Trees

Let us explain how to construct the $\ell$ common partial decision tree $\mathcal{T}$ of $F^1\restriction_\sigma, \ldots, F^M\restriction_\sigma$. Start with $\mathcal{T}$ empty. We proceed in *rounds*. In each round we consider a leaf $\tau$ of $\mathcal{T}$ such that there is a formula $F_j$ that cannot be represented by a depth $\ell$ decision tree under $\sigma$ and $\tau$. We extend $\mathcal{T}$ at $\tau$ as follows.

Let $j$ be minimum such that $F^j\restriction_{\sigma\tau}$ cannot be represented by a depth $\ell$ decision tree. Create the extended canonical decision tree $T^j$ of $F^j\restriction_{\sigma\tau}$ (similar to Section 8.3 – see below) and denote by $\psi$ a branch of length at least $\ell$ in $T^j$. Extend $\mathcal{T}$ at $\tau$ by querying all variables on $\psi$ and some extra variables as specified in the following.

Modulo the precise definition of the extended canonical decision tree used and the extra variables to be queried this describes the entire creation process of an $\ell$ common partial decision tree. Let us first discuss how we construct the extended canonical decision trees in above procedure. The biggest difference to the definition in Section 8.3 is that we rely on a different game: instead of relying on the simple matching game as defined in Section 3, we are going to rely on the more complicated game defined in Section 3.1. Furthermore we initialize the various objects used in the creation of the extended canonical decision tree with information from previous rounds. Let us explain this in more detail.

Throughout the creation of the $\ell$ common partial decision tree $\mathcal{T}$ we maintain the following objects for each leaf $\tau$ of $\mathcal{T}$: a set of exposed centers $S = S(\tau, \sigma)$, a set of information pieces $I = I(\tau, \sigma)$ as well as a (state of a) game $\mathcal{G} = \mathcal{G}(\tau, \sigma)$ as defined in Section 3.1, played on the chosen centers of $\sigma$. Let us stress that this is *not* the game used in Section 8.3.

Initially the set of exposed centers $S$ and the information set $I$ are empty and the game $\mathcal{G}$ is a new game. Throughout the creation of $\mathcal{T}$ and the various extended canonical decision trees we maintain the same invariants as in Section 8.3 except Invariant 2:

1. No element is ever removed from $S$ or $I$.
2. The picked nodes in the game $\mathcal{G}$ are precisely the chosen centers in $S$.
3. The information set $I$ does not contain paths between chosen and non-chosen centers.
4. For non-chosen centers in $S$, the set $I$ consists of the closed information pieces corresponding to their component in $\pi$ (both edges and non-edges). If one center of such a connected component belongs to $S$, then so does the entire component.
5. For every chosen center in $S$ we have queried all incident variables $x_p$ in $\tau$ and this is the information that is present as information pieces in $I$. The one-answers are recorded in the form of a path while the zero answers as two non-edges, one at the neighboring chosen center in the appropriate direction which may or may not be an element of $S$.

In each round of the construction of $\mathcal{T}$, when building the extended canonical decision tree $T^j$ of a formula $F^j$, we initialize the sets $S, I$ and $\mathcal{G}$ used in the creation of $T^j$ with the

corresponding objects maintained for the creation of the common partial decision tree $\mathcal{T}$. Other than that the creation of $T^j$ essentially follows Section 8.3: in each stage we find a new forceable branch $\psi_i^j$, the corresponding forcing information $J_i^j$ and add all vertices in $\mathrm{supp}(J_i^j)$ and $U_i^j$ to $S$. The adversary in the game $\mathcal{G}$ does a simple move with every chosen center $v$ just added to $S$. All vertices picked by the player $P$ in response to these simple moves are then added to the set $S$.

To find out whether the forceable branch $\psi_i^j$ is followed we get information sets $I_i^j$ consisting of pieces from $\pi$ and answers from $T^j$.

We continue with the next stage until at least $\ell/4$ centers have been added in this round to a set $S(\lambda^j, \sigma)$ for some leaf $\lambda^j$ of $T^j$. We know that this happens as $F^j\lceil_{\sigma\tau}$ could not be decided by a decision tree of depth $\ell$ (recall that $\tau$ is the leaf of $\mathcal{T}$ we are considering).

Once we have such a long branch $\lambda^j$ in $T^j$ we extend $\mathcal{T}$ at $\tau$ by querying all variables on $\lambda^j$. For each newly created leaf $\tau'$ in $\mathcal{T}$ we need to update $S, I$ and $\mathcal{G}$. For now we copy $S$ and $\mathcal{G}$ as in the leaf $\lambda^j$ of $T^j$ and set $I$ according to $\tau'$ (which includes the same components of $\pi$ as in $\lambda^j$ but the information pieces next to chosen centers may differ).

Finally we can explain what extra variables are queried at the end of a round: in each such leaf $\tau'$ the adversary in the game $\mathcal{G}$ announces that the round is over. Further, the adversary makes all edges between exposed and non-exposed centers active if their corresponding assignment on $\tau'$ and $\lambda^j$ differs. Note that because values in both decision trees are locally consistent, the number of edges they differ in incident to any connected component is even. Hence this is a legitimate move for the adversary.

All nodes picked by $P$ in response are added to $S$ and all variables incident to these nodes are queried (in $\mathcal{T}$) and recorded in $I$. This completes the description.

Note that at the end of each round the information pieces in $I$ are determined by the branch $\tau$ of $\mathcal{T}$ and the matching $\pi$. The information pieces $I_1^j, I_2^j, \ldots$ on the chosen centers used to determine $\lambda^j$ in $T^j$ are now forgotten. These answers were only used to find the long branch $\lambda^j$.

Clearly the above process creates an $\ell$ common partial decision tree and we need to analyze the probability that we get a tree of depth at least $s$.

## 9.3 Some Properties of Common Partial Decision Trees

As in the standard switching lemma case, the creation of the common decision tree remains the same if we negate the answers, simultaneously in both the extended canonical decision tree and the common partial decision tree, and the suggested values on the paths between chosen live centers that are exposed. We state this as a lemma.

**Lemma 9.1.** *Let $\sigma_1$ be obtained from $\rho_1$ and $\pi$ and $\sigma_2$ from $\rho_2$ and $\pi$ where $\rho_1$ and $\rho_2$ pick the same set of centers and fixed values. Assume furthermore that the only difference between $\rho_1$ and $\rho_2$ is that for each chosen path $P$ there is a bit $c_P$ such that for each grid-edge $e$ on $P$ the preferred values of $x_e$ differ by $c_P$ in $\rho_1$ and $\rho_2$. Then the only difference between the common decision decision trees of $(F^j\lceil_{\sigma_1})_{j=1}^M$ and $(F^j\lceil_{\sigma_2})_{j=1}^M$ is the labeling of the internal edges.*

Next we need to define the notion of a closed path in the common partial decision tree. Informally we want any answer between an exposed center and a non-exposed center to be 0. As we query variables both in the extended canonical decision tree $T^j$ of $F^j$ and the common partial decision tree $\mathcal{T}$ let us be more specific. We require the following queries to have 0 answers.

- At any stage in the processing of $F^j$ an edge between a center exposed in this stage and a non-exposed center. This is the answer in $T^j$.
- At the end of the round any answer between a chosen center exposed in the round and a non-exposed center (after the completion move in $\mathcal{G}$). This is the answer in $\mathcal{T}$.

Note that an edge of the first type, if it remains an edge between an exposed center and a non-exposed center also after the completion of the round, then the answer is 0 also in the common partial decision tree $\mathcal{T}$. Indeed if the answers differ in $T^j$ and $\mathcal{T}$, the non-exposed node is exposed by the rules of the game.

**Lemma 9.2.** *If the probability that $(F^j\lceil_\sigma)_{j=1}^M$ needs a $\ell$ common partial decision tree of depth $s$ is at least $q$, then the probability that this happens with a closed execution of length at least $s$ is at least $2^{-s}q$.*

*Proof.* We just need to show that there are locally consistent assignments that gives the required values. By the rules of our combinatorial game, at each stage of the game each connected component is of even size and hence by Lemma 3.3 we can get border values that are all zero. As each connected component of the complement is of even size we can make the assignment also locally consistent.

Similarly at the end of the round. An active edge corresponds to a value that is 1 in the common decision tree (as it is 0 in $T^j$ and they are different). The condition that the number of active edges next to any component is even is implied by local consistency. □

## 9.4 Putting the Pieces Together

Once we have set up the machinery the proof parallels the proof of the standard switching lemma. We need to verify that it works but no new complications arise.

Using fresh centers we again extend $J_i^j$ to make them closed forming information sets $\gamma_i^j$. There might be $O(s/\log n)$ centers for which this process fails and this gives $O(s/\log n)$ bad centers as in the standard switching lemma. The restriction $\rho^*$ is obtained by applying these $\gamma_i^j$ to $\rho$. We need to specify the information needed to invert this mapping. Each round is very similar to the standard switching of a single formula and we use the following information.

- The identities of which $F^j$ are processed.
- The inverting information for each single formula, $F^j$, as used in the inversion process in the standard switching lemma.
- The difference in values of variables queried in the decision tree for $F^j$ and the same variables in the common decision tree.
- The identities of the centers exposed at the end of each round.

The inversion process of each round runs completely parallel to the inversion for the standard switching lemma. The information of which $F^j$ to process is here crucial as $\rho$ does force many $F^j$s to constants. We recover the information pieces used in the single formula process. At the end of the round we use the knowledge of the differences to turn this into the information pieces for the common decision tree. We recover the identities of vertices exposed at the end of each round and add these information pieces to our information set before starting the next round.

For the final calculation, as in the standard case, there is a profit of $\Omega(\log(\Delta) - \log\log n - \log t)$ bits for each center discovered if it is the closest endpoint of a variable on a forceable branch $\psi_i^j$. This corresponds to the simple moves of the adversary in the combinatorial game.

All other exposed centers are retrieved at cost $O(\log\log n)$. The key to the analysis is Lemma 3.12 that establishes that a constant fraction of all moves are profitable.

Of the extra information needed, only the identities of the processed formulas cannot be absorbed into the constant $A$ or in poly-logarithmic factors, and it gives the first factor of the bound in Lemma 7.5.

## 10 Conclusion

Of course our bounds are not exactly tight so there is always room for improvement. We could hope to get truly exponential exponential for a bounded depth Frege proof, i.e., essentially bounds $2^n$ where $n$ is the number of variables. Since any formula given by a small CNF has a resolution proof this is the best we could hope for. As our formulas have $O(n^2)$ variables we are off by a square. If one is to stay with the Tseitin contradiction one would need to change the graph and the first alternative that comes to mind is an expander graph. We have not really studied this question but as our current proof relies heavily on properties of the grid; significant modifications are probably needed.

This brings up the question for which probability distributions of restrictions it is possible to prove a (multi) switching lemma. Experience shows that this is possible surprisingly often. It seems, however, that it needs to be done on a case by case basis. Probably it is too much to ask for a general characterization but maybe it could be possible to prove switching lemmas that cover several of the known cases.

**Acknowledgments.** We are grateful to Mrinal Ghosh, Björn Martinsson and Aleksa Stanković for helpful discussions on the topic of this paper.

## References

[Ajt94] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994. Preliminary version in *FOCS '88*. 1

[Ben02] Eli Ben-Sasson. Hard examples for the bounded depth Frege proof system. *Computational Complexity*, 11(3-4):109–136, 2002. 1

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979. 19

[FSS84] M. Furst, J.B. Saxe, and M. Sipser. Parity, circuits and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984. 1

[Hak85] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297 – 308, 1985. 1

[Hås86] J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, STOC '86, pages 6–20, New York, NY, USA, 1986. ACM. 1, 3

[Hås14] J. Håstad. On the correlation of parity and small-depth circuits. *SIAM Journal on Computing*, 43:1699–1708, 2014. 2, 3

[Hås20] J. Håstad. On small-depth frege proofs for tseitin for grids. *Journal of the ACM*, 68:1–31, 2020. 0, 1, 2, 3, 4, 5, 6, 12, 13, 15, 18, 19, 20, 21, 24, 32, 34

[IMP12] Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithms for $AC^0$. In *Proceeding of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 961–972, 2012. 2

[KPW95] Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures & Algorithms*, 7(1):15–39, 1995. 1, 18

[PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993. Preliminary version in *STOC '92*. 1

[PRST16] Toniann Pitassi, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Polylogarithmic frege depth lower bounds via an expander switching lemma. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 644–657, New York, NY, USA, 2016. Association for Computing Machinery. 1, 2, 3, 19

[PRT22] T. Pitassi, P. Ramakrishnan, and L. Tan. Tradeoffs for small-depth frege proofs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 445–456, Los Alamitos, CA, USA, feb 2022. IEEE Computer Society. 0, 2, 3, 18, 19

[Raz88] A. Razborov. Bounded-depth formulae over the basis { AND,XOR} and some combintorial problems (in russian). *Problems of Cybernetics. Complexity Theory and Applied Mathematical Logic*, pages 149–166, 1988. 1

[Raz95] A. A. Razborov. *Bounded Arithmetic and Lower Bounds in Boolean Complexity*, pages 344–386. Birkhäuser Boston, Boston, MA, 1995. Editors Peter Clote and Jeffrey Remmel. 25

[RST15] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1030–1048, 2015. 14

[Sip83] M. Sipser. Borel sets and circuit complexity. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, STOC '83, pages 61–69, New York, NY, USA, 1983. ACM. 1

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pages 77–82, New York, NY, USA, 1987. ACM. 1

[Tse68] G. S. Tseitin. On the complexity of derivation in the propositional calculus. In A. O. Slisenko, editor, *Studies in constructive mathematics and mathematical logic, Part II*, 1968. 1

[UF96] Alasdair Urquhart and Xudong Fu. Simplified lower bounds for propositional proofs. *Notre Dame Journal of Formal Logic*, 37(4):523–544, 1996. 1, 18

[Yao85] A. C-C. Yao. Separating the polynomial-time hierarchy by oracles. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 1 –10, oct. 1985. 1