Lifting with Colourful Sunflowers

Susanna de Rezende Marc Vinyals

Abstract

We show that a generalization of the DAG-like query-to-communication lifting theorem, when proven using sunflowers over non-binary alphabets, yields lower bounds on the monotone circuit complexity and proof complexity of natural functions and formulas that are better than previously known results obtained using the approximation method. These include an $n^{\Omega(k)}$ lower bound for the clique function up to $k \leq n^{1/2-\epsilon}$, and an $\exp(\Omega(n^{1/3-\epsilon}))$ lower bound for a function in P.

1 Introduction

In 1985, Razborov [Raz85] proved the first superpolynomial size lower bound for monotone Boolean circuits for a monotone function in NP, and, independently, Andreev [And85] obtained exponential size lower bounds. Razborov's result was for the perfect matching and the clique functions, and shortly after, Alon and Boppana [AB87] improved the lower bound for clique to exponential. These lower bounds introduced the so-called *approximation method*, which consists of showing that every gate computes a function close to a class of functions that cannot distinguish between the expected answers. The interpretation of close becomes looser in gates further away from the inputs. The classical combinatorial *sunflower lemma* is used to ensure that the class of functions computed by gates increases slowly, and therefore only contains the expected output function once the circuit size surpasses the bound we are set to prove.

The approximation method also found great success in proof complexity, after Krajíček [Kra97, Kra98] introduced interpolation to proof complexity and Pudlák [Pud97] showed it could be used to prove cutting planes lower bounds from lower bounds to a generalisation of monotone circuits called monotone real circuits. For a long time the only known method for proving lower bounds for monotone circuits and for cutting planes was via approximation method. Other methods like bottleneck counting were later shown to be equivalent to the approximation method for monotone circuits [ST00, BU99, Juk97]. In contrast, lower bounds for proof systems such as resolution and polynomial calculus can be readily obtained from measuring width and degree respectively.

Not long after the approximation method was introduced, new and very influential techniques were developed to prove lower bounds for monotone formulas and later extended to capture tree-like cutting planes. Karchmer and Wigderson [KW90] proved superpolynomial formula size lower bounds for connectivity by establishing a relation between communication complexity and monotone circuit depth. Raz and McKenzie [RM99] introduced a new technique, now called *lifting theorems*, for obtaining communication lower bounds from query complexity lower bounds, and obtained separations between $\mathsf{mon\text{-}NC}$ and $\mathsf{mon\text{-}P}$, and between levels of the $\mathsf{mon\text{-}NC}$ hierarchy. Their result was extended [BEGJ00] to obtain lower bounds on the size of tree-like cutting planes proofs.

A decade ago, Göös, Pitassi, and Watson [GPW18] brought to light the generality of the result of Raz and McKenzie [RM99] and reignited this line of work. A notable extension is the lifting theorem [GGKS20] for a model of DAG-like communication [Raz95, Sok17a] that corresponds to circuit size. These theorems, in their different flavours, have been instrumental in addressing many open questions in monotone circuit complexity, including: optimal $2^{\Omega(n)}$ lower bounds on the size of monotone Boolean formulas computing an explicit function in NP [PR17]; a complete picture of the relation between the $\mathsf{mon\text{-}AC}$ and $\mathsf{mon\text{-}NC}$ hierarchies [dRNV16]; a near optimal separation between monotone circuit and monotone formula size [dRMN+20]; and an exponential separation between NC^2 and $\mathsf{mon\text{-}P}$ [GGKS20, GKRS19].

One criticism to the lifting technique is that the lower bounds obtained are for artificial functions and formulas. This is because the lower bounds are not for the same function whose query complexity is high, but for the composition of two functions. This criticism applies to most lower bounds obtained via lifting discussed above. Nevertheless, we can obtain lower bounds for natural functions and formulas if we can reduce them to their lifted counterparts. Furthermore, we can carry the reduction already within the communication complexity model, which is easier than reducing between circuits or proofs. This idea appears already in Raz and McKenzie's work [RM99], where they lift the collision-finding relation—equivalent to the pigeonhole principle—and subsequently apply a reduction to obtain a depth lower bound for clique. More recent results in circuit complexity also combine lifting and reductions [Oli15, Rob18, dRMN+24] (see also [dRGR22]), as do results in proof complexity [IR21]. Another issue is that bounds obtained through lifting are often weaker than those obtained through more direct means, because generally composition increases the size of a function more than its complexity. The question of minimising the overhead due to lifting is recurring in the area.

Following on the topic of the strength of lower bounds, it is very natural to ask how well these match the corresponding upper bounds, and to aim for truly exponential lower bounds of the form $\exp(\Omega(n))$ rather than only $\exp(n^{\Omega(1)})$. Razborov's result [Raz85] is for two monotone functions on graphs: $n^{\Omega(\log n)}$ lower bound for perfect matching and $n^{\Omega(k)}$ lower bound for k-clique, for $k \leq \log n$. Shortly after, Alon and Boppana [AB87, Weg87] extended the result on clique to $n^{\Omega(k^{1/2})}$ for $k \leq n^{2/3-\epsilon}$, and Amano and Maruoka [AM04] provided lower-order improvements. More recently, Cavalar et al. [CKR22] showed a stronger result of $n^{\Omega(k)}$ for $k \leq n^{1/3-\epsilon}$.

Besides clique, if we consider lower bounds on any function in NP, Andreev [And85] proved exponential lower bounds of the form $\exp(\Omega(n^{1/8}))$. Alon and Boppana [AB87], Andreev [And87], and Harnik and Raz [HR00] successively improved the bound to $\exp(\Omega(n^{1/3-\epsilon}))$, and also recently Cavalar et al. [CKR22] proved a lower bound of the form $\exp(\Omega(n^{1/2-\epsilon}))$.

It is also of significant interest to prove lower bounds for functions in P, showing that monotone circuits are weaker than circuits with negations. The first such separation is Razborov's superpolynomial bound on the matching function, and Tardos [Tar88] proved the first exponential separation of the form $\exp(\Omega(n^{1/6}))$. This latter result follows from the exponential lower bound for the clique–colouring function [AB87] and an upper bound Tardos proved for a monotone function extending clique–colouring. The more recent lower bounds of Harnik and Raz and Cavalar et al. apply to functions that are not known to be in P.

The breakthrough that allowed Cavalar et al. to prove lower bounds beyond $\exp(n^{1/3})$ was the robust sunflower lemma [Ros14, Rao20, ALWZ20]. This improved sunflower lemma was also used to improve the parameters in the lifting theorem [LMM+22]. Indeed, the lifting overhead was reduced to the point that bounds obtainable by lifting could match those obtained via the approximation method. However, to our knowledge, this has not been applied to get better bounds in circuit or proof complexity.

1.1 Our results

In this paper we go one step further and use lifting to prove even better lower bounds than those currently known using the approximation method. We do so by generalising the lifting theorem to non-binary alphabets, using what we call *colourful sunflowers* (simply sunflowers over a larger alphabet). This allows us to obtain smaller lifted problems than what we would through a binary alphabet. We then reduce this lifted problem to the clique–colouring and related functions. This gives us better lower bounds on the size of monotone circuits and cutting planes proofs.

Clique–colouring We begin with the monotone circuit complexity of the clique function, which distinguishes graphs containing a k-clique from graphs that do not. In fact we consider the clique–colouring partial function, which distinguishes n-vertex graphs containing a k-clique from graphs that are c-colourable, for $c < k$. Since the clique function extends the clique–colouring function, it is at least as hard to compute.

The best lower bound to date for the clique–colouring function was $n^{\Omega(k^{1/2})}$ for $k \leq n^{2/3-\epsilon}$.

Theorem 1.1 ([AB87])**.** *Given $c, k, n \in \mathbb{N}$ satisfying $c < k$, $\sqrt{c}k \leq n/(8 \log n)$, it holds that any monotone Boolean circuit computing* $(\text{clique}_k\text{-col}_c)_n$ *must have size*

$$\left(\frac{n}{\sqrt{c}k \log n}\right)^{\Omega\left(\sqrt{c}\right)} . \tag{1}$$

A better $n^{\Omega(k)}$ bound for $k \leq n^{1/3-\epsilon}$ was also known for the clique function, but not for the clique–colouring function.

Theorem 1.2 ([CKR22])**.** *Given $k, n \in \mathbb{N}$ satisfying $k \leq n^{1/3-\epsilon}$, it holds that any monotone Boolean circuit computing k-clique must have size*

$$\left(\frac{n}{k^3}\right)^{\Omega(k)} . \tag{2}$$

We obtain a $n^{\Omega(k)}$ bound for $k \leq n^{1/2-\epsilon}$ for the clique–colouring function, improving on both results

Theorem 1.3 (Clique-colouring)**.** *There exists a universal constant A such that given any $c, k, n \in \mathbb{N}$ satisfying $c < k \leq n$, it holds that any monotone Boolean circuit computing* $(\text{clique}_k\text{-col}_c)_n$ *must have size*

$$\left(\frac{n}{A \cdot kc \log k \log n}\right)^{\Omega(c)} . \tag{3}$$

If we assume $c = \epsilon k$ for a constant $\epsilon > 0$, then we can get a slightly better lower bound: any monotone Boolean circuit computing $(\text{clique}_k\text{-col}_{\epsilon k})_n$ *must have size*

$$\left(\frac{n}{A \cdot k^2 \log n}\right)^{\Omega(k)} . \tag{4}$$

Note that we get tight $n^{\Omega(c)}$ lower bound even for large gap, that is, for any $k = n^{1-\delta}$ and $c \in [\log n, n^{\epsilon}]$, for any constants $\epsilon < \delta$. Even if $k = n/(\log^4 n)$ and $c = \log n$, we obtain a superpolynomial lower bound. Finally, observe that the largest lower bound we can get is $\exp(\Omega((n/\log n)^{1/2}))$, where we recall that n is the number of vertices in the input graph, and the number of variables in the function is $\Theta(n^2)$.

Colouring–cocolouring Since a partition of a graph into n/k cliques contains a clique of size at least k, an even more specific task than clique–colouring is distinguishing q-colourable graphs from q-purely-cocolourable graphs. We state the result in terms of monotone real circuits, since that is needed to prove cutting planes lower bounds.

Theorem 1.4 ([HP17])**.** *For $n > q^2$, every monotone function that distinguishes n-vertex graphs G with $\chi(G) \leq q$ from graphs with $\chi(\overline{G}) \leq q$ has monotone real circuit complexity $2^{\Omega(q^{1/4})}$.*

We get a stronger lower bound, but only for $n = q^2 + 1$.

Theorem 1.5 (Colouring-cocolouring)**.** *Every monotone function that distinguishes (q^2+1)-vertex graphs G with $\chi(G) \leq q$ from graphs with $\chi(\overline{G}) \leq q$ has monotone real circuit complexity $2^{\Omega((q/\log q)^{1/2})}$.*

We believe this result is not tight and that there should be a lower bound of 2^q. We can, however, get a tight lower bound in a non-balanced parameter regime.

Theorem 1.6. *For $c \leq n^{1/3-\epsilon}$, every monotone function that distinguishes n-vertex graphs G with $\chi(G) \leq c$ from graphs with $\chi(\overline{G}) \leq n/c - 1$ has monotone real circuit complexity $n^{\Omega(c)}$.*

Bit pigeonhole principle Turning to proof complexity, we consider the complexity of refuting the bit pigeonhole principle in the cutting planes proof system, where proofs manipulate linear inequalities. The *bit pigeonhole principle formula* bitPHP_N^M for $M > N$ falsely asserts that there are M distinct binary strings of length $\log N$. If $M \geq 2N$, we refer to this formula as a *weak* bit pigeonhole principle formula. Hrubeš and Pudlák [HP17] provided the first cutting planes lower bound for bitPHP, which prior to this work was the best bound known for pigeonhole principle. Note that their result works also for the weak version.

Theorem 1.7 ([HP17])**.** *Every cutting planes refutation of the weak bit pigeonhole principle bitPHP_N^M, $M > N$, has size $2^{\Omega(N^{1/8})}$.*

Our result is only for the bit pigeonhole principle (not weak), but we obtain a better bound.

Theorem 1.8 (Bit pigeonhole principle)**.** *Every cutting planes refutation of the bit pigeonhole principle bitPHP_N^{N+1} has size $2^{\Omega((N/\log N)^{1/3})}$.*

Stronger lower bounds on the size of refutations of the (not weak) bit pigeonhole principle of the form $2^{n^{1-o(1)}}$ are known for the tree-like restriction of the cutting planes proof system [IR21, BW24].

Separation between monotone and non-monotone Boolean circuits We also obtain the best known separation between monotone Boolean circuits and non-monotone Boolean circuits.

Theorem 1.9 ([AB87, Tar88])**.** *There exists a monotone function $f \in \mathsf{P}$ such that any monotone circuit computing f is of size at least $\exp(\Omega(n^{1/6-\epsilon}))$.*

Using the result of [Tar88] that there exists a total monotone function in P that extends the partial clique-colouring function and our lower bound for the clique-colouring, we immediately obtain that there exists a monotone function $f \in \mathsf{P}$ over n variables such that any monotone circuit computing f is of size at least $\exp(\Omega(n^{1/4-o(1)}))$. We can, furthermore, obtain an even better separation by considering a function f obtained by restricting most of the edges of the input graph. By doing this, the lower bound for clique-colouring still applies, but the number of variables is reduced considerably and we get the following result.

Theorem 1.10 (Monotone vs non-monotone)**.** *There exists a monotone function $f \in \mathsf{P}$ such that any monotone circuit computing f is of size at least* $\exp(\Omega(n^{1/3-o(1)}))$*.*

1.2 Technical contribution

The main technical contribution is to show that we can obtain better parameters by proving a generalised lifting theorem over large alphabets. We prove the lifting theorem following the same high-level structure as many others before. The proof consists of two key lemmas, a full range lemma and a triangle lemma, and a simulation procedure that relies on these two lemmas. The triangle lemma and the simulation procedure only change minimally, and the full range lemma follows from the sunflower lemma after adapting the probability distributions appropriately.

Once the lifting theorem is in place, we define a 2-CSP version of the graph pigeonhole principle. That is, we encode that k elements can be coloured with $c < k$ colours without repetitions, each element having a pool of constantly many available colours. We show how clique–colouring reduces to this lifted CSP using a similar reduction to previous works [RM99, dRMN$^+$24].

It is immediate that colouring-cocolouring reduces to clique-colouring. We show a reduction in the other direction which gives us lower bounds for colouring-cocolouring. A known further reduction [HP17], unmodified, yields lower bounds on the size of cutting planes refutations of the bit pigeonhole principle.

To explain why we obtain better parameters, it is convenient to think in terms of the falsified constraint search relation. For a fixed CSP, this relation consists of pairs of an assignment to variables and a constraint falsified by that output. This relation is universal, in the sense that for every relation S there exists a CSP whose search relation is S.

To apply the lifting technique we start with the search relation of a CSP on n variables and c constraints of arity k, which is hard according to some query complexity measure. Then we compose that relation with a function on m variables, a gadget, and we obtain a search relation of a CSP on nm variables and $O(c \cdot m^k)$ constraints, which is hard according to a communication complexity measure. When we reduce the lifted relation to e.g. clique, the size of the graph depends on the size of the lifted CSP, while the size of the clique depends on the original number of variables. Therefore minimising the arity of the CSP, along with the number of variables of the gadget, results in the least overhead.

Often it is enough to lift a CSP with constant arity over the binary alphabet, and the exact constant might not be too important if it is overshadowed by the gadget size. However, lifting theorems obtained through sunflowers can be applied with small gadgets of size $m^{1+\epsilon}$, and arity becomes more relevant. For example, we could lift the standard 3-CSP Boolean encoding of the pigeonhole principle and obtain lower bounds for the clique function that hold for cliques of size up to $n^{1/3-\epsilon}$. Using a 2-CSP with a constant-sized alphabet, however, allows us to obtain lower bounds for cliques of size up to $n^{1/2-\epsilon}$.

We remark that the work introducing lifting theorems [RM99] already proves a lifting theorem for a large alphabet, which is used to lift a 2-CSP over an alphabet of linear size, precisely with the goal of proving formula lower bounds for the clique function. However, subsequent works only considered the Boolean case.

2 Lifting Theorem

A communication problem is a relation $S\colon X \times Y \to Z$. We refer to an input $x \in X$ as Alice's input, and to an input $y \in Y$ as Bob's input. A DAG-like communication protocol computing a communication problem is a DAG where each node is identified with a set $R \subseteq X \times Y$, with the following two properties. Each internal node is covered by its children. Each leaf is labelled with an element $z \in Z$ such that $z \in S(x, y)$ for all $(x, y) \in R$.

A subcube-DAG protocol has the additional property that all nodes are subcubes, a rectangle-DAG protocol has the property that all nodes are combinatorial rectangles, and a triangle-DAG protocol has the property that all nodes are combinatorial triangles.

We assume that the fan-out of a subcube-DAG is Σ when $X \times Y = \Sigma^n$, and that the fan-out of rectangle- and triangle-DAGs is 2.

The width of a subcube-DAG is the maximum width (or codimension) of any node, and the width of a relation S, denoted $\mathsf{w}(S)$, is the minimum width of any subcube-DAG computing it. Width is equivalent to following variation of the standard query complexity game, also called *Prover–Adversary* game [Pud00, AD08]. A player may query a variable or forget the value of a variable. The player aims to minimise the number of values they remember. The answer to a query is given by an adversary who aims to maximise that number and may give different answers to the same query.

The indexing function $\mathsf{Ind}_m\colon [m] \times \Sigma^m \to \Sigma$ is defined as $\mathsf{Ind}_m(x, y) = y_x$. The composition of a relation $S\colon W^n \to Z$ and a function $g\colon V \to W$, which we call a gadget, is $S \circ g^n\colon V^n \to Z$.

Theorem 2.1 (Lifting Theorem). *There exists a large enough absolute constant A such that for any $m, n, w \in \mathbb{N}$ and any relation $S\colon \Sigma^n \to Z$ with $\mathsf{w}(S) \geq w$, the size of any triangle-DAG protocol solving $S \circ \mathsf{Ind}_m^n$ is at least*

$$\left(\frac{m}{A \cdot |\Sigma| \cdot w \cdot \log(mn)}\right)^w . \tag{5}$$

The theorem applies to rectangle-DAG protocols, which are a particular case of triangle-DAG protocols. It also holds that the communication complexity of $S \circ \mathsf{Ind}_m^n$ is bounded below by the query complexity of S.

The search problem of a CSP F is the relation $\mathsf{Search}(F)\colon \Sigma^n \to F$ that, given an assignment $\alpha\colon [n] \to \Sigma$ returns a constraint falsified by α. The relation is total if the CSP is unsatisfiable. In the context of a communication problem, we write $\mathsf{Search}^{X,Y}$ to explicitly refer to the variable partition.

Corollary 2.2. *For any $\epsilon > 0$, any alphabet Σ, any unsatisfiable CSP F on n Σ-variables, and any $m \geq (|\Sigma| \cdot \mathsf{w}(S) \cdot \log n)^{1+\epsilon}$, the size of any triangle-DAG protocol solving $\mathsf{Search}(F) \circ \mathsf{Ind}_m^n$ is at least $m^{\Omega(\mathsf{w}(\mathsf{Search}(F)))}$.*

We prove the lifting theorem following the proof of the same theorem over the binary alphabet by de Rezende et al. [dRFJ+24]. Their proof simplifies proof of Lovett et al. [LMM+22], which in turn builds upon many other lifting theorems [GLM$^+$16, GPW20, GGKS20]. The proof of the lifting theorem relies on two lemmas, the full range lemma and the triangle lemma. A procedure that extracts a subcube-DAG from a triangle-DAG completes the proof. We describe how the proof needs to be adapted to a larger alphabet next. Since many parts of the proof change minimally or not at all, we defer the full proof to the appendix.

2.1 Full Range Lemma

The full range lemma states that, given a rectangle comprised of a well-structured X part and a large enough Y part, $\mathsf{Ind}(X, Y)$ takes all possible values. Following Lovett et al. [LMM$^+$22], we prove this lemma as a consequence of the robust sunflower lemma, adapting their proof to a larger alphabet. Note that we cannot use the self-contained proof of de Rezende et al., which would result in a gadget size of at least $(n|\Sigma|)^{2+\epsilon}$, while a gadget of size $(n|\Sigma|)^{1+\epsilon}$ is enough for a proof based on the sunflower lemma. In fact we can use gadgets of variable size $(w|\Sigma|)^{1+\epsilon}$, depending on the width of the DAG we extract [GKMP20], assuming that $w = \Omega(\log n)$. It is possible to remove the assumption $w = \Omega(\log n)$ with a more complex version of the full range lemma [LMM$^+$22], but we do not require such small width.

We proceed to state the sunflower lemma. For a set $\mathcal{F}$, $\mathcal{U}(\mathcal{F})$ is the distribution where we pick an element from $\mathcal{F}$ with probability $1/|\mathcal{F}|$. $\mathcal{U}(X, p)$ is the distribution where we pick a set $S \subset X$ where $\Pr[x \in S] = p$ for all $x \in X$ independently. $\mathcal{D}_1(\{0,1\}^N, p)$ is the distribution where we pick a string $s \in \{0,1\}^N$ where $\Pr[s_i = 1] = p$ for all $i \in [N]$ independently.

The projection of an element $x \in X^N$ to a subset of coordinates $I \subseteq [N]$ is $x\restriction_I = (x_i)_{i \in I}$. The projection of a set $S \subseteq X^N$ to a subset of coordinates $I \subseteq [N]$ is $S\restriction_I = \{x\restriction_I \mid x \in S\}$. The marginal distribution of a random variable A over a set $S \subseteq X^N$ on a subset of coordinates $I \subseteq [N]$ is $A\restriction_I$. We omit the projection symbol and write x_I when it is clear from context.

A set system $\mathcal{F}$ over A is r-spread if for all $Z \subset A$, it holds that

$$\Pr_{S \sim \mathcal{U}(\mathcal{F})}[Z \subset S] \leq r^{-|Z|} \tag{6}$$

The following robust sunflower lemma appears was proven by Rao [Rao20, Lemma 4], although with a different distribution on sets W, and was reproven by Tao [Tao20, Proposition 5] and Bell et al. [BCW21, Theorem 3] in the exact form we need.

Lemma 2.3 ([Rao20, Tao20, BCW21])**.** *There is a universal constant K such that the following holds. Let $0 < p, \epsilon \leq 1/2$, $s \geq 2$, $r \geq K \cdot (1/p) \cdot \log(s/\epsilon)$. Let $\mathcal{F}$ be a set system over A such that:*

1. *for all $S \in \mathcal{F}$, $|S| \leq s$; and*

2. *$\mathcal{F}$ is r-spread.*

Then

$$\Pr_{W \sim \mathcal{U}(A,p)}[\forall S \in \mathcal{F},\, S \not\subseteq W] \leq \epsilon \tag{7}$$

Next we use the sunflower lemma to prove the full range lemma, using the same concepts and adapting the proof of Lovett et al. [LMM$^+$22].

The *min-entropy* of a random variable A over a finite set S, is

$$H_\infty(A) := \log \frac{1}{\max_{\beta \in S} \Pr[A = \beta]}. \tag{8}$$

We write $H_\infty(S) = H_\infty(\mathcal{U}(S))$ in an abuse of notation, and note that $H_\infty(S) = \log|S|$.

A set $X \subseteq [m]^N$ has blockwise min-entropy h with respect to a subset $J \subseteq [N]$ if for all $I \subseteq J$ it holds that $H_\infty(\mathcal{U}(X)\restriction_I) \geq h|I|$. Equivalently, if for all $I \subseteq J$ and $\beta \in [m]^I$ it holds that

$$\frac{\log(1/\Pr_{x\sim\mathcal{U}(X)}[x\restriction_I = \beta])}{|I|} \geq h. \tag{9}$$

If J is omitted we assume $J = [N]$.

The component list S_x of a vector $x \in [m]^N$ is a set $S_x = \{(i, x_i) \mid i \in [N]\} \subset [N] \times [m]$. The component list set system of a set $X \subseteq [m]^N$ is $\mathcal{F} = \{S_x \mid x \in X\}$.

The following key observation relates sunflowers and min-entropy, showing that the concepts of blockwise min-entropy and spreadness are equivalent.

Claim 2.4 ([LMM+22]). *A set $X \subseteq [m]^N$ has blockwise min-entropy $h = \log r$ iff the component list set system $\mathcal{F}$ of X is r-spread.*

Claim 2.5 (Monotonicity). *For all $X \subseteq [m]^N$, $\beta\colon X \to \Sigma^N$ and $\gamma \in \Sigma^N$ it holds*

$$|y \in (\Sigma^m)^N : \forall x \in X, y[x] \neq \beta(x)| \leq |y \in (\Sigma^m)^N : \forall x \in X, y[x] \neq \gamma| \tag{10}$$

Lemma 2.6 (Full Range Lemma). *Let $X \times Y \subseteq [m]^N \times (\Sigma^m)^N$ be such that X has block-wise min-entropy $\log r$ and $|Y| > \epsilon \cdot |\Sigma|^{mN}$.*

Let K be a large enough constant. If r, ϵ, and Σ satisfy $r \geq K \cdot |\Sigma| \cdot \log(N/\epsilon)$ then there exists an $x^* \in X$ such that for every $\beta \in \Sigma^N$, there exists a $y_\beta \in Y$ such that $\mathsf{Ind}_m^N(x^*, y_\beta) = \beta$.

Proof. Let $\mathcal{F}$ be the component list set system of X, which is r-spread by Claim 2.4. Let $p = 1/|\Sigma|$. We can apply Lemma 2.3 with $s = N$ and get that:

$$\Pr_{S_y\sim\mathcal{U}([mN],p)}[\forall S_x \in \mathcal{F}, S_x \not\subseteq S_y] \leq \epsilon\,. \tag{11}$$

Let y be the indicator vector for S_y. This implies:

$$\Pr_{y\sim\mathcal{D}_1(\{0,1\}^{mN},p)}[\forall x \in X, y[x] \neq 1^N] \leq \epsilon\,. \tag{12}$$

Suppose, for the sake of contradiction, that for all $x \in X$ there exists $\beta_x \in \Sigma^N$ such that for all $y \in Y$, $y[x] \neq \beta_x$. By counting:

$$\begin{aligned}
|Y| &\leq |\{y \in (\Sigma^m)^N : \forall x \in X, y[x] \neq \beta_x\}| && (13)\\
&\leq |\{y \in (\Sigma^m)^N : \forall x \in X, y[x] \neq 1^N\}| && (14)\\
&= |\Sigma|^{mN} \cdot \Pr_{y\sim\mathcal{U}(\Sigma^{mN})}[\forall x \in X, y[x] \neq 1^N] && (15)\\
&= |\Sigma|^{mN} \cdot \Pr_{y\sim\mathcal{D}_1(\{0,1\}^{mN},p)}[\forall x \in X, y[x] \neq 1^N] && (16)\\
&\leq |\Sigma|^{mN} \cdot \epsilon\,, && (17)
\end{aligned}$$

where the first inequality is by our assumption on Y, the second inequality is by Claim 2.5, and the last inequality is by Equation 12. This contradicts the bound on $|Y|$. □

In particular, the Full Range Lemma holds when $r \geq m^\delta \geq K \cdot |\Sigma| \cdot 4w \cdot \log(mn^2)$ and $\epsilon \leq 2^{-4w\log mn}$, which is the setting in which we apply it.

2.2 Triangle Lemma

The triangle lemma shows how to cover each triangle in the protocol with sets of well-structured rectangles plus error rows and columns. The proof of this lemma is largely oblivious of the exact shape of the columns, which only comes into play when we need to prove that the set of error columns is not too large. The only changes we need to make are to consider the set of columns Y as a subset of Σ^{mn} rather than $\{0,1\}^{mn}$, and the following calculation. For sets $A \subseteq B \neq \emptyset$, the *density* of A in B is $|A|/|B|$.

Claim 2.7. *Consider a collection of sets indexed by $\alpha \in ([m] \cup \{\star\})^n$, with $|\mathrm{fix}(\alpha)| \leq w$, and $\beta \in (\Sigma \cup \{\star\})^n$, with $\mathrm{fix}(\alpha) = \mathrm{fix}(\beta)$. If every set has density at most $2^{-4w \log mn}$, then the density of the whole collection is at most $2^{-w \log mn}$.*

Proof. There are at most $\sum_{i=0}^{w} \binom{n}{i} m^i \leq (mn)^{w+1} \leq (mn)^{2w}$ many choices of α. Given a fixed α, there are at most $|\Sigma|^{|\mathrm{fix}(\alpha)|} \leq (mn)^w$ many choices of β. By a union bound, the density of the union of all sets is at most $2^{-4w \log mn} \cdot (mn)^{2w} \cdot (nm)^w = 2^{-w \log mn}$. □

2.3 Simulation

An informal sketch of how to extract a subcube-DAG from a triangle-DAG is as follows. We start at the root of the triangle-DAG, which corresponds to no queried variables. We can ensure this is the case only if the protocol is small with respect to the size of the error sets, which results in a small enough cumulative error. With the help of the triangle lemma, we traverse the triangle-DAG while maintaining a well-structured subset of the current triangle T. If the block min-entropy of the well-structured subrectangle becomes larger with respect to a subset of variables I' that is different from the currently queried variables I, then we do the following operation. First we forget all variables in $I \setminus I'$, then we query all variables in $I' \setminus I$. We use the full range lemma to guarantee that the answer to our queries is compatible with T, and the answers themselves to choose which child of T to follow in our traversal.

Again, nothing of substance changes when we use a larger alphabet. We rephrase the simulation to produce a subcube-DAG rather than a resolution refutation, reverting to the original formulation of DAG-like lifting [GGKS20]. This allows us to avoid syntactic issues with the negation of a term not being equivalent to a clause in multi-valued logic.

We provide a full proof in an appendix to the full version of this article.

2.4 Gadget Size

It is not possible to remove the linear dependence of the gadget size in terms of the alphabet size. We exhibit an explicit counterexample to lifting theorems with large alphabet and small gadgets.

Proposition 2.8. *There is a relation $R\colon \Sigma^n \to \Sigma$ with width $\Omega(n)$ but such that if $m = o(|\Sigma|)$, then $R \circ \mathsf{Ind}_m^n$ has a protocol of size $O(|\Sigma|)$.*

Proof. Let R be defined as $(x, y) \in R$ iff $|\{i \in [n] \mid x_i = y\}| \leq n/2$. On the one hand, the certificate complexity of the relation is $n/2$, since every certificate must comprise at least $n/2$ coordinates where $x_i \neq y$. Certificate complexity lower bounds width. On the other hand, if $m < |\Sigma|/2$, then there exists an element $y \in \Sigma$ that appears less than $n/2$ times in Bob's input to $R \circ \mathsf{Ind}_m^n$. Hence a communication protocol is for Bob to simply announce that element, which gives a protocol of depth $\log |\Sigma|$ and size $|\Sigma|$. □

Note that this example does not rule out a sublinear dependence when $\Sigma = \exp(n)$, nor does it rule out an additive dependence.

3 Reductions

Let $G = ((P \cup H), E)$ be a bipartite graph with uniform left degree d. We assume G is represented with an adjacency list (enforcing a fixed order on neighbours) and we denote the i-th neighbour of $p \in P$ as $\Gamma(p, i)$. The (functional) graph pigeonhole principle states that every left vertex maps to exactly one right vertex, and that the mapping is injective. We can encode this as a CSP, denoted cPHP(G), with variables x_p for $p \in P$, with the intended meaning that $z_p = i$ if p is mapped to its ith neighbour. The constraints of the CSP express injectivity: $[z_p \neq i] \vee [z_{p'} \neq i']$ for each pair $p \neq p'$ and i, i' such that $\Gamma(p, i) = \Gamma(p', i')$. Totality and functionality are implicit from the choice of variables.

Specialised to cPHP(G), the falsified constraint search problem $\mathsf{Search}(\text{cPHP}(G))\colon [d]^{|P|} \times (P \times P)$ is the relation containing $(z, (p, p'))$ if $\Gamma(p, z_p) = \Gamma(p', z_{p'})$; in other words, if there exists $i, i' \in [d]$ such that $\Gamma(p, i) = \Gamma(p', i')$ and the constraint $[z_p \neq i] \vee [z_{p'} \neq i']$ is violated. Recall that if cPHP(G) is unsatisfiable, then $\mathsf{Search}(\text{cPHP}(G))$ is a total search problem.

The monotone Karchmer–Wigderson game of a (partial) monotone Boolean function f is a communication problem mKW$_f$ where Alice gets an input $x \in f^{-1}(0)$, Bob gets an input $y \in f^{-1}(1)$, and the set of correct outputs are coordinates $i \in [n]$ such that $x_i < y_i$. The monotone (real) circuit size complexity of f is equivalent to the rectangle-DAG (resp. triangle-DAG) size complexity of mKW$_f$ [Sok17b, HP18].

Lemma 3.1. *Let $G = ((P \cup H), E)$ be a bipartite graph with $|P| = k$ and $|H| = c < k$ and uniform left degree d. Then for any m it holds that $\mathsf{Search}(\text{cPHP}(G)) \circ \mathsf{Ind}_m^k$ reduces to $\text{mKW}_{(\text{clique}_k\text{-col}_c)_{mk}}$.*

Proof. We need to show that there exists a function $\mu_A\colon [m]^k \to \{0,1\}^{\binom{mk}{2}}$ mapping Alice's input and a function $\mu_B\colon [d]^{mk} \to \{0,1\}^{\binom{mk}{2}}$ mapping Bob's input for $\mathsf{Search}(\text{cPHP}(G)) \circ \mathsf{Ind}_m^k$ to their respective inputs for $\text{mKW}_{(\text{clique}_k\text{-col}_c)_{mk}}$.

Moreover, we need a function $\mu_O\colon \{0,1\}^{\binom{mk}{2}} \to P \times P$ mapping from answers of $\text{mKW}_{(\text{clique}_k\text{-col}_c)_{mk}}$ to answers of $\mathsf{Search}(\text{cPHP}(G)) \circ \mathsf{Ind}_m^k$ such that $\forall x, y$ it holds that

$$\mu_O(\text{mKW}_{(\text{clique}_k\text{-col}_c)_{mk}}(\mu_A(x), \mu_B(y))) \subseteq \mathsf{Search}(\text{cPHP}(G)) \circ \mathsf{Ind}_m^k(x, y)\,. \tag{18}$$

We start by defining μ_A and μ_B. Both functions output a grid graph on the vertex set $V = \{(s, p) : s \in [m], p \in [k]\}$; however, μ_A outputs a graph containing a k-clique and μ_B outputs a c-colourable graph.

Alice's input to $\mathsf{Search}(\text{cPHP}(G)) \circ \mathsf{Ind}_m^k$ are pointers $x_p \in [m]$ for $p \in [k]$. Alice's graph (V, E_A) output by μ_A is defined as $E_A = \{((x_p, p), (x_{p'}, p')) : p, p' \in [k], p \neq p'\}$; in other words, E_A is precisely a clique of size k over the vertices $(x_p, p)_{p \in [k]}$.

Bob's input to $\mathsf{Search}(\text{cPHP}(G)) \circ \mathsf{Ind}_m^k$ are strings $y_p \in [d]^m$ for $p \in [k]$. Bob's graph (V, E_B) output by μ_B is the maximal graph that admits the following c-colouring: vertex (s, p) is coloured $\Gamma(p, i)$ for $i = \mathsf{Ind}(s, y_p)$, that is, (s, p) is coloured q, where q is the $\mathsf{Ind}(s, y_p)$-th neighbour of p in G.

Finally, we define $\mu_O((s, p), (s', p')) = (p, p')$. It remains to argue that Equation 18 is satisfied. By definition of $\text{mKW}_{(\text{clique}_k\text{-col}_c)_{mk}}(\mu_A(x), \mu_B(y))$, we have that $((s, p), (s', p'))$ is an edge that belongs to Alice's graph but not to Bob's. We need to argue that (p,p') is a valid answer to $\mathsf{Search}(\mathrm{cPHP}(G))$ on input $z = \mathsf{Ind}^k_m(x,y)$, that is, we must show that for $z_p = \mathsf{Ind}(x_p, y_p)$ and $z_{p'} = \mathsf{Ind}(x_{p'}, y_{p'})$ it holds that $\Gamma(p, z_p) = \Gamma(p', z_{p'})$. By construction of Bob's graph, $\Gamma(p,i) = \Gamma(p', i')$, for $i = \mathsf{Ind}(s, y_p)$ and $i' = \mathsf{Ind}(s', y_{p'})$, and by construction of Alice's graph, $s = x_p$ and $s' = x_{p'}$. Together this implies that $\Gamma(p, z_p) = \Gamma(p', z_{p'})$, as we wished to prove. □

Given two graphs $G_a = (V, E_a)$ $G_b = (V, E_b)$ with $E_a \cap E_b = \emptyset$ and $c, k \in \mathbb{N}$ with $c < k$, we define the function $\mathrm{clique}_k\text{-}\mathrm{col}_c(G_a, G_b) : \{0,1\}^{|E_a|}$ to be the function that receives as input a subset of edges E of E_a and outputs 0 if $E \cup E_b$ is c-colourable and 1 if $E \cup E_b$ has a k-clique. Observe that $\mathrm{clique}_k\text{-}\mathrm{col}_c(G_a, G_b)$ is a restriction of the function $(\mathrm{clique}_k\text{-}\mathrm{col}_c)_{|V|}$ where all the edges in E_b are set to 1 and all the edges in neither E_a nor E_b are set to 0.

Let $G = ((P \,\dot\cup\, H), E)$ be a bipartite graph with $P = \{p_1, \ldots, p_k\}$. We define the graphs $\widetilde{G}_{a,m} = (\widetilde{V}, \widetilde{E}_a)$ and $\widetilde{G}_{b,m} = (\widetilde{V}, \widetilde{E}_b)$ to be k-partite graphs with vertex set $\widetilde{V} = V_1 \,\dot\cup\, V_2 \,\dot\cup \ldots \dot\cup\, V_k$, each V_i with m vertices, and edge sets

$$\widetilde{E}_a = \{(u,v) \in V_i \times V_j : i,j \in [k], i \neq j, \exists h \in H \text{ s.t. } (p_i, h), (p_j, h) \in E\}; \text{ and} \tag{19}$$

$$\widetilde{E}_b = \{(u,v) \in V_i \times V_j : i,j \in [k], i \neq j, \nexists h \in H \text{ s.t. } (p_i, h), (p_j, h) \in E\}\,. \tag{20}$$

Note that, if G has left degree at most $d = O(1)$, then for each i, there are at most d^2 different j's such that $\exists h \in H$ s.t. $(p_i, h), (p_j, h) \in E$. Therefore, the graph $\widetilde{G}_{m,a}$ has at most $m^2 \cdot kd^2$ edges.

Lemma 3.2. *Let $G = ((P \cup H), E)$ be a bipartite graph with $|P| = k$ and $|H| = c < k$ and uniform left degree d. For any $m \in \mathbb{N}$ it holds that $\mathsf{Search}(\mathrm{cPHP}(G)) \circ \mathsf{Ind}^k_m$ reduces to $\mathrm{mKW}_{\mathrm{clique}_k\text{-}\mathrm{col}_c(\widetilde{G}_{a,m}, \widetilde{G}_{b,m})}$.*

Proof. The proof is exactly the same as for Lemma 3.1, with the only difference that some of the edges do not need to be defined by the reductions μ_A and μ_B. Note that the edges not in either $\widetilde{E}_a$ nor $\widetilde{E}_b$ are always set to 0 in Alice's graph, and the edges in $\widetilde{E}_b$ are always set to 1 in Bob's graph, so these edges are never a solution output by μ_O. □

Observation 3.3. *Let $c_1, c_2, n \in \mathbb{N}$ be such that $c_1 c_2 < n$. We have that $\mathrm{mKW}_{(\mathrm{col}_{c_1}\text{-}\mathrm{cocol}_{c_2})_n}$ reduces to $\mathrm{mKW}_{(\mathrm{clique}_{n/c_2}\text{-}\mathrm{col}_{c_1})_n}$.*

Lemma 3.4. *Let $c, r, k, n \in \mathbb{N}$ be such that $c \leq r < k \leq n$. It holds that $\mathrm{mKW}_{(\mathrm{clique}_k\text{-}\mathrm{col}_c)_n}$ reduces to $\mathrm{mKW}_{(\mathrm{col}_r\text{-}\mathrm{cocol}_{n-k+r})_{nr}}$.*

Proof. We define functions $\mu_A, \mu_B \colon \{0,1\}^{\binom{n}{2}} \to \{0,1\}^{\binom{nr}{2}}$ such that μ_B maps c-colourable to r-colourable graphs and μ_A maps graphs containing a k-clique to graphs whose complement is $(n - k + r)$-colourable.

Suppose Alice's input for $\mathrm{mKW}_{(\mathrm{clique}_k\text{-}\mathrm{col}_c)_n}$ is the graph $G_A = (V, E_A)$ and Bob's input is $G_B = (V, E_B)$. The graphs output by the functions μ_A, μ_B are defined on vertex set $U = \{(v,i) : v \in V, i \in [r]\}$ as follows.

Let $\chi\colon V \to [c]$ be a proper c-colouring of G_B. Bob's graph $(U, \widetilde{E}_B)$ output by μ_B is the maximal graph that admits the following r-colouring: vertex (v,i) is coloured with colour $(\chi(v) + i) \bmod r$. Since this graph admits an r-colouring it is a valid input for Bob for $\mathrm{mKW}_{(\mathrm{col}_r\text{-}\mathrm{cocol}_{n-k+r})_{nr}}$.

Let $K \subseteq V$ be a set of k vertices such that $G_A[K]$ is a clique. Alice's graph $(U, \widetilde{E}_A)$ output by μ_A contains edges $\widetilde{E}_A = \{((v, i_1), (v, i_2)) : v \notin K, i_1, i_2 \in [r], i_1 \neq i_2\} \cup \{((v_1, i), (v_2, i)) : v_1, v_2 \in K, v_1 \neq v_2, i \in [r]\}$. Note that this graph is formed by $n - k + r$ disjoint cliques and, therefore, it is a valid input for Alice for $\mathrm{mKW}_{(\mathrm{col}_r\text{-}\mathrm{cocol}_{n-k+r})_{nr}}$.

Finally, we define the function $\mu_O(((v_1, i_1), (v_2, i_2))) = (v_1, v_2)$ mapping outputs of $\mathrm{mKW}_{(\mathrm{col}_r\text{-}\mathrm{cocol}_{n-k+r})_{nr}}$ to outputs of $\mathrm{mKW}_{(\mathrm{clique}_k\text{-}\mathrm{col}_c)_n}$. It remains to argue that valid answers are mapped to valid answers. Let $((v_1, i_1), (v_2, i_2))$ be an edge present in Alice's graph and absent in Bob's graph. By Alice's construction, either $v_1 = v_2$ or $i_1 = i_2$. By Bob's construction, if $v_1 = v_2$ and $i_1 \neq i_2$ then $(\chi(v_1) + i_1) \bmod r \neq (\chi(v_2) + i_2) \bmod r$ and thus there is an edge between (v_1, i_1) and (v_2, i_2). It must therefore be the case that $i_1 = i_2$, which implies that $v_1, v_2 \in K$ and that $\chi(v_1) \neq \chi(v_2)$. We conclude that (v_1, v_2) is a valid answer for $\mathrm{mKW}_{(\mathrm{clique}_k\text{-}\mathrm{col}_c)_n}$. □

Lemma 3.5. *Let $c_1, c_2, \ell_1, \ell_2, n \in \mathbb{N}$ be such that $c_1c_2 < n$ and $\ell_1, \ell_2 \geq 1$. It holds that $\mathrm{mKW}_{(\mathrm{col}_{c_1}\text{-}\mathrm{cocol}_{c_2})_n}$ reduces to $\mathrm{mKW}_{(\mathrm{col}_{\ell_1 c_1}\text{-}\mathrm{cocol}_{\ell_2 c_2})_{\ell_1\ell_2 n}}$.*

Proof. Given the symmetry of $\mathrm{mKW}_{(\mathrm{col}_{c_1}\text{-}\mathrm{cocol}_{c_2})_n}$, it is enough to reduce $\mathrm{mKW}_{(\mathrm{col}_{c_1}\text{-}\mathrm{cocol}_{c_2})_n}$ to $\mathrm{mKW}_{(\mathrm{col}_{\ell c_1}\text{-}\mathrm{cocol}_{c_2})_{\ell n}}$.

The reduction is given by functions $\mu_A, \mu_B \colon \{0,1\}^{\binom{n}{2}} \to \{0,1\}^{\binom{n\ell}{2}}$ such that μ_A maps a c_1-colourable graph into an ℓc_1-colourable graph, and μ_B maps a graph whose complement is c_2-colourable into a graph whose complement is c_2-colourable.

Suppose Alice's input for $\mathrm{mKW}_{(\mathrm{col}_{c_1}\text{-}\mathrm{cocol}_{c_2})_n}$ is the graph $G_A = (V, E_A)$ and Bob's input is $G_B = (V, E_B)$. The graphs output by the functions μ_A, μ_B are defined on vertex set $U = \{(v, i) : v \in V, i \in [\ell]\}$ as follows. Alice's graph $(U, \widetilde{E}_A)$ has edges $\widetilde{E}_A = \{((v_1, i_1), (v_2, i_2)) : (v_1, v_2) \in E_A, i_1, i_2 \in [\ell]\}$, and admits the colouring $\xi(v, i) = \ell\xi(v) + i$. Bob's graph $(U, \widetilde{E}_B)$ has edges $\widetilde{E}_B = \{((v_1, i_1), (v_2, i_2)) : i_1 \neq i_2 \text{ or } (v_1, v_2) \in E_B\}$. Since its complement consists of ℓ independent copies of the complement of G_B, it is c_2-colourable.

Finally, we map outputs according to the function $\mu_O(((v_1, i_1), (v_2, i_2))) = (v_1, v_2)$. To show that the reduction is correct, we observe that if $((v_1, i_1), (v_2, i_2))$ is not present in Bob's graph, then it must be the case that $i_1 = i_2$ and $(v_1, v_2) \notin E_B$, while if $((v_1, i_1), (v_2, i_2))$ is present in Alice's graph, then it must be that $(v_1, v_2) \in E_A$. □

The CSP bitPHP^n_c for n pigeons and c holes is defined over $n \log c$ binary variables and has a constraint expressing $\neg \bigwedge_{j \in [\log c]} (x_{i,j} = x_{i',j})$, expanded into CNF, for each $i \neq i' \in [n]$.

Lemma 3.6 ([HP17])**.** *Let $c_1, c_2, \ell_1, \ell_2, n \in \mathbb{N}$ be such that $c_1c_2 < n$. Partition the inputs of $\mathsf{Search}^{X,Y}(\mathrm{bitPHP}^n_{c_1c_2})$, where X is the first $\log c_1$ variables and Y is the last $\log c_2$ variables of each pigeon. Then $\mathrm{mKW}_{(\mathrm{col}_{c_1}\text{-}\mathrm{cocol}_{c_2})_n}$ and $\mathsf{Search}^{X,Y}(\mathrm{bitPHP}^n_{c_1c_2})$ are equivalent.*

Proof. There is a direct correspondence between problems. An instance of the communication problem $\mathsf{Search}^{X,Y}(\mathrm{bitPHP}^n_{c_1c_2})$ can be viewed as two colourings of a graph on n vertices: the first one is a colouring with c_1 colours, and the second with c_2 colours. Bob's graph can be constructed by taking the maximal graph respecting the c_1-colouring; and Alice's graph can be constructed by taking the minimal graph such that the c_2-colouring is a valid cocolouring. An edge present in Alice's graph and missing in Bob's corresponds precisely to two rows of their input which are the same.

The other direction is similar. Let χ_1 be a colouring of Bob's graph, and χ_2 be a cocolouring of Alice's graph. Alice and Bob can create a matrix with n rows such that Alice's (resp. Bob's) part of row i is $\chi_2(i)$ (resp. $\chi_1(i)$) written in binary. Two rows that are the same correspond precisely to an edge present in Alice's graph and missing in Bob's. □

4 Applications

In this section, we prove the theorems stated in the introduction, which are all obtained via an application of the lifting theorem to the CSP cPHP(G) for an appropriate expander graph G, and the appropriate reduction from section 3.

To obtain a width lower bound we need the bipartite graph G, over which cPHP(G) is defined, to be a good vertex expander as per the following definition. A *(m, n, d, r, e)-bipartite expander graph* is a bipartite graph $G = ((U \cup V), E)$ with $|U| = m$, $|V| = n$, with every vertex in U having degree at most d, and where for every subset $S \subseteq [m]$ such that $|S| \leq r$, the neighbourhood of S, denoted $N(S)$, satisfies $|N(S)| \geq e|S|$. A standard calculation [HLW06] shows that random graphs are good expanders. We record two different parameter regimes that we use in our applications.

The first family of expander graphs have constant degree, and are fairly balanced.

Lemma 4.1. *Let $m, n \in \mathbb{N}$ be such that $m = \Theta(n)$. Then there exists $d = O(1)$ and $r = \Omega(n)$ such that with high probability a random graph $G \sim \mathbf{G}(m, n, d)$ is an $(m, n, d, r, d/2)$-bipartite expander graph.*

The second family has logarithmic degree and can be very unbalanced.

Lemma 4.2. *Let $r, d, n, m \in \mathbb{N}$ be such that $n \leq m$, $d = 6 \ln m$ and $r = n/de^3$. With high probability a random graph $G \sim \mathbf{G}(m, n, d)$ is an $(m, n, d, r, d/2)$-bipartite expander graph.*

We can easily obtain a width lower bound for refuting cPHP(G) from known width lower bounds on the $\{0, 1\}$-encoding of the graph pigeonhole principle [BW01] at the cost of a linear factor in the degree of the graph. We prefer, however, to prove the lower bound directly and avoid such loss.

Let $G = ((U \cup V), E)$ be a bipartite graph. A collection of matchings $\mathcal{M}$ is an s-online matching if $\emptyset \in \mathcal{M}$; $\mathcal{M}$ is closed under taking subsets; and if for each matching $M \in \mathcal{M}$ such that $|M| < s$, and for each $u \in U$ not covered by M there exists $v \in V$ such that $M \cup \{(u, v)\} \in \mathcal{M}$.

Lemma 4.3 ([FFP88])**.** *If G is an (m, n, d, r, e)-bipartite expander graph, then it has an $(e - 1)r/2$-online matching.*

Since online matchings correspond precisely to adversary strategies in the Prover–Adversary games, we obtain the following, slightly tighter, lower bound. The upper bound is given by a Prover strategy that queries any $n + 1$ left vertices.

Theorem 4.4. *If G is an (m, n, d, r, e)-bipartite expander graph then*

$$\frac{(e-1)r}{2} \leq \mathsf{w}(\mathsf{Search}(\mathrm{cPHP}(G))) \leq n + 1\,.$$

We can now apply our lifting theorem (Theorem 2.1) to cPHP(G) for an appropriate G and obtain the following result.

Theorem 1.3 (Restated) (Clique-colouring)**.** *There exists a universal constant A such that given any $c, k, n \in \mathbb{N}$ satisfying $c < k \leq n$, it holds that any monotone Boolean circuit computing* $(\mathrm{clique}_k\text{-}\mathrm{col}_c)_n$ *must have size*

$$\left(\frac{n}{A \cdot kc \log k \log n}\right)^{\Omega(c)}. \tag{3}$$

If we assume $c = \epsilon k$ for a constant $\epsilon > 0$, then we can get a slightly better lower bound: any monotone Boolean circuit computing $(\text{clique}_k\text{-col}_{\epsilon k})_n$ *must have size*

$$\left(\frac{n}{A \cdot k^2 \log n}\right)^{\Omega(k)} . \tag{4}$$

Proof. For the first part, let G be an $(k, c, d, r, d/2)$-bipartite expander graph for $d = O(\log k)$ and $dr = \Omega(c)$. By Theorem 4.4, we have that $c + 1 \geq \mathsf{w}(\mathsf{Search}(\text{cPHP}(G))) = \Omega(c)$. Applying Theorem 2.1 we conclude that there exists a large enough absolute constant A' such that any triangle-DAG protocol solving $\mathsf{Search}(\text{cPHP}(G)) \circ \text{IND}_m^k$ requires size

$$\left(\frac{m}{A' \cdot d \cdot (c+1) \cdot \log(mk)}\right)^{(d/2-1)r/2} \geq \left(\frac{m}{A \cdot \log k \cdot c \cdot \log(mk)}\right)^{\Omega(c)} . \tag{21}$$

Finally, the result follows from the reduction to clique-colouring given by Lemma 3.1, where we note that $n = mk$.

The second part is similar, but we choose G to be an $(k, \epsilon k, d, r, d/2)$-bipartite expander graph for $d = O(1)$ and $r = \Omega(k)$. The result follows, as before, by applying the lifting theorem (Theorem 2.1) to the width lower bound (Theorem 4.4) and using the reduction to clique-colouring (Lemma 3.1). □

We are now ready to prove Theorem 1.6, restated below.

Theorem 1.6 (Restated). *For $c \leq n^{1/3-\epsilon}$, every monotone function that distinguishes n-vertex graphs G with $\chi(G) \leq c$ from graphs with $\chi(\overline{G}) \leq n/c - 1$ has monotone real circuit complexity $n^{\Omega(c)}$.*

We actually prove a slightly more general statement that we use later on.

Theorem 4.5. *There exists an absolute constant A such that any monotone real circuit computing* $(\text{col}_c\text{-cocol}_{n/c-1})_n$ *has size at least*

$$\left(\frac{n}{A \cdot c^3 \log(n)}\right)^{\Omega(c)} .$$

Proof. It follows from Lemma 3.4, by setting $k = c + 1$ and $r = c$, that there is a reduction from $\text{mKW}_{(\text{clique}_{c+1}\text{-col}_c)_{n'}}$ to $\text{mKW}_{(\text{col}_c\text{-cocol}_{n'-1})_{n'c}}$. By renaming $n = n' \cdot c$ we get that the latter is equivalent to $\text{mKW}_{(\text{col}_c\text{-cocol}_{n/c-1})_n}$.

By Theorem 1.3, this implies that any monotone real circuit computing $(\text{col}_c\text{-cocol}_{n'-1})_{n'c}$ must have size at least

$$\left(\frac{n'}{A' \cdot k^2 \log(n')}\right)^{\Omega(k)} = \left(\frac{n}{A \cdot c^3 \log(n)}\right)^{\Omega(c)} , \tag{22}$$

for some universal constants A and A'. □

We can now easily prove Theorem 1.5 and Theorem 1.8.

Theorem 1.5 (Restated) (Colouring-cocolouring)**.** *Every monotone function that distinguishes $(q^2 + 1)$-vertex graphs G with $\chi(G) \leq q$ from graphs with $\chi(\overline{G}) \leq q$ has monotone real circuit complexity $2^{\Omega((q/\log q)^{1/2})}$.*

Proof. As argued in the proof above, we have that any monotone real circuit computing $(\mathrm{col}_c\text{-}\mathrm{cocol}_{n'-1})_{n'c}$ requires size

$$\left(\frac{n'}{A' \cdot k^2 \log(n')}\right)^{\Omega(k)} \tag{23}$$

for $k = c + 1$. Setting $\ell_1 = (n' - 1)/c$ and applying Lemma 3.5, and setting $c = \epsilon(n'/\log n')^{1/2}$ for $\epsilon = 1/4A'$, we conclude that any monotone real circuit computing $(\mathrm{col}_{n'-1}\text{-}\mathrm{cocol}_{n'-1})_{n'(n'-1)}$ requires size $2^{\Omega\left((n'/\log n')^{1/2}\right)}$. Substituting $q = n' - 1$, gives us the stated bound. □

To formally state Theorem 1.8 we need to define the cutting planes proof system. We choose to define the semantic version, which is simpler and stronger than the syntactic one. A *semantic cutting planes refutation* of a CSP F given by a set of linear inequalities $F = \{A_1, \dots, A_m\}$ is a sequence of linear inequalities $C_1, \dots, C_L$ where C_j is either identical to some A_i, or is semantically implied by C_i and $C_{i'}$ with $i, i' < j$. In the proof below, we use the fact that, given a cutting planes refutation of a CSP F of length L, there exists a triangle-DAG protocol of size L computing $\mathsf{Search}^{X,Y}(F)$ for any partition X, Y [Sok17b, HP18].

Theorem 1.8 (Restated) (Bit pigeonhole principle)**.** *Every cutting planes refutation of the bit pigeonhole principle* bitPHP_N^{N+1} *has size* $2^{\Omega((N/\log N)^{1/3})}$.

Proof. Immediate from Theorem 4.5, Lemma 3.6, and the correspondence between the size of cutting planes proofs and triangle-DAGs. □

Finally, we prove Theorem 1.10, which requires a slightly more careful argument than that in the proof of Theorem 1.3.

Theorem 1.10 (Restated) (Monotone vs non-monotone)**.** *There exists a monotone function* $f \in \mathsf{P}$ *such that any monotone circuit computing* f *is of size at least* $\exp(\Omega(n^{1/3-o(1)}))$.

We define the graphs $\widetilde{G}_{a,m} = (\widetilde{V}, \widetilde{E}_a)$ and $\widetilde{G}_{b,m} = (\widetilde{V}, \widetilde{E}_b)$ to be k-partite graphs with vertex set $\widetilde{V} = V_1 \dot\cup V_2 \dot\cup \dots \dot\cup V_k$, where $|V_i| = m$, and edge sets

$$\widetilde{E}_a = \{(u,v) \in V_i \times V_j : i,j \in [k], i \neq j, \exists h \in H \text{ s.t. } (p_i,h),(p_j,h) \in E\}; \text{ and} \tag{24}$$

$$\widetilde{E}_b = \{(u,v) \in V_i \times V_j : i,j \in [k], i \neq j, \nexists h \in H \text{ s.t. } (p_i,h),(p_j,h) \in E\}. \tag{25}$$

Note that, if G has left degree at most $d = O(1)$, then for each i, there are at most d^2 different j's such that there exists $h \in H$ with $(p_i, h), (p_j, h) \in E$. Therefore, the graph $\widetilde{G}_{m,a}$ has at most $m^2 \cdot kd^2$ edges.

Proof of Theorem 1.10. Let $G = (P \dot\cup H, E)$ be a $(k, k-1, d, r, d/s)$-bipartite expander graph given by Lemma 4.1, with $d = O(1)$ and $r = \Omega(k)$. For any m the following holds.

Let $\widetilde{G}_{a,m} = (\widetilde{V}, \widetilde{E}_a)$ and $\widetilde{G}_{b,m} = (\widetilde{V}, \widetilde{E}_b)$ be the graphs defined for Lemma 3.2. Note that the graph $\widetilde{G}_{m,a}$ has at most $O(m^2 \cdot k)$ edges, so the function $\mathrm{clique}_k\text{-}\mathrm{col}_c(\widetilde{G}_{a,m}, \widetilde{G}_{b,m})$ has $n = O(m^2 \cdot k)$ inputs. By Theorem 2.1 applied to $\mathrm{cPHP}(G)$, and using the reduction in Lemma 3.2, we conclude that any monotone circuit computing $\mathrm{clique}_k\text{-}\mathrm{col}_c(\widetilde{G}_{a,m}, \widetilde{G}_{b,m})$ requires size at least

$$\left(\frac{m}{A \cdot k \cdot \log(mk)}\right)^{\Omega(k)} \tag{26}$$

for a large enough constant A.

By choosing $m = A'k \log k$, for large enough constant A', we have that

$$\left(\frac{m}{A \cdot k \cdot \log(mk)}\right)^{\Omega(k)} \geq \exp(\Omega(k)) \geq \exp\left(\Omega\left(\left(n/\log^2 n\right)^{1/3}\right)\right) \geq \exp(\Omega(n^{1/3-o(1)}))\,. \tag{27}$$

Now let g be the total monotone function in P that extends $(\text{clique}_k\text{-col}_{k-1})_N$ for $N = \binom{km}{2}$ given by Tardos [Tar88]. Consider the function f that is obtained from g by setting all edges in $\widetilde{E}_b$ to 1 and all edges not in either $\widetilde{E}_a$ nor $\widetilde{E}_b$ to 0. Note that this function computes $\text{clique}_k\text{-col}_c(\widetilde{G}_{a,m}, \widetilde{G}_{b,m})$. Moreover, since it is a restriction of a function in P, it follows that $f \in \mathsf{P}$. This concludes the proof. □

5 Concluding Remarks

While the lower bounds we proved are better than what is currently known via the approximation method, we believe that it is possible to prove matching bounds with the approximation method. Both methods have common traits: the procedure in which we identify error rectangles in a triangle-DAG by traversing its nodes from the leaves is not too unlike the procedure in which we identify error functions in a monotone circuit, and the best bounds for both are obtained using sunflower lemma. It might even be possible to prove that lifting is a particular instantiation of the approximation method. The advantage of our approach is that once the lifting theorem is established, we can apply it as a black box and obtain new lower bounds by simply proving reductions in the communication world.

The lower bounds in this paper have been known, and presented in various occasions, since 2021. Since then, some of the results have been obtained with the approximation method. A very recent paper independently obtained the same bound for clique-colouring using the approximation method and sunflowers [BM25]. A bottleneck counting approach (which is the same as approximation method) was used to directly prove a nearly optimal cutting planes lower bound for a different formula [Sok24]. Extending the bottleneck counting argument, another very recent paper independently obtained the same bound for cutting planes proofs of the bit pigeonhole principle [BW25], even in the weak setting. It may also be possible to obtain an $\exp(\Omega(n^{1/3-\epsilon}))$ lower bound for a monotone function in P using the approximation method [Cav25], albeit for a different function.

References

[AB87] N. Alon and R. B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7:1–22, 1987. doi:10.1007/BF02579196.

[AD08] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008. doi:10.1016/j.jcss.2007.06.025.

[ALWZ20] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM*

SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020, pages 624–630. ACM, 2020. doi:10.1145/3357713.3384234.

[AM04] Kazuyuki Amano and Akira Maruoka. The potential of the approximation method. *SIAM Journal on Computing*, 33(2):433–447, jan 2004. Preliminary version in *FOCS '96*. doi:10.1137/s009753970138445x.

[And85] A. E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Sov. Math., Dokl.*, 31:530–534, 1985.

[And87] A. E. Andreev. A method for obtaining efficient lower bounds for monotone complexity. *Algebra Logic*, 26(1):1–18, 1987. doi:10.1007/BF01978380.

[BCW21] Tolson Bell, Suchakree Chueluecha, and Lutz Warnke. Note on sunflowers. *Discret. Math.*, 344(7):112367, 2021. doi:10.1016/j.disc.2021.112367.

[BEGJ00] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comput.*, 30(5):1462–1484, 2000. doi:10.1137/S0097539799352474.

[BM25] Jarosław Błasiok and Linus Meierhöfer. Hardness of clique approximation for monotone circuits, 2025. URL: https://arxiv.org/abs/2501.09545, arXiv:2501.09545, doi:10.48550/arXiv.2501.09545.

[BU99] Christer Berg and Staffan Ulfberg. Symmetric approximation arguments for monotone lower bounds without sunflowers. *Computational Complexity*, 8(1):1–20, jun 1999. doi:10.1007/s000370050017.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

[BW24] Paul Beame and Michael Whitmeyer. Multiparty communication complexity of collision finding. *CoRR*, abs/2411.07400, 2024. arXiv:2411.07400, doi:10.48550/arXiv.2411.07400.

[BW25] Paul Beame and Michael Whitmeyer. Multiparty communication complexity of collision-finding and cutting planes proofs of concise pigeonhole principles. Technical report, Electronic Colloquium on Computational Complexity (ECCC), 2025. URL: https://eccc.weizmann.ac.il/report/2025/057, arXiv:TR25-057.

[Cav25] Bruno Pasqualotto Cavalar. Private communication, 2025.

[CKR22] Bruno Pasqualotto Cavalar, Mrinal Kumar, and Benjamin Rossman. Monotone circuit lower bounds from robust sunflowers. *Algorithmica*, 84(12):3655–3685, 2022. doi:10.1007/s00453-022-01000-3.

[dRFJ$^+$24] Susanna F. de Rezende, Noah Fleming, Duri Andrea Janett, Jakob Nordström, and Shuo Pang. Truly supercritical trade-offs for resolution, cutting planes, monotone circuits, and weisfeiler-leman. Technical report, Electronic Colloquium on Computational Complexity (ECCC), 2024. URL: https://eccc.weizmann.ac.il/report/2024/185, arXiv:TR24-185.

[dRGR22] Susanna F. de Rezende, Mika Göös, and Robert Robere. Guest column: Proofs, circuits, and communication. *ACM SIGACT News*, 53(1):59–82, 2022. `doi:10.1145/3532737.3532746`.

[dRMN$^+$20] Susanna F. de Rezende, Or Meir, Jakob Nordstrom, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS '20)*, nov 2020. `doi:10.1109/focs46700.2020.00011`.

[dRMN$^+$24] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, and Robert Robere. KRW composition theorems via lifting. *Comput. Complex.*, 33(1):4, apr 2024. Preliminary version in *FOCS '20*. `doi:10.1007/s00037-024-00250-7`.

[dRNV16] Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In *Proceedings of the 57th IEEE Annual Symposium on Foundations of Computer Science (FOCS '16)*, oct 2016. `doi:10.1109/focs.2016.40`.

[FFP88] Paul Feldman, Joel Friedman, and Nicholas Pippenger. Wide-sense nonblocking networks. *SIAM J. Discret. Math.*, 1(2):158–173, 1988. `doi:10.1137/0401018`.

[GGKS20] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory of Computing*, 16(13):1–30, 2020. Preliminary version in *STOC '18*. URL: `http://www.theoryofcomputing.org/articles/v016a013`, `doi:10.4086/toc.2020.v016a013`.

[GKMP20] Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is np-hard. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 68–77. ACM, 2020. `doi:10.1145/3357713.3384248`.

[GKRS19] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference (ITCS '19)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:19, jan 2019. `doi:10.4230/LIPIcs.ITCS.2019.38`.

[GLM$^+$16] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, oct 2016. Preliminary version in *STOC '15*. `doi:10.1137/15M103145X`.

[GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM J. Comput.*, 47(6):2435–2450, 2018. `doi:10.1137/16M1059369`.

[GPW20] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM J. Comput.*, 49(4), 2020. `doi:10.1137/17M115339X`.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Am. Math. Soc., New Ser.*, 43(4):439–561, 2006. doi:10.1090/S0273-0979-06-01126-8.

[HP17] Pavel Hrubes and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 121–131. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.20.

[HP18] Pavel Hrubes and Pavel Pudlák. A note on monotone real circuits. *Inf. Process. Lett.*, 131:15–19, 2018. doi:10.1016/j.ipl.2017.11.002.

[HR00] Danny Harnik and Ran Raz. Higher lower bounds on monotone size. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 378–387. ACM, 2000. doi:10.1145/335305.335349.

[IR21] Dmitry Itsykson and Artur Riazanov. Proof complexity of natural formulas via communication arguments. In *Proceedings of the 36th Annual IEEE Conference on Computational Complexity (CCC '21)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CCC.2021.3.

[Juk97] Stasys Jukna. Finite limits and monotone computations: the lower bounds criterion. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity (CCC '97)*, pages 302–313, 1997. doi:10.1109/ccc.1997.612325.

[Kra97] Jan Krajícek. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62(2):457–486, 1997. doi:10.2307/2275541.

[Kra98] Jan Krajícek. Interpolation by a game. *Math. Log. Q.*, 44:450–458, 1998. doi:10.1002/malq.19980440403.

[KW90] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discret. Math.*, 3(2):255–265, 1990. doi:10.1137/0403021.

[LMM$^+$22] Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 104:1–104:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.104.

[Oli15] Igor Carboni Oliveira. *Unconditional lower bounds in complexity theory.* PhD thesis, Columbia University, USA, 2015. doi:10.7916/D8ZP45KT.

[PR17] Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC '17)*, pages 1246–1255, jun 2017. doi:10.1145/3055399.3055478.

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

[Pud00] Pavel Pudlák. Proofs as games. *Am. Math. Mon.*, 107(6):541–550, 2000. URL: http://www.jstor.org/stable/2589349.

[Rao20] Anup Rao. Coding for Sunflowers. *Discrete Anal.*, 2020:8, 2020. Id/No 2. doi:10.19086/da.11887.

[Raz85] A. A. Razborov. Lower bounds for the monotone complexity of some Boolean functions. *Sov. Math., Dokl.*, 31:354–357, 1985.

[Raz95] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: Mathematics*, pages 205–227, feb 1995. doi:10.1070/im1995v059n01abeh000009.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999. doi:10.1007/s004930050062.

[Rob18] Robert Robere. *Unified lower bounds for monotone computation.* PhD thesis, University of Toronto, Canada, 2018. URL: http://hdl.handle.net/1807/92007.

[Ros14] Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM J. Comput.*, 43(1):256–279, 2014. doi:10.1137/110839059.

[Sok17a] Dmitry Sokolov. Dag-like communication and its applications. In *Proceedings of the 12th International Computer Science Symposium in Russia (CSR '17)*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, jun 2017. doi:10.1007/978-3-319-58747-9_26.

[Sok17b] Dmitry Sokolov. Dag-like communication and its applications. In Pascal Weil, editor, *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2017. doi:10.1007/978-3-319-58747-9_26.

[Sok24] Dmitry Sokolov. Random ($\log n$)-cnf are hard for cutting planes (again). In *Proceedings of the 56th Annual ACM SIGACT Symposium on Theory of Computing (STOC '24)*, pages 2008–2015, 2024. doi:10.1145/3618260.3649636.

[ST00] Janos Simon and Shi-Chun Tsai. On the bottleneck counting argument. *Theoretical Computer Science*, 237(1–2):429–437, apr 2000. Preliminary version in *CCC '97*. doi:10.1016/s0304-3975(99)00321-7.

[Tao20] Terence Tao. The sunflower lemma via shannon entropy, 2020. URL: https://terrytao.wordpress.com/2020/07/20/the-sunflower-lemma-via-shannon-entropy/.

[Tar88] Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, mar 1988. doi:10.1007/bf02122563.

[Weg87] Ingo Wegener. *The complexity of Boolean functions*. Wiley-Teubner, 1987. URL: `http://ls2-www.cs.uni-dortmund.de/monographs/bluebook/`.

A Lifting Theorem

A.1 Set Up: Full Range Lemma and Triangle Lemma

For most of this section the structures of $[m]^n$ and Σ^{mn} are immaterial and we can simply view $[m]^n \times \Sigma^{mn}$ as a product of two sets. Under this perspective, we refer to an $x \in [m]^n$ as a *row* and a $y \in \Sigma^{mn}$ as a *column*. For any set $S \subseteq [m]^n \times \Sigma^{mn}$ we will denote by S^X and S^Y the projection of S to the rows and columns, respectively, that is, $S^X := \{x \in [m]^n \mid (\{x\} \times \Sigma^{mn}) \cap S \neq \emptyset\}$, and $S^Y := \{y \in \Sigma^{mn} \mid ([m]^n \times \{y\}) \cap S \neq \emptyset\}$. In particular, for any rectangle $R \subseteq [m]^n \times \Sigma^{mn}$, we have that $R = R^X \times R^Y$. Given a row $x \in [m]^n$, we denote the set of columns in S along the row x by

$$S[x] \ := \ \{y \in \Sigma^{mn} \mid \{x\} \times \{y\} \in S\} \tag{28}$$

and note that $\{x\} \times S[x] = (\{x\} \times \Sigma^{mn}) \cap S$.

For each triangle $T \subseteq [m]^n \times \Sigma^{mn}$ we fix some arbitrary choice of a_T and b_T for which $T = \{(x, y) \in [m]^n \times \Sigma^{mn} \mid a_T(x) < b_T(y)\}$. For convenience, we arrange all rows in ascending order of a_T from top to bottom and all columns in descending order of b_T from left to right. We call this the *ordering for the triangle T*.

Throughout the proof we consider blockwise partial assignments $\alpha : [n] \to [m] \cup \{*\}$, which we refer to as *pointers*. Let $\text{fix}(\alpha) \ := \ \{i \in [n] \mid \alpha^{-1}(i) \neq *\}$. We say $x \in [n]^m$ *is consistent with* α if $x_i = \alpha(i)$ for all $i \in \text{fix}(\alpha)$. For $X \subseteq [n]^m$, we define

$$X_\alpha := \{x \in X \mid x \text{ is consistent with } \alpha\}. \tag{29}$$

The proof of the lifting theorem relies on sets that have high blockwise min-entropy. In the following definition, $\delta > 0$ is a parameter to be specified later and w is the parameter from Theorem 2.1.

Definition A.1. Say that a set $X \subseteq [m]^n$ is *α-dense* if it has blockwise min-entropy $\delta \log m$ with respect to $[n] \setminus \text{fix}(\alpha)$. We say $X \subseteq [m]^n$ is *α-predense* if X contains an α-dense subset.

The basic objects in our analysis are rectangles of the following type.

Definition A.2. A rectangle $R := X \times Y$ is *α-pre-structured* if

i) X is α-predense; and

ii) $|Y| \geq |\Sigma|^{mn} \cdot 2^{-4w \log mn}$.

We say that R is *α-structured* if condition i) is replaced with R being α-dense.

We restate Lemma 2.6 for convenience. Here, and throughout this section, we omit the subscript m in IND_m, and for $X \subseteq [m]^n$ and $Y \subseteq \Sigma^{mn}$, we denote by $\text{IND}(X, Y) \subseteq \Sigma^n$ the image of $X \times Y$ under the map IND.

Full Range Lemma. *Let $X \times Y \subseteq [m]^N \times (\Sigma^m)^N$ be such that X has blockwise min-entropy $\log r$ and $|Y| > \epsilon \cdot |\Sigma|^{mN}$.*

Let K be a large enough constant. If r, ϵ, and Σ satisfy $r \geq K \cdot |\Sigma| \cdot \log(N/\epsilon)$ then there exists an $x^* \in X$ such that $\mathsf{Ind}(\{x^*\}, Y) = \Sigma^N$.

The following corollary is a simple application of the lemma to (almost) pre-structured rectangles. We denote the subcube of Σ^n defined by $\rho \in (\Sigma \cup \{\star\})^*$ by $C(\rho)$.

Corollary A.3. *Let $m^\delta \geq 2K \cdot |\Sigma| \cdot 4w \cdot \log(2mn^2)$, where K is the constant given by the Full Range Lemma, let α be a pointer, and let $R = R^X \times R^Y$ be a rectangle that satisfies the following two properties (which are a slightly weaker version of Definition A.2):*

1. *$(R^X)_\alpha$ contains a subset D that has blockwise min-entropy $\delta \log m - 1$ with respect to $[n] - \mathrm{fix}(\alpha)$; and*
2. *R^Y has density at least $2^{-4w \log mn - 1}$.*

If for some $\rho \in \Sigma^{\mathrm{fix}(\alpha)}$ it holds that $\mathrm{IND}(x_{\mathrm{fix}(\alpha)}, y_{\mathrm{fix}(\alpha)}) = \rho$ for all $(x, y) \in R$, then there exists an $x^* \in R^X$ such that $\mathrm{IND}(\{x^*\}, Y) = C(\rho)$.

Proof. If $\mathrm{fix}(\alpha) = [n]$, the conclusion follows from the fact that $R \neq \emptyset$ and for any $(x, y) \in R$, $\mathrm{IND}(x, y) = C(\rho)$, which is a singleton set. Hence, we assume $|\mathrm{fix}(\alpha)| < n$.

Let $D \subseteq (R^X)_\alpha$ be a subset witnessing property 1. Let $J := [n] \setminus \mathrm{fix}(\alpha)$. We consider the rectangle $R' := D_J \times R^Y_J \subseteq [m]^{|J|} \times \Sigma^{m|J|}$, where D_J and R^Y_J are the projections of D and of R^Y to the coordinates J. We want to show that the Full Range Lemma is applicable to rectangle R'. For this purpose, observe that the blockwise min-entropy of D_J is $\log r \geq \delta \log m - 1$ by item 1. Secondly, note that

$$\frac{|R^Y_J|}{\Sigma^{m|J|}} \geq \frac{|R^Y|}{\Sigma^{mn}} \geq 2^{-4w \log mn - 1}, \tag{30}$$

where the first inequality follows from the definition of projection (i.e., projection does not decrease density), and the second inequality follows from item 2.

It follows that

$$r \geq m^\delta/2 \geq K \cdot |\Sigma| \cdot 4w \log(2mn^2) \geq K \cdot |\Sigma| \cdot \log\left(|J| \cdot 2^{4w \log mn + 1}\right), \tag{31}$$

thus satisfying the conditions in the Full Range Lemma, whose application to R' gives us an element $x^{**} \in D_J$ such that $\mathrm{IND}(\{x^{**}\} \times R^Y_J) = \Sigma^{|J|}$. The concatenation of α and x^{**} provides the desired element $x^* \in D \subseteq R^X$. □

In order to extract a subcube-DAG from a triangle-DAG, we cover a triangle by a set of pre-structured rectangles (from which Corollary A.3 allows us to extract low-width subcubes), along with a small number of "error" rows and columns. Unlike previous approaches which partition the triangle into rectangles, we will cover the triangle with (potentially overlapping) *strips*—sets of pre-structured rectangles which all share the same rows, along with a set of "secured" rows on which can apply Corollary A.3. This overlapping covering, as opposed to partitioning, allows us to reduce the number of rows and columns which are not within any pre-structured rectangle and hence reduce the error sets.

We now formally define this notion of a strip. For this, let $w \leq n$ be a parameter which corresponds to the width of the subcube-DAG to be extracted. For a triangle T, recall that $T^X \subseteq [m]^n$ and $T^Y \subseteq \Sigma^{mn}$ are the row and column projections of T.

Definition A.4 (Strips)**.** For a triangle $T \subseteq T^X \times T^Y$ a *strip* S of T is a subset of rows $S \subseteq T^X$ that is α-predense for some pointer $\alpha \in ([m] \cup \{\star\})^n$ with $|\mathrm{fix}(\alpha)| \leq w$. Associated with S are the following:

i) A collection of α-pre-structured rectangles $\mathcal{R}^S = \{R_\beta\}_\beta$ indexed by a set of $\beta \in (\Sigma \cup \{\star\})^n$ with $\mathrm{fix}(\beta) = \mathrm{fix}(\alpha)$, where each $R_\beta = S \times Y_\beta$ is such that $\mathrm{IND}(\alpha_{\mathrm{fix}(\alpha)}, y_{\mathrm{fix}(\alpha)}) = \beta_{\mathrm{fix}(\alpha)}$ for all $y \in Y_\beta$. Furthermore, within each R_β there is an "inner" sub-rectangle $R_\beta^{\mathrm{in}} \subseteq R_\beta \cap T$ which is α-structured and fully contained within T.

ii) A subset of rows $\widehat{S} \subseteq S$ which we call the rows *secured* by S.

A depiction of a strip is given in Figure 2. The purpose of the secured rows is described by the following lemma, which states that for any triangle T, we can construct a set of strips such that the associated pre-structured rectangles cover all of T except a small set of error rows—rows that are not secured by any strip constructed—and error columns. We note that the definition of strips depends on parameters n, m, w and, due to the definition of α-pre-structured and α-structured, also on δ.

Triangle Lemma. *For any positive integers m, n and $w \leq n$, and parameter $\delta \in (0, 1)$ and any triangle $T \subseteq T^X \times T^Y \subseteq [m]^n \times \Sigma^{mn}$ there is a set of strips $\mathsf{Strips}(T)$ of T and "error" sets $X_{\mathrm{err}}^T \subseteq [m]^n, Y_{\mathrm{err}}^T \subseteq \Sigma^{mn}$ such that for any $x \in T^X$ one of the following cases holds:*

- Security. *If x is secured by a strip $S \in \mathsf{Strips}(T)$, then $\{x\} \times T[x]$ is covered by the rectangles in $\mathcal{R}^S$ together with the error columns, that is,*

$$\{x\} \times T[x] \subseteq \bigcup_{R \in \mathcal{R}^S} R \cup \left(\{x\} \times Y_{\mathrm{err}}^T\right). \tag{32}$$

- Error. *If x is not secured by any strip in $\mathsf{Strips}(T)$, then $x \in X_{\mathrm{err}}^T$.*
- Maximality. *If there exists a rectangle $R \subseteq T^X \times (T^Y \setminus Y_{\mathrm{err}}^T)$ that is α-pre-structured for some pointer α with $|\mathrm{fix}(\alpha)| \leq w$ and $\mathrm{IND}(R) \subseteq C(\beta)$ for some $\beta \in (\Sigma \cup \{\star\})^n$ with $\mathrm{fix}(\beta) = \mathrm{fix}(\alpha)$, then there exists a strip $S \in \mathsf{Strips}(T)$ with associated pointer α such that $\mathcal{R}^S$ contains a rectangle indexed by β.*

Furthermore, $|X_{\mathrm{err}}^T| \leq m^{n-(1-\delta)w}$ and $|Y_{\mathrm{err}}^T| \leq |\Sigma|^{mn} \cdot 2^{-w \log mn}$.

We defer the proof of the lemma together with the construction of strips to subsection A.4 in favor of first completing the proof of the lifting theorem.

A.2 Proof of Lifting Theorem

Now we prove Theorem 2.1 using the Triangle Lemma and Corollary A.3.

Proof of Theorem 2.1. Let δ be such that $m^\delta = 2K \cdot |\Sigma| \cdot 4w \cdot \log(2mn^2)$, where K is the constant given by the Full Range Lemma. Let Π be any triangle DAG of size $m^{(1-\delta)w}/2$ solving $\mathcal{S} \circ \mathrm{IND}_m^n$. Note that

$$\frac{m^{(1-\delta)w}}{2} = \frac{1}{2} \cdot \left(\frac{m}{2K \cdot |\Sigma| \cdot 4w \cdot \log(2mn^2)}\right)^w \geq \left(\frac{m}{A \cdot |\Sigma| \cdot w \cdot \log(mn)}\right)^w \tag{33}$$

for a large enough constant A. We first remove the error rows and columns from Π as follows.

Error Removal. Sort the triangles of Π in any topological order $T_1, \ldots, T_s$ from the leaves to the root. That is, if T is a child of T' then T comes before T' in the order. We process Π by the following procedure.

Initialize $X^0_{\text{err}} = Y^0_{\text{err}} := \emptyset$. For $i = 1, \ldots, s$ do the following in order:

1. Remove from T_i the error rows and columns accumulated at $i-1$, that is,

$$T_i \leftarrow T_i \setminus \big((X^{i-1}_{\text{err}} \times \Sigma^{mn}) \cup ([m]^n \times Y^{i-1}_{\text{err}})\big). \tag{34}$$

2. Let $X^{T_i}_{\text{err}}$ and $Y^{T_i}_{\text{err}}$ be the X- and Y-error sets, respectively, obtained by applying the Triangle Lemma to T_i.

3. Define $X^i_{\text{err}} := X^{i-1}_{\text{err}} \cup X^{T_i}_{\text{err}}$ and $Y^i_{\text{err}} := Y^{i-1}_{\text{err}} \cup Y^{T_i}_{\text{err}}$.

Note that in this procedure, the children nodes will each contribute some error rows/columns to the parents, and every node remains a triangle, as we have only removed whole rows/whole columns from it. Henceforth, Π will refer to the resulting triangle-DAG after this procedure.

We extract from Π a subcube-DAG protocol solving S by showing that the following two items hold.

- *Subcubes.* We can associate with every triangle T in Π a set $\mathcal{C}(T)$ of subcubes—each of width at most w—such that if T is a leaf of Π then each subcube $C \in \mathcal{C}(T)$ is a certificate that the label of T is a correct output for relation S, and if T is the root then the full cube $X \times Y$ is contained in $\mathcal{C}(T)$.
- *Queries.* If triangle T has children T_1 and T_2 in Π then for each subcube in $\mathcal{C}(T)$ there is a subprotocol with leaves in $\mathcal{C}(T_1) \cup \mathcal{C}(T_2)$ of width and depth w.

We now prove these items.

Subcubes. For each triangle T in Π, apply the Triangle Lemma to obtain a set of strips $\mathsf{Strips}(T)$ of T. We define $\mathcal{C}(T)$ as follows: for each strip $S \in \mathsf{Strips}(T)$ and each pre-structured rectangle $R_\beta \in \mathcal{R}^S$, we include the subcube $C(\beta)$; that is,

$$\mathcal{C}(T) := \bigcup_{S \in \mathsf{Strips}(T)} \left\{ C(\beta) \mid R_\beta \in \mathcal{R}^S \right\}. \tag{35}$$

To see that $C(\beta)$ is a subcube of width at most w, let α with $|\text{fix}(\alpha)| \le w$ be the pointer associated with the strip S. Then Corollary A.3 guarantees that $\text{IND}(R_\beta) = C(\beta)$ where the width of $C(\beta)$ is $|\text{fix}(\alpha)| \le w$.

We now verify that these sets of subcubes satisfy the desired root and leaf properties.

- *Root.* Let $R = R^X \times R^Y$ be the triangle at the root of Π (which is a rectangle, though we won't need this). By the Triangle Lemma and a union bound over the triangles in Π, the density X-error accumulated at the root is at most

$$m^{-(1-\delta)w} \cdot |\Pi'| \le m^{-(1-\delta)w} \cdot m^{(1-\delta)w}/2 = 1/2. \tag{36}$$

Hence R^X has density at least $1/2$. This implies that for any $\emptyset \neq I \subseteq [n]$,

$$H_\infty\left(R_I^X\right) \geq |I| \log m - 1 \geq \delta |I| \log m, \tag{37}$$

and so we have that R^X is $\star^n$-predense.

Similarly, the density of the Y-errors accumulated at the root is at most

$$2^{-w \log mn} \cdot |\Pi| < 2^{w \log m - w \log mn} < 1/2. \tag{38}$$

Therefore, $|R^Y \setminus Y_{\text{err}}^R| \geq |\Sigma|^{mn}/2 \geq |\Sigma|^{mn} \cdot 2^{-4w \log mn}$. We therefore conclude that $R^X \times (R^Y \setminus Y_{\text{err}}^R)$ is a $\star^n$-(pre-)structured rectangle.

By the maximality condition of Triangle Lemma applied to $R^X \times (R^Y \setminus Y_{\text{err}}^R)$, we have that there exists a strip $S \in \mathsf{Strips}(R)$ with associated pointer $\star^n$ and such that the collection $\mathcal{R}^S$ is non-empty. By the item i) of Definition A.4 $\mathcal{R}^S = \{R_\beta\}_\beta$ has to be a singleton set since only the empty string can be a subscript β. Let R_β be the unique rectangle in $\mathcal{R}^S$. Since R_β is $\star^n$-pre-structured, by Corollary A.3, it holds that $\text{IND}(R_\beta) = \Sigma^n = C(\star^n)$. Therefore $C(\star^n) \in \mathcal{C}(R)$.

- *Leaves.* Consider any leaf triangle T of Π. By definition, T is labelled with an output z such that $T \times \{z\} \subseteq \mathcal{S} \circ \mathsf{Ind}$ or, equivalently, $\mathsf{Ind}(T) \times \{z\} \subseteq \mathcal{S}$. Therefore, for any subcube $C(\beta) \in \mathcal{C}(T)$, we have $C(\beta) = \text{IND}(R_\beta) = \text{IND}(R_\beta^{\text{in}}) \subseteq \text{IND}(T) \subseteq \mathcal{S}^{-1}(z)$, meaning that $C(\beta)$ is a valid certificate.

Queries. Let T be any non-leaf triangle in Π with children T_1 and T_2. Consider any subcube $C \in \mathcal{C}(T)$ generated by some pre-structured rectangle $R_\beta = S \times Y_\beta^T$ in a strip S defined from some pointer α. We first show that C is covered by $\mathcal{C}(T_1) \cup \mathcal{C}(T_2)$.

Consider the "inner" structured sub-rectangle $R_\beta^{\text{in}} \subseteq R_\beta \cap T$. Since T is covered by its children T_1 and T_2,

$$R_\beta^{\text{in}} \subseteq T \subseteq T_1 \cup T_2. \tag{39}$$

We claim that at least one of T_1 or T_2 contains a sub-rectangle $Q = Q^X \times Q^Y \subseteq R_\beta^{\text{in}}$ with X- and Y-density at least half that of R_β^{in}. To see this, order the rows/columns according to the ordering of T_1, then the center p of R_β^{in} divides R_β^{in} into four quadrants. If $p \in T_1$ then, as T_1 is a triangle, the top-left quadrant Q of R_β^{in} is contained entirely within T_1; see Figure 1. Otherwise, if $p \notin T_1$, then as T_1 is a triangle, the bottom-right quadrant Q is disjoint from T_1 and so it must be contained within T_2. In either case, $H_\infty(Q^Y) \geq H_\infty(Y_\beta^T) - 1$ and $H_\infty(Q_J^X) \geq H_\infty(S_J) - 1$ for any $\emptyset \neq J \subseteq [n] \setminus I$. In particular, Q satisfies the premises of Corollary A.3. Suppose without loss of generality that $Q \subseteq T_1$.

Applying Corollary A.3 to Q, we get a row $x^* \in Q^X \subseteq T_1^X$ such that $\text{IND}(\{x^*\} \times Q^Y) = C(\beta)$. As we have removed $X_{\text{err}}^{T_1}$ and $Y_{\text{err}}^{T_1}$ from T in the Error Removal step, $x^* \notin X_{\text{err}}^{T_1}$ and $Q^Y \subseteq T^Y$ is disjoint from $Y_{\text{err}}^{T_1}$. Thus, x^* is secured by a strip S' of T_1 defined by some pointer α'. By the Triangle Lemma,

$$\{x^*\} \times Q_Y \subseteq \{x^*\} \times (T_1[x^*] \setminus Y_{\text{err}}^{T_1}) \subseteq \bigcup_{R_\xi \in \mathcal{R}^{S'}} R_\xi, \tag{40}$$

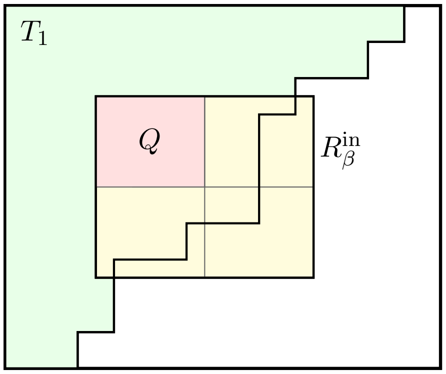

Figure 1: The structured rectangle R_β^{in} for triangle T, whose quadrant Q is contained entirely within child T_1.

where $\mathcal{R}^{S'}$ is the set of pre-structured rectangles in strip S'. By Corollary A.3, each α'-pre-structured rectangle $R_\xi \in \mathcal{R}^{S'}$ satisfies $\mathrm{IND}(R_\xi) = C(\xi)$, and so

$$C = \mathrm{IND}(R_\beta) = \mathrm{IND}(\{x^*\} \times Q_Y) \subseteq \bigcup_{R_\xi \in \mathcal{R}^{S'}} \mathrm{IND}(R_\xi) = \bigcup_{R_\xi \in \mathcal{R}^{S'}} C(\xi). \tag{41}$$

Consider the subprotocol where, having C as the root, we first forget $\mathrm{fix}(\alpha) \setminus \mathrm{fix}(\alpha')$, then query $\mathrm{fix}(\alpha') \setminus \mathrm{fix}(\alpha)$. The leaves of this protocol are $\{C(\xi)\}$, which cover C as we just established. The width and depth of the protocol are at most w. □

A.3 Proof of Full Range Lemma

We prove Claim 2.4 for completeness.

Claim A.5 ([LMM+22])**.** *A set $X \subseteq [m]^N$ has blockwise min-entropy $h = \log r$ iff the component list set system $\mathcal{F}$ of X is r-spread.*

Proof. Assume X has blockwise min-entropy $h = \log r$. Fix $Z \subseteq [N] \times [m]$ and let $I = Z\restriction_{[N]}$, $\beta = Z\restriction_{[m]}$ (assuming I are all distinct). Then

$$\Pr_{S\sim\mathcal{U}(\mathcal{F})}[Z \subset S] = \Pr_{x\sim\mathcal{U}(x)}[x_i = \alpha \ \forall (i,\alpha) \in Z] \tag{42}$$
$$= \Pr_{x\sim\mathcal{U}(x)}[x\restriction_I = \beta \text{ and } \beta \text{ well defined}] \le 2^{-h|Z|} \le r^{-|Z|} \tag{43}$$

hence $\mathcal{F}$ is r-spread.

Assume $\mathcal{F}$ is r-spread. Fix $I \subseteq [N]$ and $\beta \in [m]^I$ and let $Z = \{(i, \beta_i) \mid i \in I\}$. Then

$$\Pr_{x\sim\mathcal{U}(X)}[x\restriction_I = \beta] = \Pr_{S\sim\mathcal{U}(\mathcal{F})}[S\restriction_I = Z] = \Pr_{S\sim\mathcal{U}(\mathcal{F})}[S \subset Z] \le r^{-|Z|} \tag{44}$$

hence X has min-entropy $\log r$. □

A.4 Proof of Triangle Lemma

The rest of this section is dedicated to the proof of the Triangle Lemma. That is, our goal is to describe, for any given parameter w, how to associate with any triangle $T \subseteq T^X \times T^Y$ a set of strips $\mathsf{Strips}(T)$ and error sets which satisfy the *security*, *error* and *maximality* properties of the lemma.

Let parameters w and δ be given. For every pointer α with $|\mathrm{fix}(\alpha)| \le w$ such that the rows in T^X that are consistent with α form an α-predense set, i.e., $(T^X)_\alpha$ is α-predense, we construct a strip $S \coloneqq (T^X)_\alpha$, to be included in $\mathsf{Strips}(T)$, by associating S with the following structures.

- *Secured Rows.* Let $x^S \in S$ be the highest row (according to the ordering of T) such that the elements in S above or equal to x^S form an α-predense set. Let the secured rows $\widehat{S} \subseteq S$ be those below or equal to x^S.
- *Pre-Structured Rectangles.* Generate the set of pre-structured rectangles $\mathcal{R}^S$ as follows: for every $\beta \in (\Sigma \cup \{\star\})^n$ with $\mathrm{fix}(\beta) = \mathrm{fix}(\alpha)$ consider the set of columns
$$Y_\beta \ \coloneqq\ \{y \in \Sigma^{mn} \mid \mathrm{IND}\left(\alpha_{\mathrm{fix}(\alpha)}, y_{\mathrm{fix}(\alpha)}\right) = \beta_{\mathrm{fix}(\alpha)}\}\,. \tag{45}$$
If $|Y_\beta| \ge |\Sigma|^{mn} \cdot 2^{-4w \log mn}$ then we include the rectangle $R_\beta \coloneqq S \times Y_\beta$ in $\mathcal{R}^S$. Otherwise, we include the columns Y_β in a set Y^S_{err}.
- *Inner Rectangle.* It remains to show that we can find some sub-rectangle $R^{\mathrm{in}}_\beta \subseteq R_\beta \cap T$ which is α-structured and contained entirely within T. Since S is α-predense there is some α-dense subset of rows $S' \subseteq S$. Note that by definition S' is only above (and including) x^S, and so the rectangle $R^{\mathrm{in}}_\beta \coloneqq S' \times Y_\beta$ is only above (and including) $\{x^S\} \times Y_\beta \subseteq T$. Hence, as T is a triangle, $R^{\mathrm{in}}_\beta \subseteq T$. Finally, note that as R_β was not categorized as "error", R^{in}_β is α-structured.

Observe that with this construction each strip in $\mathsf{Strips}(T)$ is uniquely determined by a pointer α. Finally, define the associated error sets $X^T_{\mathrm{err}} \subseteq [m]^n$ and $Y^T_{\mathrm{err}} \subseteq \Sigma^{mn}$ as follows:

- *X-Error.* Let X^T_{err} be the set of rows in T^X which are not secured by *any* strip in $\mathsf{Strips}(T)$.
- *Y-Error.* Let Y^T_{err} be collected over all strips $S \in \mathsf{Strips}(T)$, that is, $Y^T_{\mathrm{err}} \coloneqq \bigcup_{S \in \mathsf{Strips}(T)} Y^S_{\mathrm{err}}$.

A depiction of a strip is in Figure 2.

We first argue that the error, security and maximality properties of the Triangle Lemma hold. The error property holds by construction. To see why the security property holds, note that given a strip $S \in \mathsf{Strips}(T)$, the row x^S in T is covered by the α-pre-structured rectangles and the error columns in S. That is,
$$\{x^S\} \times T[x^S] \subseteq \bigcup_{R \in \mathcal{R}^S} R\ \cup \left(\{x^S\} \times Y^T_{\mathrm{err}}\right)\,. \tag{46}$$

Fix any secured row $x \in \widehat{S} \subseteq S$. Then x is below or equal to x^S and, therefore, since T is a triangle, $T[x] \subseteq T[x^S]$. Hence,
$$\{x^S\} \times T[x] \subseteq \bigcup_{R \in \mathcal{R}^S} R\ \cup \left(\{x^S\} \times Y^T_{\mathrm{err}}\right)\,. \tag{47}$$

Now, for the maximality property, assume there is a rectangle $R \subseteq T^X \times (T^Y \setminus Y^T_{\mathrm{err}})$ that is α-pre-structured for some pointer α with $|\mathrm{fix}(\alpha)| \le w$ and $\mathrm{IND}(R) \subseteq C(\beta)$ for some $\beta \in (\Sigma \cup \{\star\})^n$ with

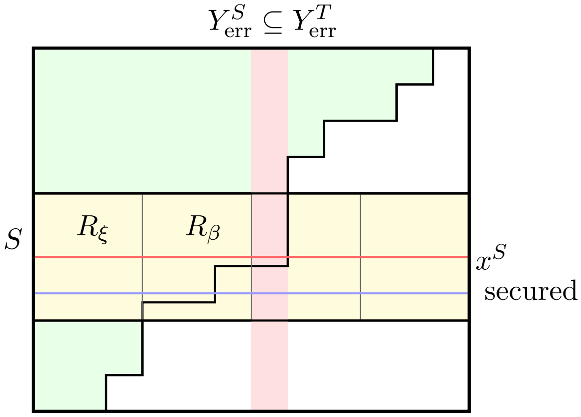

Figure 2: A strip S of a triangle, including two pre-structured rectangles R_ξ, R_β, a set of error columns Y^S_{err}, and an example of a secured row.

$\text{fix}(\beta) = \text{fix}(\alpha)$. Note that $(R^X)_\alpha$ is α-predense and hence so is its superset $(T^X)_\alpha$, thus by our construction there is a strip $S \in \mathsf{Strips}(T)$ associated with α. Since $\text{IND}(R) \subseteq C(\beta)$ it follows that R^Y is a subset of $Y_\beta = \{y \in \Sigma^{mn} \mid \text{IND}(\alpha_{\text{fix}(\alpha)}, y_{\text{fix}(\alpha)}) = \beta_{\text{fix}(\alpha)}\}$ defined in (45). As $R^Y \cap Y^T_{\text{err}} = \emptyset$ and $R^Y \subseteq Y_\beta$ (and $R^Y \neq \emptyset$ since R is α-pre-structured), it must be the case that $Y_\beta \not\subseteq Y^T_{\text{err}}$ and thus, by the construction of *Pre-Structured Rectangles*, it must be the case that $|Y_\beta| \geq |\Sigma|^{mn} \cdot 2^{-4w \log mn}$. Therefore, the rectangle $R_\beta := (T^X)_\alpha \times Y_\beta$ is α-pre-structured and thus, by our construction, is in $\mathcal{R}^S$.

Finally, we bound the size of the error sets, using the following two claims.

Claim A.6. *For any triangle T, the density of X^T_{err} in $[m]^n$ is less than $m^{-(1-\delta)w}$.*

Proof. Suppose for contradiction that X^T_{err} has density at least $m^{-(1-\delta)w}$. For simplicity, we denote $\widehat{X} := X^T_{\text{err}}$. Let $I \subseteq [n]$ be a maximal set of blocks where $\widehat{X}$ is not dense—meaning that $H_\infty(\widehat{\mathbf{X}}_I) < \delta|I| \log m$—and fix any pointer α with $\text{fix}(\alpha) = I$ that witnesses $\Pr[\widehat{\mathbf{X}}_I = \alpha_I] \geq m^{-\delta|I|}$. If no such I exists, we let $I := \emptyset$ and $\alpha = \star^n$. We record the following two basic properties:

(1) $|I| \leq w$,

(2) $\widehat{X}_\alpha := \{x \in X^T_{\text{err}} \mid x_I = \alpha_I\}$ is α-dense.

To see item (1), observe that by the definition of α,

$$|\widehat{X}| \leq \frac{|\widehat{X}_\alpha|}{m^{-\delta|I|}} \leq \frac{|\{x \in [m]^n \mid x_I = \alpha_I\}|}{m^{-\delta|I|}} = m^{n-(1-\delta)|I|}. \tag{48}$$

From this and our assumption that $|\widehat{X}|$ has density at least $m^{-(1-\delta)w}$, it follows that $|I| \leq w$. To prove item (2), we show that if $\widehat{X}_\alpha$ is not α-dense then this contradicts the maximality of I. Indeed, if $\widehat{X}_\alpha$ is not α-dense then there exists a nonempty subset $J \subseteq [n] \setminus I$ and a witness $\alpha' \in ([m] \cup \{\star\})^n$ with $\text{fix}(\alpha') = J$ such that $\Pr_{x \sim \widehat{X}_\alpha}[x_J = \alpha'_J] \geq m^{-\delta|J|}$. Let $\alpha \circ \alpha'$ be the pointer with $\text{fix}(\alpha \circ \alpha') = I \cup J$ such that $(\alpha \circ \alpha')_I = \alpha_I$ and $(\alpha \circ \alpha')_J = \alpha_J$. Then

$$\Pr_{x\sim\widehat{X}}[x_{I\cup J} = (\alpha \circ \alpha')_{I\cup J}] = \Pr_{x\sim\widehat{X}}[x_I = \alpha_I] \cdot \Pr_{x\sim\widehat{X}}[x_J = \alpha'_J \mid x_I = \alpha_I] \tag{49}$$

$$= \Pr_{x\sim\widehat{X}}[x_I = \alpha_I] \cdot \Pr_{x\sim\widehat{X}_\alpha}[x_J = \alpha'_J] \tag{50}$$

$$\geq m^{-\delta(|I|+|J|)}, \tag{51}$$

meaning that $\widehat{X}$ is also not dense on $I \cup J$, which contradicts the maximality of I.

By item (1) and item (2) there is a strip $S \in \mathsf{Strips}(T)$ with associated pointer α, consisting of the rows $x \in T^X$ for which $x_I = \alpha_I$. Note that $\widehat{X}_\alpha \subseteq S$, and since $\widehat{X}_\alpha$ is α-predense, the distinguished row x^S of strip S cannot be strictly below all rows in $\widehat{X}_\alpha$. However, this implies that some row $x \in \widehat{X}_\alpha$ is secured by S. This is a contradiction, as $x \in \widehat{X}_\alpha \subseteq X^T_{\text{err}}$ where X^T_{err} contains only rows of T that are not secured by any strip in $\mathsf{Strips}(T)$. □

Claim A.7. *For any triangle T, the density of Y^T_{err} in Σ^{mn} is at most $2^{-w\log mn}$.*

Proof. Immediate from Claim 2.7. □

This completes the proof of the Triangle Lemma.