Electronic Colloquium on Computational Complexity, Report No. 99 (2020)

KRW Composition Theorems via Lifting

Susanna F. de Rezende* Or Meir† Jakob Nordström‡ Toniann Pitassi§
Robert Robere¶

July 6, 2020

Abstract

One of the major open problems in complexity theory is proving super-logarithmic lower bounds on the depth of circuits (i.e., $\mathbf{P} \not\subseteq \mathbf{NC}^1$). Karchmer, Raz, and Wigderson [KRW95] suggested to approach this problem by proving that depth complexity behaves "as expected" with respect to the composition of functions $f \diamond g$. They showed that the validity of this conjecture would imply that $\mathbf{P} \not\subseteq \mathbf{NC}^1$.

Several works have made progress toward resolving this conjecture by proving special cases. In particular, these works proved the KRW conjecture for every outer function f, but only for few inner functions g. Thus, it is an important challenge to prove the KRW conjecture for a wider range of inner functions.

In this work, we extend significantly the range of inner functions that can be handled. First, we consider the *monotone* version of the KRW conjecture. We prove it for every monotone inner function g whose depth complexity can be lower bounded via a query-to-communication lifting theorem. This allows us to handle several new and well-studied functions such as the s-t-connectivity, clique, and generation functions.

In order to carry this progress back to the *non-monotone* setting, we introduce a new notion of *semi-monotone* composition, which combines the non-monotone complexity of the outer function f with the monotone complexity of the inner function g. In this setting, we prove the KRW conjecture for a similar selection of inner functions g, but only for a specific choice of the outer function f.

*Institute of Mathematics of the Czech Academy of Sciences. Research supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) ERC grant agreement no. 279611, as well as by the Knut and Alice Wallenberg grants KAW 2016.0066 and KAW 2018.0371.

†Department of Computer Science, University of Haifa, Haifa 3498838, Israel. ormeir@cs.haifa.ac.il. Research supported by the Israel Science Foundation (grant No. 1445/16).

‡University of Copenhagen and Lund University. Research supported by the Swedish Research Council grant 2016-00782, the Knut and Alice Wallenberg grant KAW 2016.006, and the Independent Research Fund Denmark grant 9040-00389B.

§Department of Computer Science, University of Toronto, Canada, and Institute of Advanced Study, Princeton, USA. toni@cs.toronto.edu. Research supported by NSERC and by NSF CCF grant 1900460

¶DIMACS and Institute for Advanced Study. Robert Robere was supported by NSERC, the Charles Simonyi Endowment, and indirectly supported by the National Science Foundation Grant No. CCF-1900460.

ISSN 1433-8092

Contents

1 Introduction **4**
- 1.1 Our results . . . 6
  - 1.1.1 The monotone composition theorem . . . 6
  - 1.1.2 The semi-monotone composition theorem . . . 7
- 1.2 Our techniques . . . 10
  - 1.2.1 The monotone composition theorem . . . 10
  - 1.2.2 The semi-monotone composition theorem . . . 11

2 Preliminaries **11**
- 2.1 Communication complexity . . . 12
- 2.2 Subadditive measures on trees . . . 13
- 2.3 Monotone formulas and Karchmer-Wigderson relations . . . 13
- 2.4 Decision trees . . . 15
- 2.5 The Razborov rank measure . . . 15
- 2.6 The Nullstellensatz proof system . . . 15
- 2.7 Lifting theorems . . . 16
  - 2.7.1 Lifting from query complexity . . . 16
  - 2.7.2 Lifting from Nullstellensatz degree . . . 17
- 2.8 Min-entropy . . . 17
- 2.9 Prefix-free codes . . . 18
- 2.10 Degrees of sets of strings . . . 18
- 2.11 Kronecker product . . . 19

3 The monotone composition theorem **20**
- 3.1 Reductions . . . 20
  - 3.1.1 The observation of [KRW95] . . . 20
  - 3.1.2 The problem $mKW_f \circledast S_{\mathrm{gd}}$ . . . 21
- 3.2 The structure theorem . . . 22
  - 3.2.1 Statement of the structure theorem . . . 22
  - 3.2.2 The lower bound on $mKW_f \circledast S_{\mathrm{gd}}$ . . . 23
  - 3.2.3 Proof of structure theorem from lemmas . . . 25
- 3.3 Proof of Lemma 3.13 . . . 27
  - 3.3.1 The initial set $\mathcal{W}_0$ . . . 29
  - 3.3.2 The iterative procedure . . . 30
- 3.4 Proof of Lemma 3.14 . . . 32

4 The semi-monotone composition theorem **33**
- 4.1 The rank of M . . . 36
- 4.2 The rank of monochromatic rectangles . . . 37
- 4.3 The existence of the matrix A . . . 39

5 A generalized lifting theorem **41**
- 5.1 Proof overview . . . 42
- 5.2 Lifting machinery . . . 43
- 5.3 The construction of the decision tree T . . . 45
- 5.4 The query complexity of T . . . 46

6 Open questions **48**

A Composition theorems for classic functions **48**

1 Introduction

A major frontier of the research on circuit complexity is proving super-logarithmic lower bounds on the depth complexity of an explicit function, i.e., proving that $\mathbf{P} \not\subseteq \mathbf{NC}^1$. This question is an important milestone toward proving lower bounds on general circuits, and also captures the natural question of whether there are tractable computational tasks that cannot be parallelized. The state of the art is the work of Håstad [Hås98], who proved a lower bound of $(3 - o(1)) \cdot \log n$, following a long line of work [Sub61, Khr72, And87, PZ93, IN93]. This lower bound has not been improved for more than two decades except for the lower order terms [Tal14], and it is an important problem to break this barrier.

Karchmer, Raz, and Wigderson [KRW95] proposed to approach this problem by studying the (block-)composition of Boolean functions, defined as follows: if $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ are Boolean functions, then their composition $f \diamond g$ takes inputs in $(\{0,1\}^n)^m$ and is defined by

$$f \diamond g(x_1, \ldots, x_m) = f\left(g(x_1), \ldots, g(x_m)\right). \tag{1}$$

Let us denote by $\mathsf{D}(f)$ the minimal depth of a circuit with fan-in 2 that computes f. The circuit that computes $f \diamond g$ using Equation (1) has depth $\mathsf{D}(f) + \mathsf{D}(g)$. Karchmer et. al. [KRW95] conjectured that this upper bound is roughly optimal:

Conjecture 1.1 (The KRW conjecture)**.** *Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant functions. Then*

$$\mathsf{D}(f \diamond g) \approx \mathsf{D}(f) + \mathsf{D}(g). \tag{2}$$

Karchmer et. al. observed that their conjecture, if proved, would imply that $\mathbf{P} \not\subseteq \mathbf{NC}^1$. The meaning of "approximate equality" in Equation (2) is intentionally left vague, since there are a many variants that would imply that separation.

While we are still far from resolving the KRW conjecture, several works [KRW95, EIRS01, HW93, Hås98, GMWW17, DM18, KM18] have made progress toward it by proving special cases. The state of the art is that the KRW conjecture is known to hold for every outer function f, but only when combined with two specific choices of the inner function g: the parity function, and the universal relation. There are no results proving the KRW conjecture for a broader family of inner functions.

In this work, we prove the KRW conjecture for a rich family of inner functions g, namely, those functions whose depth complexity can be lower bounded using *lifting theorems*. This includes functions that are considerably more interesting than previous composition theorems could handle. We prove these results in the *monotone* setting, and in a new setting which we call the *semi-monotone* setting. Below, we discuss the background to this work and present our results.

Karchmer-Wigderson relations. It is useful to study the KRW conjecture through the lens of communication complexity, and in particular, using the framework of *Karchmer-Wigderson relations*. Let us denote the (deterministic) communication complexity of a problem R by $\mathsf{CC}(R)$. The *Karchmer-Wigderson relation* of a function $f : \{0,1\}^n \to \{0,1\}$, denoted KW_f, is the communication problem in which the inputs of Alice and Bob are $x \in f^{-1}(1)$ and $y \in f^{-1}(0)$ respectively, and their goal is to find a coordinate i such that $x_i \neq y_i$. Karchmer and Wigderson [KW90] observed that $\mathsf{D}(f) = \mathsf{CC}(KW_f)$. This connection between functions and communication problems allows us to study the depth complexity of functions using techniques from communication complexity.

The KRW conjecture from the KW perspective. Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant functions. It will be useful to denote the KW relation $KW_{f \diamond g}$ of the composed function by $KW_f \diamond KW_g$. In this relation, Alice and Bob get $X \in (f \diamond g)^{-1}(1)$ and $Y \in (f \diamond g)^{-1}(0)$, viewed as $m \times n$ matrices, and their goal is to find an entry (i,j) such that $X_{i,j} \neq Y_{i,j}$. The KRW conjecture can be restated as:

$$\mathsf{CC}(KW_f \diamond KW_g) \approx \mathsf{CC}(KW_f) + \mathsf{CC}(KW_g).$$

It is worth noting the obvious protocol for solving $KW_f \diamond KW_g$: Let a, b be the column vectors that are obtained from applying g to the rows of X, Y, and observe that they constitute an instance of KW_f. The players begin by solving KW_f on a and b, thus obtaining a coordinate $i \in [m]$ such that $a_i \neq b_i$. Then, they solve KW_g on the rows X_i, Y_i, which constitute an instance of KW_g, thus obtaining a coordinate $j \in [n]$ where $X_{i,j} \neq Y_{i,j}$. The communication complexity of this protocol is $\mathsf{CC}(KW_f) + \mathsf{CC}(KW_g)$, and the KRW conjecture says that this obvious protocol is roughly optimal.

Previous work on the KRW conjecture. The KRW conjecture has been studied extensively, and a long line of papers have made progress on important restricted cases. These papers can be broadly divided into two categories.

The first category involves proving the KRW conjecture for a simplified communication problem. Specifically, Karchmer et. al. [KRW95] proposed a simplification of KW relations called the *universal relation* (denoted U_n) which is the following communication problem: Alice and Bob get two *distinct* strings $x, y \in \{0,1\}^n$, and their goal is to find a coordinate on which they disagree. The universal relation is harder to solve than KW relations, since the inputs of Alice and Bob are not assumed to come from the preimage of some function f, and so the protocol cannot take advantage of any properties of f. Just as the universal relation is a simplified version of KW relations, one can define simplified versions of $KW_f \diamond KW_g$, such as the composition $U_m \diamond U_n$ of two universal relations and the composition $KW_f \diamond U_n$ of a KW relation and a function. Several works have studied this type of compositions [KRW95, EIRS01, HW93, GMWW17, KM18], and the state of the art is that the KRW conjecture holds for $KW_f \diamond U_n$ for every non-constant function $f : \{0,1\}^m \to \{0,1\}$ [GMWW17, KM18].

The second category where important progress was made is for $KW_f \diamond KW_{\bigoplus}$ where f can be any non-constant function and $\bigoplus$ is the parity function. The KRW conjecture for this case has been proved implicitly by Håstad [Hås98], and an alternative proof was recently given by Dinur and Meir [DM18].

The papers discussed so far are able to handle an arbitrary choice of the outer relation KW_f, but only very specific choices of the inner relation KW_g. This seems to suggest that the crux of the difficulty in proving the KRW conjecture lies in having to deal with an arbitrary choice of KW_g. In order to bypass this difficulty, Meir [Mei19] recently observed that in order to prove that $\mathbf{P} \not\subseteq \mathbf{NC}^1$, it suffices to prove a version of the KRW conjecture in which KW_g is replaced with a specific communication problem, namely, the *multiplexor relation* MUX of [EIRS01]. Specifically, he defined a composition of the form $KW_f \diamond MUX$, and showed that if a variant of the KRW conjecture for $KW_f \diamond MUX$ holds for every non-constant outer function f, then $\mathbf{P} \not\subseteq \mathbf{NC}^1$.

Motivation. Following the above discussion, our goal is to "replace" the relations U_n and $KW_{\bigoplus}$ in the known results with MUX. Unfortunately, this seems to be very difficult — in particular, the relation MUX seems to be significantly more complicated than U_n and $KW_{\bigoplus}$.

In order to make progress, we propose that a good intermediate goal would be to try to prove the KRW conjecture for the composition $KW_f \diamond KW_g$ *for inner functions g that are as complex*

and expressive as possible. Ideally, by extending the range of inner functions g that we can handle, we will develop stronger techniques, which would eventually allow us to prove the conjecture for $KW_f \diamond MUX$.

An additional motivation for proving the KRW conjecture for harder inner functions is that it may allow us to improve the state of the art lower bounds on depth complexity. The best known lower bound of $(3-o(1)) \cdot \log n$ [And87, PZ93, IN93, Hås98] was achieved by implicitly proving the KRW conjecture for $KW_f \diamond KW_{\oplus}$, and it may be improved by proving the KRW conjecture for new inner functions.

The question is, which inner functions g would be good candidates for such a program? Ideally, a good candidate for g would be such that the KW relation KW_g is more interesting than U_n and $KW_{\oplus}$, but less complicated than MUX. Unfortunately, there are not too many examples for such relations: in fact, the relations U_n, $KW_{\oplus}$, and MUX are more or less the only relations that are well-understood. Thus, we have a shortage of good candidates g for this program.

As a way out of this shortage, we propose to consider *monotone depth complexity* in the study of inner functions. Given a *monotone* function f, the *monotone depth complexity of* f, denoted $\mathsf{mD}(f)$, is the minimal depth of a *monotone* circuit that computes f. The *monotone KW relation* of a monotone function f, denoted mKW_f, is defined similarly to KW_f, but this time the goal of Alice and Bob is to find a coordinate i such that $x_i > y_i$ (rather than $x_i \neq y_i$). Karchmer and Wigderson [KW90] observed that $\mathsf{mD}(f) = \mathsf{CC}(mKW_f)$.

Fortunately, there are many monotone KW relations that are well-understood, and which are significantly more interesting than U_n and $KW_{\oplus}$. We would like to study compositions in which these monotone KW relations serve as the "inner part", in the hope that such study would lead us to discover new techniques.

1.1 Our results

1.1.1 The monotone composition theorem

Motivated by considerations discussed above, our first result concerns the *monotone KRW conjecture*. This conjecture says that for every two non-constant monotone functions f, g it holds that

$$\mathsf{CC}(mKW_f \diamond mKW_g) \approx \mathsf{CC}(mKW_f) + \mathsf{CC}(mKW_g)$$

(where $mKW_f \diamond mKW_g \stackrel{\text{def}}{=} mKW_{f \diamond g}$). This conjecture was studied in the original paper of Karchmer et. al. [KRW95], who proved it for the case where both f and g are the set-cover function. However, it received far less attention than the non-monotone conjecture, perhaps because the monotone analogue of $\mathbf{P} \not\subseteq \mathbf{NC}^1$ has been known to hold for a long time, and monotone depth complexity is considered to be very well understood in general.

Nevertheless, we believe that this conjecture is interesting for several reasons: First, it is a very natural question in its own right. Second, if we cannot prove the KRW conjecture in the monotone setting, what hope do we have to prove it in the non-monotone setting, which is far less understood? Finally, proving the monotone KRW conjecture might prove useful for tackling other important questions on monotone depth complexity, such as proving lower bounds on slice functions (which in particular would imply non-monotone lower bounds).

Our first main result is a proof of the monotone KRW conjecture for every non-constant monotone function f, and for a wide range of monotone functions g. Specifically, our result holds for every function g whose monotone depth complexity can be lower bounded using a "lifting theorem": A *lifted search problem* $S \diamond \mathrm{gd}$ is obtained by composing a search problem S with an appropriate

"gadget" function gd. A *lifting theorem* is is a theorem that translates a lower bound for S in a weak model of computation to a lower bound for $S \diamond \mathrm{gd}$ in a strong model.

Here, the relevant weak model of computation is query complexity. Informally, the *query complexity* of a search problem S, denoted $\mathsf{Q}(S)$, is the number of queries one should make to the input in order to find a solution (see Section 2.4 for a formal definition). Fix a gadget $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ of input length t. A few lifting theorems [RM99, CKLM17, WYY17, CFK$^+$19] establish that if the gadget gd satisfies certain conditions, then $\mathsf{CC}(S \diamond \mathrm{gd}) = \Omega(\mathsf{Q}(S) \cdot t)$. In this work, we use a lifting theorem of Chattopadhyay et. al. [CFK$^+$19], which hold for every gadget gd that has sufficiently low discrepancy and sufficiently large input length (see Theorem 2.26 for the formal statement).

Our result says that the monotone KRW conjecture holds whenever the lower bound on mKW_g can be proved using the theorem of [CFK$^+$19]. More specifically, there should exist a reduction to mKW_g from a lifted search problem $S \diamond \mathrm{gd}$ that satisfies the conditions of [CFK$^+$19]. This is a much wider family of inner functions than what previous composition theorems could handle (i.e., universal relation and parity), though we are now working in the monotone rather than the non-monotone setting. Informally, the composition theorem can be stated as follows (see Theorem 3.1 for the formal statement):

Theorem 1.2 (monotone composition theorem, informal)**.** *Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant monotone functions. If there is a lifted search problem $S \diamond \mathrm{gd}$ that reduces to mKW_g and satisfies the conditions of the theorem of [CFK$^+$19], then*

$$\mathsf{CC}(mKW_f \diamond mKW_g) \geq \mathsf{CC}(mKW_f) + \Omega(\mathsf{Q}(S) \cdot t).$$

In particular, if $\mathsf{CC}(mKW_g) = \tilde{O}\left(\mathsf{Q}(S) \cdot t\right)$, then

$$\mathsf{CC}(mKW_f \diamond mKW_g) \geq \mathsf{CC}(mKW_f) + \tilde{\Omega}(\mathsf{CC}(mKW_g)). \tag{3}$$

We would like to note that the the theorem is applicable to many interesting cases, including the classic s-t-connectivity function [KW90, GS91] clique function [GH92, RW92], and generation function [RM99] (see Appendix A for details). Moreover, we would like to mention that the bound of Equation (3) is good enough for the purposes of the KRW conjecture.

We would also like to stress that while the statement of our monotone composition theorem refers to the lifting theorem of [CFK$^+$19], we believe it can be adapted to work with similar lifting theorems such as the ones of [RM99, CKLM17, WYY17] (in other words, the specific choice of the lifting theorem is not particularly crucial). Finally, it should be mentioned that the formal statement of the monotone composition theorem actually refers to formula complexity rather than depth complexity.

In order to prove Theorem 1.2, we introduce a generalization of the lifting theorem of [CFK$^+$19], which may be of independent interest. Roughly, our generalization shows a lower bound for the lifted problem $S \diamond \mathrm{gd}$ even when restricted to a subset of its inputs, as long as this subset satisfies a certain condition. See Section 1.2.1 for further discussion.

1.1.2 The semi-monotone composition theorem

Recall our end goal is to gain insight into the *non-monotone* setting. To this end, we define a new form of composition, called *semi-monotone composition*, which composes a *non-monotone* outer KW relation with a *monotone* inner KW relation. The purpose of this new composition is to enjoy the best of both worlds: On the one hand, this notion allows us to use candidates for the inner function g that come from the monotone setting. On the other hand, we believe that this notion is

much closer to the non-monotone setting. Thus, by studying semi-monotone composition we can tackle issues that come up in the non-monotone setting but not in the monotone setting.

In order to gain intuition for the definition of this composition, consider the obvious protocol for the non-monotone composition $KW_f \diamond KW_g$. Recall that the inputs to this protocol are matrices $X, Y \in \{0,1\}^{m \times n}$, and that we denote by a, b the column vectors that are obtained by applying g to the rows of those matrices. Observe that there are two key properties of $KW_f \diamond KW_g$ that allow the obvious protocol to work:

- The players can find a row $i \in [m]$ such that $a_i \neq b_i$ by solving KW_f on a, b.
- For every $i \in [m]$ such that $a_i \neq b_i$, the players can find a solution for $KW_f \diamond KW_g$ by solving mKW_g on the rows X_i, Y_i.

Note that, while the obvious protocol always finds a solution in a row i where $a_i \neq b_i$, the rows where $a_i = b_i$ might contain solutions as well.

We define the semi-monotone composition of KW_f and mKW_g as a communication problem that is identical to $KW_f \diamond KW_g$, except that in the second property above, the non-monotone relation KW_g is replaced with the monotone relation mKW_g. Formally, we define semi-monotone composition as follows.

Definition 1.3 (Semi-monotone composition)**.** Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant (possibly non-monotone) function, and let $g : \{0,1\}^n \to \{0,1\}$ be a non-constant monotone function. The *semi-monotone composition* $KW_f \diamond mKW_g$ is the following communication problem. Alice and Bob get as inputs $m \times n$ binary matrices X and Y respectively. Let $a, b \in \{0,1\}^m$ denote the column vectors that are obtained by applying g to each row of X and Y respectively. Then, $f(a) = 1$ and $f(b) = 0$, and the goal of the players is to find an entry (i, j) that satisfies one of the following three options:

- $a_i > b_i$ and $X_{i,j} > Y_{i,j}$.
- $a_i < b_i$ and $X_{i,j} < Y_{i,j}$.
- $a_i = b_i$ and $X_{i,j} \neq Y_{i,j}$.

Note that this communication problem has the desired structure: Indeed, it is not hard to see that when $a_i \neq b_i$, finding a solution in the the i-th row is equivalent to solving mKW_g on X_i, Y_i. It is also not hard to show that $\mathsf{CC}(KW_f \diamond mKW_g) \leq \mathsf{CC}(KW_f) + \mathsf{CC}(mKW_g)$ bits, by using an appropriate variant of the obvious protocol of $KW_f \diamond KW_g$. Therefore, a natural "semi-monotone variant" of the KRW conjecture would be the following.

Conjecture 1.4 (Semi-monotone KRW conjecture)**.** *For every non-constant function $f : \{0,1\}^m \to \{0,1\}$ and non-constant monotone function $g : \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{CC}(KW_f \diamond mKW_g) \gtrapprox \mathsf{CC}(KW_f) + \mathsf{CC}(mKW_g).$$

Our result. Ideally, we would have liked to prove Conjecture 1.4 for every outer function f and for a wide range of inner functions g. Unfortunately, we are only able to prove it for the case where the outer relation mKW_f is replaced with the (non-monotone) universal relation, i.e., the composition $U_m \diamond mKW_g$. This composition is defined similarly to Definition 1.3, with the following difference: instead of promising that $f(a) = 1$ and $f(b) = 0$, we only promise that $a \neq b$. The natural conjecture in this case would be that

$$\mathsf{CC}(U_m \diamond mKW_g) \gtrapprox \mathsf{CC}(U_m) + \mathsf{CC}(mKW_g) \geq m + \mathsf{CC}(mKW_g), \tag{4}$$

where the second inequality holds since $\mathsf{CC}(U_m) = m + \Theta(1)$ (see [KRW95, TZ97]). Our semi-monotone composition theorem proves such a result for every monotone inner function g for which a lower bound on $\mathsf{CC}(mKW_g)$ can be proved using a lifting theorem of [dRMN+19].

Before describing our result, we briefly describe the lifting theorem of [dRMN+19]. Given an unsatisfiable CNF formula ϕ, its *associated search problem* S_ϕ is the following task: given an assignment z to ϕ, find a clause of ϕ that is violated by z. The *Nullstellensatz degree* of ϕ, denoted $NS_{\mathbb{F}}(\phi)$, is a complexity measure that reflects how hard it is prove that ϕ is unsatisfiable in the Nullstellensatz proof system over a field $\mathbb{F}$ (see Section 2.6 for a formal definition). Fix a gadget $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ of input length t. The lifting theorem of [dRMN+19] says that $\mathsf{CC}(S_\phi \diamond \mathrm{gd}) \geq \Omega(NS_{\mathbb{F}_2}(\phi) \cdot t)$ provided that the gadget gd has sufficiently large rank.

Our result says that Equation (4) holds whenever there is a reduction from such a lifted problem $S_\phi \diamond \mathrm{gd}$ to mKW_g. We require the gadget gd to be the equality function eq, and require the reduction to be *injective* (see Definition 2.8 for the definition of injective reduction). Informally, our semi-monotone composition theorem can be stated as follows (see Theorem 4.1 for the formal statement):

Theorem 1.5 (semi-monotone composition theorem, informal)**.** *Let $g : \{0,1\}^n$ be a non-constant monotone function, and let* eq *be the equality function on strings of length t. Suppose there exists a lifted search problem $S_\phi \diamond \mathrm{eq}$ that reduces to mKW_g via an injective reduction and satisfies the conditions of the theorem of [dRMN+19]. Then*

$$\mathsf{CC}(U_m \diamond mKW_g) \geq m + \Omega(NS_{\mathbb{F}_2}(\phi) \cdot t).$$

In particular, if $\mathsf{CC}(mKW_g) = \tilde{O}(NS_{\mathbb{F}_2}(\phi) \cdot t)$, then

$$\mathsf{CC}(U_m \diamond mKW_g) \geq m + \tilde{\Omega}(\mathsf{CC}(mKW_g))$$

for some $\varepsilon > 0$.

As in the case of the monotone composition theorem, the semi-monotone theorem is applicable to many interesting cases, including the classic s-t-connectivity, clique, and generation functions mentioned above (see Appendix A for details), and the bound that it gives is good enough for the purposes of the KRW conjecture.

Comparison to monotone composition. Recall that our goal in defining semi-monotone composition is to captures issues that arise in the non-monotone setting but are not captured by the monotone setting. We claim that our definition succeeds in this task for at least one significant issue, to be discussed next.

Recall that the KRW conjecture says that the obvious protocol for $KW_f \diamond KW_g$ is essentially optimal. Intuitively, this should be the case since it seems that the best strategy for the players is to work on a row where $a_i \neq b_i$, and to do it, they must first find such a row. While it seems reasonable that the best strategy is to work on a row where $a_i \neq b_i$, it is not clear how to prove it: indeed, this is a central challenge in the proofs of known composition theorems (though not the only challenge).

On the other hand, Karchmer et. al. [KRW95] observed that in the monotone setting, the players can be forced to solve the problem on a row where $a_i > b_i$. This means that in the monotone setting, we can easily bypass a central challenge of the non-monotone case. An important feature of semi-monotone composition is that the observation of [KRW95] fails for this composition. Hence, we believe that the semi-monotone setting is much closer to the non-monotone KRW conjecture than the monotone setting.

1.2 Our techniques

1.2.1 The monotone composition theorem

We use the high level proof strategy that was introduced by [EIRS01], and further developed in [DM18, Mei17, KM18]. The main technical lemma is a structure theorem, formalizing that any correct protocol must first solve mKW_f, and then solve mKW_g. A bit more formally, we show that for any partial transcript π_1 of Π, if mKW_f has not yet been solved at π_1, then Π must send $\approx \mathsf{CC}(mKW_g)$ additional bits before it can find a solution for $mKW_f \diamond mKW_g$.

To accomplish this, at π_1, we partition the rows of X, Y into two types: (1) "revealed" rows where π_1 knows much information, and (2) "unrevealed" rows, where π_1 knows only a small amount of information. We then show that the revealed rows can be forced to be useless (that is, we can ensure that the players cannot find a coordinate (i, j) where i is a revealed row). It follows that in order for the protocol to finish after π_1, it has to solve mKW_g on one of the unrevealed rows.

The remaining step is therefore to show that in order to solve mKW_g on one of the unrevealed rows, the protocol must transmit $\approx \mathsf{CC}(mKW_g)$ additional bits. While this claim sounds intuitive, proving it is non-trivial since some (small amount of) information has been learned about each unrevealed row, and this revealed information can be highly dependent. Moreover, the protocol is allowed to choose on which unrevealed row it would solve mKW_g, and this could in principle make the task significantly easier. In previous works, those issues were dealt with in a way that was tailored to the particular choice of g. Specifically, one would take a known lower bound proof for KW_g, and show that it still goes through even after accounting for the aforementioned complications.

In our case, we do not know the particular choice of g, but we do know that the lower bound for mKW_g is proved using the lifting theorem of [CFK$^+$19]. Hence, our goal is show that this lower bound proof still goes through. To this end, we prove a generalization of that lifting theorem which may be of independent interest (see Theorem 5.1). Informally, our generalization shows that $S \diamond \mathrm{gd}$ remains hard even if we restrict it to a subset $\mathcal{X} \times \mathcal{Y}$ of its inputs, as long as the coordinates remain *unpredictable*. Since this is the case for the unrevealed rows, we get the lower bound that we desire.

The notion of unpredictability required by our lifting theorem is based on *average degree* as defined by [EIRS01, RM99]: given a set of strings $\mathcal{W} \in \Lambda^\ell$ and a subset of coordinates $I \subseteq [\ell]$, the *average degree* $\mathrm{AvgDeg}_I(\mathcal{W})$ is the average number of ways to complete a string in $\mathcal{W}|_{[\ell]-I}$ to a string in $\mathcal{W}$. Informally, our generalized lifting theorem says the following (see Theorem 5.1 for the formal statement):

Theorem 1.6 (informal)**.** *Let $S \diamond \mathrm{gd}$ be a lifted search problem that satisfies the conditions of of [CFK$^+$19]. Let $\mathcal{X} \times \mathcal{Y}$ be a subset of the inputs of $S \diamond \mathrm{gd}$ such that $\mathrm{AvgDeg}_I(\mathcal{X})$ and $\mathrm{AvgDeg}_I(\mathcal{Y})$ are sufficiently large for every set of coordinates I. Then, the communication complexity of solving $S \diamond \mathrm{gd}$ on the inputs in $\mathcal{X} \times \mathcal{Y}$ is at least $\Omega\left(\mathsf{Q}(S) \cdot t\right)$.*

Our proof of the generalized lifting theorem mostly follows the proof of [CFK$^+$19], except for one significant issue: In both proofs, the communication complexity is bounded using a potential argument. In the original proof of [CFK$^+$19], the potential function is the min-entropy deficiency with respect to the uniform distribution *over all the inputs*. In our proof, on the other hand, the potential function measures the deficiency with respect to the uniform distribution *over the restricted set of inputs*. The latter distribution is less structured, and hence the potential argument requires a more refined analysis.

1.2.2 The semi-monotone composition theorem

We prove the lower bound on $U_m \diamond mKW_g$ using the Razborov rank method (see Section 2.5). Basically, in order to use this method to prove a lower bound on a communication problem $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$, one needs to construct an matrix A of order $|\mathcal{X}| \times |\mathcal{Y}|$ such that A has a high rank, but its restriction to every S-monochromatic rectangle has a low rank. Roughly, the lifting theorem of [dRMN+19] gives such a matrix A for mKW_g, and we use this matrix to construct a corresponding matrix M for $U_m \diamond mKW_g$.

The matrix M for $U_m \diamond mKW_g$ is constructed as follows. The rows and columns of M are indexed by matrices X and Y respectively. We view the matrix M as a block matrix that consists of $2^m \cdot 2^m$ blocks — a block for each value of a and b. For every a, b such that $a = b$, the corresponding block is the all-zeroes matrix. For every other choice of a, b, the corresponding block is formed by taking the Kronecker product, for every $i \in [m]$, of either A (if $a_i \neq b_i$) or the identity matrix I (if $a_i = b_i$).

The matrix M is constructed in that way in order to guarantee that all its restrictions to monochromatic rectangles have low rank. On the one hand, having the matrix A in rows i where $a_i \neq b_i$ guarantees that monochromatic rectangles that solve mKW_g on such rows X_i, Y_i have low rank. On the other hand, having the identity matrix I in rows i where $a_i = b_i$ guarantees that monochromatic rectangles that find different entries $X_{i,j} \neq Y_{i,j}$ are all-zeroes rectangles.

An important part of the proof is the observation that the when the theorem of [dRMN+19] is applied with the equality gadget over $\mathbb{F}_2$ (as we do), it gives a matrix A that satisfies that $A^2 = I$. This property creates a connection between A and I that allows us to analyze the rank of M and its sub-matrices using Gaussian elimination.

Organization of this paper. We cover the necessary preliminaries in Section 2. Then, we prove the monotone composition theorem in Section 3, and the semi-monotone composition theorem Section 4. We prove our generalization of the lifting theorem of [CFK+19] in Section 5. Finally, in Section 6 we discuss open problems for future research, and in Appendix A, we show how to apply our theorems to the classic functions s-t-connectivity, clique, and generation.

2 Preliminaries

Throughout the paper, we use bold letters to denote random variables. For any $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. We denote by $\mathbb{F}_2$ the finite field of size 2. We say that a CNF formula ϕ is a *CNF contradiction* if and only if it is unsatisfiable.

Given two strings $x, y \in \{0,1\}^n$, we write $x \geq y$ if $x_i \geq y_i$ for every $i \in [n]$. We say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if for every $x, y \in \{0,1\}^n$ such that $x \geq y$ it holds that $f(x) \geq f(y)$.

Given an alphabet Λ and a set $I \subseteq [n]$, we denote by Λ^I the set of strings of length $|I|$ whose coordinates are indexed by I. Given a string $w \in \Lambda^n$ and a set $I \subseteq [n]$, we denote by $w_I \in \Lambda^I$ the projection of w to the coordinates in I (in particular, $w_\emptyset$ is defined to be the empty string). Given a set of strings $\mathcal{W} \subseteq \Lambda^n$ and a set $I \subseteq [n]$, we denote by $\mathcal{W}_I$ the set of projections of strings in $\mathcal{W}$ to I.

We denote by $\Lambda^{m \times n}$ the set of $m \times n$ matrices with entries in Λ, and for sets $I \subseteq [m]$ and $J \subseteq [n]$, we denote by $\Lambda^{I \times J}$ the set of $|I| \times |J|$ matrices whose entries are indexed by $I \times J$. Given a matrix $X \in \Lambda^{m \times n}$ and a rectangle $R \stackrel{\text{def}}{=} I \times J \subseteq [m] \times [n]$, we denote by $X|_R$ the projection of X to R. We denote by $X_i \in \Lambda^n$ the i-th row of X. Given a matrix $A \in \mathbb{F}^{m \times n}$ over a finite field $\mathbb{F}$, we denote its rank by $\text{rank}_{\mathbb{F}}(A)$.

Search problems. Given a finite set of inputs $\mathcal{I}$ and a finite set of outputs $\mathcal{O}$, a *search problem* S is a relation between $\mathcal{I}$ and $\mathcal{O}$. Given $z \in \mathcal{I}$, we denote by $S(z)$ the set of outputs $o \in \mathcal{O}$ such that $(z, o) \in S$. Intuitively, a search problem S represents the following task: given an input $z \in \mathcal{I}$, find a solution $o \in S(z)$. Without loss of generality, we may assume that $S(z)$ is always non-empty, since otherwise we can set $S(z) = \{\perp\}$ where $\perp$ is some special failure symbol that does not belong to $\mathcal{O}$.

2.1 Communication complexity

We assume familiarity with the basic definitions of communication complexity (see, e.g., [KN97]). In what follows, we highlight some important basic definitions and facts that we will use, and define some less standard notions. Recall that a *(deterministic) protocol* Π is defined as a binary tree. We identify the vertices of the protocol with the transcripts that they represent. Given sets $\mathcal{X}$ and $\mathcal{Y}$, we say that the protocol has *domain* $\mathcal{X} \times \mathcal{Y}$ if the inputs of Alice and Bob are taken from the sets $\mathcal{X}$ and $\mathcal{Y}$ respectively. We say that the *range* of the protocol is a set $\mathcal{O}$ if the protocol outputs elements in $\mathcal{O}$.

Definition 2.1. Given a transcript π, we say that it is a *full transcript* if it corresponds to a leaf of the protocol tree, and otherwise we say that it is a *partial transcript*. Given a pair of inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we define the transcript of (x, y), denoted $\Pi(x, y)$, as the full transcript of the protocol when Alice and Bob get the inputs x and y respectively.

Definition 2.2. We say that two protocols Π, Π' over the same domain and range are *equivalent* if they have the same output on every pair of inputs.

Definition 2.3. A *communication problem* $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$ is the search problem in which Alice and Bob get inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively, and would like find a solution $o \in S(x, y)$. A protocol *solves* S if on every pair of inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ it outputs some $o \in S(x, y)$.

Definition 2.4. The *communication complexity* of a protocol Π, denoted $\mathsf{CC}(\Pi)$, is the the depth of the protocol tree. For a search problem S, the *(deterministic) communication complexity* of S, denoted $\mathsf{CC}(S)$, is the minimal communication complexity of a protocol that solves S.

Definition 2.5. We define the *size* of a protocol Π to be its number of leaves. We define the *protocol size* of a search problem S, denoted $\mathsf{L}(S)$, as the size of the smallest protocol that solves it (this is also known as the *protocol partition number* of S).

It is not hard to see that for every protocol Π it holds that $\mathsf{CC}(\Pi) \geq \log \mathsf{L}(\Pi)$ — informally, every "shallow" protocol is "small". The following folklore fact establishes a connection in the other direction: namely, every "small" protocol can be transformed into a "shallow" one. This transformation is sometimes called *protocol balancing*.

Fact 2.6 (protocol balancing, see, [KN97, Lemma 2.8])**.** *For every protocol* Π *there is an equivalent protocol* Π' *such that* $\mathsf{CC}(\Pi') \leq 2 \log \mathsf{L}(\Pi)$*. In particular, for every communication problem* S *it holds that*

$$\log \mathsf{L}(S) \leq \mathsf{CC}(S) \leq 2 \log \mathsf{L}(S)$$

and hence $\mathsf{CC}(S) = \Theta(\log \mathsf{L}(S))$.

Notation 2.7. Let Π be a protocol with domain $\mathcal{X} \times \mathcal{Y}$ and let π be a transcript of Π. Recall that the set of inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ on which the protocol reaches the vertex π is a combinatorial rectangle. We denote this rectangle by $\mathcal{X}_\pi \times \mathcal{Y}_\pi$. We say that π is a *full transcript* if it corresponds to a leaf of the protocol tree, and otherwise we say that it is a *partial transcript*.

Finally, we use the following definition, which generalizes the notion of a rectangular reduction of [BNS92] to search problems.

Definition 2.8. Let $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$ and $S' \subseteq \mathcal{X}' \times \mathcal{Y}' \times \mathcal{O}'$ be communication problems. A *reduction* from S to S' consists of functions $R_A : \mathcal{X} \to \mathcal{X}'$, $R_B : \mathcal{Y} \to \mathcal{Y}'$, and $R_{\text{out}} : \mathcal{O}' \to \mathcal{O}$ that satisfy the following condition: for every $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $o' \in \mathcal{O}'$, if o' is a solution for S' on inputs $R_A(x)$ and $R_B(y)$, then $R_{\text{out}}(o')$ is a solution for S on (x, y).

We say that the reduction is *injective* if the functions R_A, R_B are injective (but the function R_{out} is not required to be injective).

Remark 2.9. An important aspect of Definition 2.8 is that the function R_{out} is required not to depend on the inputs x, y. This stands in contrast to other definitions of reductions for search problems (e.g. a Levin reduction), which do allow their analogue of R_{out} to depend on the inputs. We note that this requirement is used in the proof of the semi-monotone composition theorem (Theorem 4.1), but not in the proof of the monotone composition theorem (Theorem 3.1).

2.2 Subadditive measures on trees

We use the following notions of a subadditive measure and a separating set of a tree.

Definition 2.10. Given a binary tree $T = (V, E)$, we say that a function $\gamma : V \to \mathbb{N}$ is a *subadditive measure on* T if for every internal vertex v with children v_0 and v_1 it holds that $\gamma(v) \leq \gamma(v_0) + \gamma(v_1)$.

Definition 2.11. Given a binary tree $T = (V, E)$, we say that a set of vertices $M \subseteq V$ is a *separating set* of T if every path from the root of T to its leaves passes through M.

We use the following fact about subadditive measures.

Claim 2.12. *Let $T = (V, E)$ be a binary tree with root r, let γ be a subadditive measure on T, and let M be a separating set of T. Then, there exists a vertex $v \in M$ such that $\gamma(v) \geq \gamma(r)/|M|$.*

Proof sketch. Let T, r, γ, and M be as in the claim. By applying the definition of subadditive measure inductively, it is not hard to show that

$$\gamma(r) \leq \sum_{v \in M} \gamma(v).$$

The claim now follows by averaging. ■

2.3 Monotone formulas and Karchmer-Wigderson relations

In this section, we define monotone formulas and KW relations formally, and state the connections between them.

Definition 2.13. A *monotone formula* ϕ is a binary tree, whose leaves are labeled with input variables x_i, and whose internal vertices are labeled as AND ($\wedge$) or OR ($\vee$) gates. We note that a single input variable x_i can be associated with many leaves. The *size* of a monotone formula is the number of its *leaves* (which up to a factor of 2 is the same as the number of edges or vertices of the tree).

Definition 2.14. A monotone formula ϕ over n variables computes a monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ in the natural way. The *monotone formula complexity* of a monotone function $f : \{0,1\}^n \to \{0,1\}$, denoted $\mathsf{mL}(f)$, is the size of the smallest monotone formula that computes f. The *monotone depth complexity* of f, denoted $\mathsf{mD}(f)$, is the smallest depth of a formula that computes f.

Remark 2.15. Note that we define here the monotone depth complexity of a function as the depth of a monotone *formula* that computes f, whereas in the introduction we defined it as the depth of a monotone *circuit* that computes f. However, it is not hard to see that the two definitions are equivalent.

Next, we generalize the above definitions from functions to promise problems, which will be useful when we discuss Karchmer-Wigderson relations.

Definition 2.16. Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$. We say that a monotone formula ϕ *separates* $\mathcal{X}$ and $\mathcal{Y}$ if $\phi(x) = 1$ for every $x \in \mathcal{X}$ and $\phi(y) = 0$ for every $y \in \mathcal{Y}$.

It is not hard to prove that two sets $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ are separated by some monotone formula if and only if they satisfy the following property: for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ it holds that $x_i > y_i$ for some coordinate $i \in [n]$. We denote this property by $\mathcal{X} \succ \mathcal{Y}$.

Definition 2.17. Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ be sets such that $\mathcal{X} \succ \mathcal{Y}$. The *monotone formula complexity of the rectangle* $\mathcal{X} \times \mathcal{Y}$, denoted $\mathsf{mL}(\mathcal{X} \times \mathcal{Y})$, is the size of the smallest formula that separates $\mathcal{X}$ and $\mathcal{Y}$. The *monotone depth complexity of the rectangle* $\mathcal{X} \times \mathcal{Y}$, denoted $\mathsf{mD}(\mathcal{X} \times \mathcal{Y})$, is the smallest depth of a formula that separates $\mathcal{X}$ and $\mathcal{Y}$. If the rectangle $\mathcal{X} \times \mathcal{Y}$ is empty, we define $\mathsf{mL}(\mathcal{X} \times \mathcal{Y}) = \mathsf{mD}(\mathcal{X} \times \mathcal{Y}) = 0$.

Note that Definition 2.14 is indeed a special case of Definition 2.16 where $\mathcal{X} = f^{-1}(1)$ and $\mathcal{Y} = f^{-1}(0)$. We turn to defining monotone KW relations. We first define them for general rectangles, and then specialize the definition to functions.

Definition 2.18. Let $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ be two sets such that $\mathcal{X} \succ \mathcal{Y}$. The *monotone KW relation* $mKW_{\mathcal{X} \times \mathcal{Y}}$ is the communication problem in which the input of Alice is $x \in \mathcal{X}$, the input of Bob us $y \in \mathcal{Y}$, and they would like to find a coordinate $i \in [n]$ such that $x_i > y_i$. Note that such a coordinate always exists by the assumption that $\mathcal{X} \succ \mathcal{Y}$.

Definition 2.19. Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant monotone function. The *monotone KW relation of* f, denoted mKW_f, is defined by $mKW_f \stackrel{\text{def}}{=} mKW_{f^{-1}(1) \times f^{-1}(0)}$.

We are now ready to state the connection of monotone KW relations to monotone depth and formula complexity.

Theorem 2.20 ([KW90], see also [Raz90])**.** *For every two sets $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^n$ such that $\mathcal{X} \succ \mathcal{Y}$ it holds that $\mathsf{mD}(\mathcal{X} \times \mathcal{Y}) = \mathsf{CC}(mKW_{\mathcal{X} \times \mathcal{Y}})$ and $\mathsf{mL}(\mathcal{X} \times \mathcal{Y}) = \mathsf{L}(mKW_{\mathcal{X} \times \mathcal{Y}})$. In particular, for every non-constant $f : \{0,1\}^n \to \{0,1\}$, it holds that $\mathsf{mD}(f) = \mathsf{CC}(mKW_f)$ and $\mathsf{mL}(f) = \mathsf{L}(mKW_f)$.*

Due to Theorem 2.20, in the rest of the paper we use the notations $\mathsf{mL}(\mathcal{X} \times \mathcal{Y})$ and $\mathsf{L}(mKW_{\mathcal{X} \times \mathcal{Y}})$ interchangeably.

Given a protocol Π that solves $mKW_{\mathcal{X} \times \mathcal{Y}}$, we can view the complexity measure mL as a subadditive measure over the protocol tree. Specifically, this measure assigns to each vertex v of Π the value $\mathsf{mL}(v) \stackrel{\text{def}}{=} \mathsf{mL}(\mathcal{X}_v \times \mathcal{Y}_v)$, where $\mathcal{X}_v \times \mathcal{Y}_v$ is the rectangle that is associated with v.

To see that this is indeed a subadditive measure, let v be an internal vertex of Π, and let v_0 and v_1 be its children. Without loss of generality, assume that at the vertex v it is Alice's turn to speak. Then, $\mathcal{X}_v = \mathcal{X}_{v_0} \cup \mathcal{X}_{v_1}$ and $\mathcal{Y}_v = \mathcal{Y}_{v_0} = \mathcal{Y}_{v_1}$. It holds that

$$\begin{aligned}
\mathsf{mL}(v) &= \mathsf{mL}(\mathcal{X}_v \times \mathcal{Y}_v) \\
&\leq \mathsf{mL}(\mathcal{X}_{v_0} \times \mathcal{Y}_v) + \mathsf{mL}(\mathcal{X}_{v_1} \times \mathcal{Y}_v) \qquad (5) \\
&= \mathsf{mL}(\mathcal{X}_{v_0} \times \mathcal{Y}_{v_0}) + \mathsf{mL}(\mathcal{X}_{v_1} \times \mathcal{Y}_{v_1}) \qquad (\text{Since } \mathcal{Y}_v = \mathcal{Y}_{v_0} = \mathcal{Y}_{v_1}) \\
&= \mathsf{mL}(v_0) + \mathsf{mL}(v_1).
\end{aligned}$$

To see why Equation (5) holds, consider the following protocol for $mKW_{\mathcal{X}_v \times \mathcal{Y}_v}$: Alice starts by saying whether her input belongs to $\mathcal{X}_{v_0}$ or to $\mathcal{X}_{v_1}$. Then, the players proceed by invoking the optimal protocol for either $mKW_{\mathcal{X}_{v_0} \times \mathcal{Y}_v}$ or $mKW_{\mathcal{X}_{v_1} \times \mathcal{Y}_v}$ respectively. It is easy to see that the size of this protocol is at most $\mathsf{mL}(\mathcal{X}_{v_0} \times \mathcal{Y}) + \mathsf{mL}(\mathcal{X}_{v_1} \times \mathcal{Y})$. Hence, mL is a subadditive measure, as required.

2.4 Decision trees

Informally, a decision tree is an algorithm that solves a search problem $S \subseteq \{0,1\}^\ell \times \mathcal{O}$ by querying the individual bits of its input. The tree is computationally unbounded, and its complexity is measured by the number of bits it queried. Formally,

Definition 2.21. A *(deterministic) decision tree* T with domain $\{0,1\}^\ell$ and range $\mathcal{O}$ is a binary tree in which every internal node is labeled with a coordinate in $[\ell]$ (which represents a query), every edge is labeled by a bit (which represents the answer to the query), and every leaf is labeled by an output in $\mathcal{O}$. Such a tree computes a function from $\{0,1\}^\ell$ to $\mathcal{O}$ in the natural way, and with a slight abuse of notation, we identify this function with T. The *query complexity* of T is the depth of the tree.

Definition 2.22. We say that a decision tree T *solves a search problem* $S \subseteq \{0,1\}^\ell \times \mathcal{O}$ if for every $z \in \{0,1\}^\ell$ it holds that $T(z) \in S(z)$. The *(deterministic) query complexity of* S, denoted $\mathsf{Q}(S)$, is the minimal query complexity of a deterministic decision tree that solves S.

2.5 The Razborov rank measure

The Razborov rank measure [Raz90] is a complexity measure that can be used to prove lower bounds on communication complexity. In order to introduce this measure, we first establish some notation. Let $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{O}$ be a communication problem. For some $o \in \mathcal{O}$, we say that a rectangle $R \subseteq \mathcal{X} \times \mathcal{Y}$ is *o-monochromatic (for S)* if $o \in S(x,y)$ for every $(x,y) \in R$. We say that R is *S-monochromatic* if it is o-monochromatic for some $o \in \mathcal{O}$. Let $\mathcal{R}$ denote the set of S-monochromatic rectangles.

Now, let $\mathbb{F}$ be a field. Given a matrix $A \in \mathbb{F}^{\mathcal{X} \times \mathcal{Y}}$, the *Razborov $\mathbb{F}$-rank measure of S with respect to A* is

$$\mu_{\mathbb{F}}(S, A) \stackrel{\text{def}}{=} \frac{\text{rank}_{\mathbb{F}}(A)}{\max\limits_{R \in \mathcal{R}} \text{rank}_{\mathbb{F}}(A|_R)}.$$

The *Razborov $\mathbb{F}$-rank measure of S*, denoted $\mu_{\mathbb{F}}(S)$, is the maximum of $\mu_{\mathbb{F}}(S, A)$ over all matrices $A \in \mathbb{F}^{\mathcal{X} \times \mathcal{Y}}$. We have the following result.

Fact 2.23 ([Raz90])**.** *For every field $\mathbb{F}$, it holds that $\mathsf{L}(S) \geq \mu_{\mathbb{F}}(S)$, and hence $\mathsf{CC}(S) \geq \log \mu_{\mathbb{F}}(S)$.*

2.6 The Nullstellensatz proof system

The Nullstellensatz proof system is a method for certifying that a set of polynomials does not have a common root. Formally, let $\mathbb{F}$ be a field, and let $P = \left\{p_i : \mathbb{F}^\ell \to \mathbb{F}\right\}_{i \in [m]}$ be a set of polynomials. By Hilbert's Nullstellensatz, the polynomials $p_1, \ldots, p_m$ do not have a common root if and only if there exist polynomials $q_1, \ldots, q_m : \mathbb{F}^n \to \mathbb{F}$ such that the following equality holds syntactically:

$$p_1 \cdot q_1 + \ldots + p_m \cdot q_m = 1. \tag{6}$$

The *Nullstellensatz degree of P with respect to $q_1, \ldots, q_m$* is the minimal degree of the polynomial $p_i \cdot q_i$ over all $i \in [m]$. The *Nullstellensatz degree of P* is the minimal Nullstellensatz degree of P over all polynomials $q_1, \ldots, q_m$ that satisfy Equation (6).

The Nullstellensatz proof system can be used to certify that a CNF formula is unsatisfiable. Let ϕ be a CNF formula over variables $x_1, \ldots, x_\ell$. Given a clause C of ϕ, we define the *polynomial encoding* of C as the polynomial that is obtained by multiplying $1 - x_i$ for every positive literal x_i that appears in C, and multiplying by x_i for every negative literal $\neg x_i$ that appears in C. Let P_ϕ denote the set of polynomials that consists of the polynomial encodings of all the clauses of ϕ, and of the polynomials $x_1^2 - x_1, \ldots, x_\ell^2 - x_\ell$. Then, a result of Buss et. al. [BIK+97] says that ϕ is unsatisfiable if and only if the set P_ϕ does not have a common root. This leads to the following natural definition of the Nullstellensatz degree of a CNF contradiction.

Definition 2.24. Let ϕ be a CNF contradiction, and let $\mathbb{F}$ be a field. The *Nullstellensatz degree of ϕ over $\mathbb{F}$*, denoted $NS_{\mathbb{F}}(\phi)$, is the Nullstellensatz degree of the set P_ϕ (where the polynomials in P_ϕ are viewed as polynomials over the field $\mathbb{F}$).

2.7 Lifting theorems

Lifting theorems are theorems that relate the complexity of a search problem S in a weak model to the complexity of the composed search problem $S \diamond \mathrm{gd}$ in a strong model. Formally, given a search problem $S \subseteq \{0,1\}^\ell \times \mathcal{O}$ and a "gadget" function $\mathrm{gd} : \Lambda \times \Lambda \to \{0,1\}$, the *lifted search problem* $S \diamond \mathrm{gd} \subseteq \Lambda^\ell \times \Lambda^\ell \times \mathcal{O}$ is the communication problem defined by

$$S \diamond \mathrm{gd}\left((x_1, \ldots, x_\ell), (y_1, \ldots, y_\ell)\right) \stackrel{\mathrm{def}}{=} S\left(\mathrm{gd}(x_1, y_1), \ldots, \mathrm{gd}(x_\ell, y_\ell)\right).$$

Lifting theorems are theorems that lower bound the complexity of $S \diamond \mathrm{gd}$ in terms of the complexity of S. The first theorems of this kind were proven by Raz and McKenzie [RM99], Shi and Zhou [SZ09], and Sherstov [She11]. The recent years have seen a flurry of results on lifting theorems and their applications (see, e.g., [GP18, GLM+16, GPW15, GPW17, dRNV16, RPRC16, CKLM17, PR17, WYY17, HHL18, PR18, CFK+19]). In this work, we use a theorem of [CFK+19] for lifting query complexity (discussed in Section 2.7.1 below), and a theorem of [dRMN+19] for lifting Nullstellensatz degree (discussed in Section 2.7.2).

2.7.1 Lifting from query complexity

It is not hard to see that for every search problem S, it holds that $\mathsf{CC}(S \diamond \mathrm{gd}) \leq \mathsf{Q}(S) \cdot \mathsf{CC}(\mathrm{gd})$: this upper bound is obtained by the protocol that simulates an optimal decision tree for S on the string $\mathrm{gd}(x_1, y_1), \ldots, \mathrm{gd}(x_\ell, y_\ell)$, and answers the queries of the tree by invoking an optimal protocol for gd. The first lifting theorem, due to Raz and McKenzie [RM99], established that if the gadget gd is the index function over sufficiently large inputs, then this upper bound is essentially tight, that is,

$$\mathsf{CC}(S \diamond \mathrm{gd}) = \Omega\left(\mathsf{Q}(S) \cdot \mathsf{CC}(\mathrm{gd})\right).$$

In other words, the theorem "lifts" the query complexity of S to a lower bound on the communication complexity of $S \diamond \mathrm{gd}$. This theorem was recently generalized to other choices of the gadget gd by [CKLM17, WYY17, CFK+19]. In this paper, we use the latter work of Chattopadhyay et. al. [CFK+19], which proved a lifting theorem for every gadget gd that has sufficiently low discrepancy. Below, we recall the definition of discrepancy, and state the relevant theorem of [CFK+19].

Definition 2.25 (discrepancy)**.** Let Λ be a finite set, let $\mathrm{gd} : \Lambda \times \Lambda \to \{0,1\}$ be a function, and let $\boldsymbol{u}, \boldsymbol{v}$ be independent random variables that are uniformly distributed over Λ. Given a combinatorial rectangle $R \subseteq \Lambda \times \Lambda$, the *discrepancy of* gd *with respect to* R, denoted $\mathrm{disc}(\mathrm{gd}, R)$, is defined as follows:

$$\mathrm{disc}(\mathrm{gd}, R) \stackrel{\mathrm{def}}{=} \left|\Pr\left[\mathrm{gd}(\boldsymbol{u}, \boldsymbol{v}) = 0 \text{ and } (\boldsymbol{u}, \boldsymbol{v}) \in R\right] - \Pr\left[\mathrm{gd}(\boldsymbol{u}, \boldsymbol{v}) = 1 \text{ and } (\boldsymbol{u}, \boldsymbol{v}) \in R\right]\right|.$$

The *discrepancy of* gd, denoted $\mathrm{disc}(\mathrm{gd})$, is defined as the maximum of $\mathrm{disc}(\mathrm{gd}, R)$ over all combinatorial rectangles $R \subseteq \Lambda \times \Lambda$.

Theorem 2.26 ([CFK+19])**.** *For every $\eta > 0$ there exists $c \in \mathbb{N}$ such that the following holds: Let S be a search problem that takes inputs from $\{0,1\}^\ell$, and let $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ be an arbitrary function such that $\mathrm{disc}(\mathrm{gd}) \leq 2^{-\eta \cdot t}$ and such that $t \geq c \cdot \log \ell$. Then*

$$\mathsf{CC}(S \diamond \mathrm{gd}) = \Omega\left(\mathsf{Q}(S) \cdot t\right).$$

2.7.2 Lifting from Nullstellensatz degree

Let ϕ be a q-CNF contradiction. The *search problem S_ϕ that corresponds to ϕ* is the following problem: given an assignment for ϕ, find a clause that is violated by the assignment. A series of works [RPRC16, PR17, PR18] showed that for appropriate gadgets gd, the communication complexity of $S_\phi \diamond \mathrm{gd}$ can be lower bounded in terms of the Nullstellensatz degree of ϕ. In fact, those theorems give a lower bound on the Razborov rank measure of $S_\phi \diamond \mathrm{gd}$, which is a stronger result.

In a recent joint work with Marc Vinyals [dRMN+19], we generalized the latter theorems to work for every gadget gd that has a large rank when viewed as a matrix. Formally, we have the following result.

Theorem 2.27 ([dRMN+19])**.** *Let ϕ be an q-CNF contradiction over ℓ variables, let S_ϕ be its corresponding search problem, let $\mathbb{F}$ be a field, and let $\mathrm{gd} : \Lambda \times \Lambda \to \{0,1\}$ be a gadget such that $\mathrm{rank}_{\mathbb{F}}(\mathrm{gd}) \geq 4$. Then*

$$\log \mu_{\mathbb{F}}(S_\phi \diamond \mathrm{gd}) \geq NS_{\mathbb{F}}(\phi) \cdot \log\left(\frac{NS_{\mathbb{F}}(\phi) \cdot \mathrm{rank}_{\mathbb{F}}(\mathrm{gd})}{e \cdot \ell}\right) - \frac{6 \cdot \ell \cdot \log e}{\mathrm{rank}_{\mathbb{F}}(\mathrm{gd})} - \log q.$$

In particular, when gd is the equality function with input length $t \geq 2 \log \ell$, we obtain the following result.

Corollary 2.28. *Let ϕ be an q-CNF contradiction over ℓ variables, let S_ϕ be its corresponding search problem, let $\mathbb{F}$ be a field, and let $\mathrm{eq} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ be the equality function such that $t \geq 2 \log \ell$. Then*

$$\log \mu_{\mathbb{F}}(S_\phi \diamond \mathrm{eq}) \geq \Omega\left(NS_{\mathbb{F}}(\phi) \cdot t\right) - \log q.$$

2.8 Min-entropy

Given a random variable $\boldsymbol{v}$ that takes values from a finite set $\mathcal{V}$, the *min-entropy* of $\boldsymbol{v}$, denoted $H_\infty(\boldsymbol{v})$, is the largest number $k \in \mathbb{R}$ such that $\Pr[\boldsymbol{v} = v] \leq 2^{-k}$ holds for every $v \in \mathcal{V}$. In other words,

$$H_\infty(\boldsymbol{v}) \stackrel{\mathrm{def}}{=} \min_{v \in \mathcal{V}} \left\{\log \frac{1}{\Pr[\boldsymbol{v} = v]}\right\}.$$

Min-entropy has the following easy-to-prove properties.

Fact 2.29. $H_\infty(\boldsymbol{v}) \leq \log |\mathcal{V}|$.

Fact 2.30. *Let $\mathcal{E} \subseteq \mathcal{V}$ be an event. Then, $H_\infty(\boldsymbol{v}|\mathcal{E}) \geq H_\infty(\boldsymbol{v}) - \log \frac{1}{\Pr[\mathcal{E}]}$.*

Fact 2.31. *Let $\boldsymbol{v}_1, \boldsymbol{v}_2$ be random variables taking values from finite sets $\mathcal{V}_1, \mathcal{V}_2$ respectively. Then, $H_\infty(\boldsymbol{v}_1) \geq H_\infty(\boldsymbol{v}_1, \boldsymbol{v}_2) - \log |\mathcal{V}_2|$.*

2.9 Prefix-free codes

A set of strings $C \subseteq \{0,1\}^*$ is called a *prefix-free code* if no string in C is a prefix of another string in C. Given a string $w \in \{0,1\}^*$, we denote its length by $|w|$. We use the following corollary of Kraft's inequality. A simple proof of this fact can be found in [CFK+19, Fact 2.8].

Fact 2.32 (Corollary of Kraft's inequality)**.** *Let $C \subseteq \{0,1\}^*$ be a finite prefix-free code, and let $\boldsymbol{w}$ be a random string taking values from C. Then, there exists a string $w \in C$ such that $\Pr[\boldsymbol{w} = w] \geq \frac{1}{2^{|w|}}$.*

2.10 Degrees of sets of strings

We use a framework of [EIRS01] for measuring the uncertainty of coordinates of strings. As a motivation, consider an unknown string $w \in \Lambda^N$, where the only thing we know about w is that it belongs to some set of strings $\mathcal{W} \subseteq \Lambda^N$. We would like to measure how much uncertainty we have about w. Perhaps the simplest way to measure it is the following notion of *density*.

Definition 2.33. The *density* of a set of strings $\mathcal{W} \subseteq \Lambda^N$ is

$$\text{density}(\mathcal{W}) \stackrel{\text{def}}{=} \frac{|\mathcal{W}|}{|\Lambda^N|}.$$

We would also like to measure the uncertainty we have about certain coordinates of w, *conditioned on the other coordinates*. The framework of [EIRS01] measures this uncertainty using the following notion of *degree*.

Definition 2.34. Let $\mathcal{W} \subseteq \Lambda^N$, and let $I \subseteq [N]$ be a set of coordinates. The *degree of a string $w' \in \Lambda^{[N]-I}$ in $\mathcal{W}$*, denoted $\deg(w', \mathcal{W})$, is the number of extensions of w' to strings in $\mathcal{W}$. The *average degree of I in $\mathcal{W}$*, denoted $\text{AvgDeg}_I(\mathcal{W})$, is the average degree over all string $w' \in \mathcal{W}|_{[N]-I}$. If $I = \{i\}$ is a singleton, we denote the average degree of I by $\text{AvgDeg}_i(\mathcal{W})$.

Intuitively, the degree of w' measures how much uncertainty we have about w_I if we know that $w_{[n]-I} = w'$. The average degree of I in $\mathcal{W}$ is a way to capture how much uncertainty we have about w_I conditioned on the other coordinates. It will be more convenient to work with the *relative* average degree, i.e., the ratio of the average degree to the largest possible degree, defined as follows.

Definition 2.35. Let $\mathcal{W}$ and I be as before. The *relative average degree of I in $\mathcal{W}$* is

$$\text{rAvgDeg}_I(\mathcal{W}) \stackrel{\text{def}}{=} \frac{\text{AvgDeg}_I(\mathcal{W})}{|\Lambda|^{|I|}}.$$

One useful property of average degree is that it behaves nicely when additional information is revealed about $\mathcal{W}$, which is captured by the following fact.

Fact 2.36 ([EIRS01])**.** *Let $\mathcal{W}' \subseteq \mathcal{W} \subseteq \Lambda^N$ be sets of strings and let $I \subseteq [N]$. Then,* $\text{rAvgDeg}_I(\mathcal{W}') \geq \frac{|\mathcal{W}'|}{|\mathcal{W}|} \cdot \text{rAvgDeg}_I(\mathcal{W})$.

Another useful property of the average degree is that when we remove a set of coordinates $I \subseteq [N]$ with a small average degree, the density of $\mathcal{W}$ *increases*. Intuitively, this means that when we drop coordinates about which a lot is known, the relative uncertainty increases.

Fact 2.37 ([RM99])**.** *Let $\mathcal{W} \subseteq \Lambda^N$ and let $I \subseteq [N]$. Then*

$$\text{density}(\mathcal{W}|_{[N]-I}) = \frac{1}{\text{rAvgDeg}_I(\mathcal{W})} \cdot \text{density}(\mathcal{W}).$$

Average degree also satisfies the following useful "chain rule".

Fact 2.38 (Implicit in [EIRS01])**.** *Let $\mathcal{W} \subseteq \Lambda^N$, and let $I, J \subseteq [N]$ be disjoint sets of coordinates. Then*

$$\text{rAvgDeg}_{I \cup J}(\mathcal{W}) = \text{rAvgDeg}_I(\mathcal{W}) \cdot \text{rAvgDeg}_J(\mathcal{W}_{[N]-I}).$$

Finally, it is interesting to note that average degree is a lower bound on another measure of uncertainty, namely, min-entropy:

Fact 2.39 ([KM18], following [EIRS01])**.** *Let $\mathcal{W} \subseteq \Lambda^N$, and let $\boldsymbol{w}$ be a random variable that is uniformly distributed over $\mathcal{W}$. Then, for every $I \subseteq [N]$ it holds that*

$$H_\infty(\boldsymbol{w}_I) \geq \log \text{AvgDeg}_I(\mathcal{W}) = |I| \cdot \log |\Lambda| - \log \frac{1}{\text{rAvgDeg}}.$$

2.11 Kronecker product

Let $\mathbb{F}$ be a field. In what follows, all the matrices are over the field $\mathbb{F}$.

Definition 2.40. Let A and B be $m \times n$ and $m' \times n'$ matrices respectively. The *Kronecker product* of A and B, denoted $A \otimes B$, is an $(m \cdot m') \times (n \cdot n')$ matrix whose rows and columns are indexed by pairs in $[m] \times [m']$ and $[n] \times [n']$ respectively, such that for every $i \in [m]$, $i' \in [m']$, $j \in [n]$, and $j' \in [n']$ it holds that

$$(A \otimes B)_{(i,i'),(j,j')} = A_{i,j} \cdot B_{i',j'}.$$

We use the following easy-to-prove facts about the Kronecker product.

Fact 2.41. *For every four matrices A, B, C, D it holds that*

$$(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D).$$

Fact 2.42. *For every three matrices A, B, C it holds that $A \otimes (B + C) = A \otimes B + A \otimes C$.*

Fact 2.43. *For every two matrices A, B it holds that $\text{rank}_{\mathbb{F}}(A \otimes B) = \text{rank}_{\mathbb{F}}(A) \cdot \text{rank}_{\mathbb{F}}(B)$.*

Fact 2.44. *Let A and B be block matrices that can be written as*

$$A = \begin{pmatrix} K_{1,1} & \dots & K_{1,q} \\ \vdots & \ddots & \vdots \\ K_{p,1} & \dots & K_{p,q} \end{pmatrix}, B = \begin{pmatrix} L_{1,1} & \dots & L_{1,q'} \\ \vdots & \ddots & \vdots \\ L_{p',1} & \dots & L_{p',q'} \end{pmatrix}.$$

where $K_{i,j}, L_{i',j'}$ denote the blocks. Then, the matrix $A \otimes B$ is a block matrix that can be written as

$$A \otimes B = \begin{pmatrix} K_{1,1} \otimes L_{1,1} & & \dots & & K_{1,q} \otimes L_{1,q'} \\ & \ddots & & & \\ \vdots & & K_{i,j} \otimes L_{i',j'} & & \vdots \\ & & & \ddots & \\ K_{p,1} \otimes L_{p',1} & & \dots & & K_{p,q} \otimes L_{p',q'} \end{pmatrix}.$$

3 The monotone composition theorem

In this section we prove our monotone composition theorem. Recall that this theorem can be stated informally as follows.

Theorem 1.2 (restated). *Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant monotone functions. If there is a lifted search problem $S \diamond \mathrm{gd}$ that reduces to mKW_g and satisfies the conditions of the theorem of [CFK+19], then*

$$\mathsf{CC}(mKW_f \diamond mKW_g) \geq \mathsf{CC}(mKW_f) + \Omega(\mathsf{Q}(S) \cdot t).$$

Formally, the theorem says the following.

Theorem 3.1. *For every $\eta > 0$ there exists $c \in \mathbb{N}$ such that the following holds: Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant monotone functions. Suppose that there exists a search problem $S \subseteq \{0,1\}^\ell \times \mathcal{O}$, and a function $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ of input length $t \geq c \cdot \log(m \cdot \ell)$ and discrepancy at most $2^{-\eta \cdot t}$, such that the lifted search problem $S \diamond \mathrm{gd}$ reduces to mKW_g. Then,*

$$\log \mathsf{L}(mKW_f \diamond mKW_g) \geq \log \mathsf{L}(mKW_f) + \Omega(\mathsf{Q}(S) \cdot t).$$

Let η, f, g, S, and gd be as in the theorem. We will choose the parameter c at the end of the proof. For convenience, we denote $S_{\mathrm{gd}} = S \diamond \mathrm{gd}$. We also denote $\Lambda \stackrel{\mathrm{def}}{=} \{0,1\}^t$, so the domain of gd is $\Lambda \times \Lambda$ and the domain of S_{gd} is $\Lambda^\ell \times \Lambda^\ell$.

Recall the communication problem $mKW_f \diamond mKW_g$: Alice and Bob get as inputs $m \times n$ binary matrices X and Y respectively. Let $a, b \in \{0,1\}^m$ denote the column vectors that are obtained by applying g to each row of X and Y respectively. Then, $f(a) = 1$ and $f(b) = 0$, and the players are required to find an entry (i,j) such that $X_{i,j} > Y_{i,j}$. The rest of this section is organized as follows.

- We start by proving that without loss of generality, it can be assumed that the players always output an entry (i,j) such that $a_i > b_i$. This is done in Section 3.1.1.
- We then show that it suffices to prove a lower bound on a simpler communication problem, denoted $mKW_f \circledast S_{\mathrm{gd}}$. This is done in Section 3.1.2.
- We prove the lower bound on $mKW_f \circledast S_{\mathrm{gd}}$ using a structure theorem, which says that, intuitively, the obvious protocol for $mKW_f \circledast S_{\mathrm{gd}}$ is the only efficient protocol for $mKW_f \circledast S_{\mathrm{gd}}$. In Section 3.2, we state this structure theorem, prove it based on two lemmas, and use it to derive the lower bound on $mKW_f \circledast S_{\mathrm{gd}}$.
- Finally, we prove the latter two lemmas in Sections 3.3 and 3.4 respectively.

3.1 Reductions

3.1.1 The observation of [KRW95]

We define the following variant of $mKW_f \diamond mKW_g$, denoted $mKW_f \circledast mKW_g$: The players get the same inputs as before, but now they are required to find an entry (i,j) that satisfies both $a_i > b_i$ and $X_{i,j} > Y_{i,j}$ (rather than just $X_{i,j} > Y_{i,j}$). Karchmer et. al. [KRW95] implicitly observed that $mKW_f \circledast mKW_g$ reduces to $mKW_f \diamond mKW_g$. This means that in order to prove Theorem 3.1, it suffices to prove a lower bound on $mKW_f \circledast mKW_g$. We now make this observation explicit.

Theorem 3.2. *The problem $mKW_f \circledast mKW_g$ reduces to $mKW_f \diamond mKW_g$.*

Proof. We describe functions R_A, R_B, R_{out} as in the definition of a reduction (Definition 2.8). Given a matrix $X \in \{0,1\}^{m \times n}$ that is an input for Alice in $mKW_f \circledast mKW_g$, the function R_A constructs an input $X' \in \{0,1\}^{m \times n}$ for Alice in $mKW_f \diamond mKW_g$ as follows: For every row index $i \in [m]$, if the i-th row X_i satisfies $g(X_i) = 1$, then we leave it intact — i.e., we set $X'_i = X_i$; otherwise, we set X'_i to be the all-zeroes string. Similarly, the function R_B takes an input matrix $Y \in \{0,1\}^{m \times n}$ and constructs a new matrix Y' by setting $Y'_i = Y_i$ if $g(Y_i) = 0$, and setting Y'_i to be the all-ones string otherwise. Finally, the function R_{out} is the identity function: it leaves the solution (i,j) for $mKW_f \diamond mKW_g$ intact.

To prove that the reduction works, we show that if (i,j) is a solution for $mKW_f \diamond mKW_g$ on (X', Y'), then it is also a solution for $mKW_f \circledast mKW_g$ on (X, Y). Let (i,j) be a solution for $mKW_f \diamond mKW_g$ on (X', Y'). This means that $X'_{i,j} > Y'_{i,j}$. In particular, X'_i is not the all-zeroes string, and Y'_i is not the all-ones string. By the definition of R_A, R_B, it follows that $X'_i = X_i$ and $Y'_i = Y_i$, and also that $g(X_i) = 1$ and $g(Y_i) = 0$. Therefore, (i,j) is an entry that satisfies both $a_i > b_i$ and $X_{i,j} > Y_{i,j}$. Hence, (i,j) is a solution for $mKW_f \circledast mKW_g$ on (X, Y), as required. ■

Remark 3.3. As discussed in the introduction, this reduction is a key technique that works in the monotone setting but not in the non-monotone and the semi-monotone settings. It is perhaps the main reason why it is easier to prove composition theorems in the monotone setting.

3.1.2 The problem $mKW_f \circledast S_{\text{gd}}$

In this section, we define a new communication problem $mKW_f \circledast S_{\text{gd}}$ and show that it reduces to $mKW_f \circledast mKW_g$. Informally, the problem $mKW_f \circledast S_{\text{gd}}$ is defined similarly to $mKW_f \circledast mKW_g$, except that the players need to solve S_{gd} on the i-th row rather than mKW_g. The reason that tis problem is useful is that it is more convenient to prove a lower bound on $mKW_f \circledast S_{\text{gd}}$ rather than directly on $mKW_f \circledast mKW_g$, since S_{gd} is a lifted search problem and thus has a structure that we can use. For the following definition, recall that the domain of S_{gd} is Λ^ℓ, and its range is $\mathcal{O}$.

Definition 3.4. The communication problem $mKW_f \circledast S_{\text{gd}}$ is defined as follows: Alice gets a matrix $X \in \Lambda^{m \times \ell}$ and a column vector $a \in f^{-1}(1)$, Bob gets a matrix $Y \in \Lambda^{m \times \ell}$ and a column vector $b \in f^{-1}(0)$, and their goal is to find a pair $(i,o) \in [m] \times \mathcal{O}$ such that $a_i > b_i$ and $o \in S_{\text{gd}}(X_i, Y_i)$ (i.e., o is a solution for S_{gd} on the i-th rows of X and Y).

Proposition 3.5. *$mKW_f \circledast S_{\text{gd}}$ reduces to $mKW_f \circledast mKW_g$.*

Proof. By assumption, S_{gd} reduces to mKW_g. Let $R_A : \Lambda^\ell \to f^{-1}(1)$, $R_B : \Lambda^\ell \to f^{-1}(0)$, and $R_{\text{out}} : [n] \to \mathcal{O}$ be the functions that define that reduction. We construct a reduction from $mKW_f \circledast S_{\text{gd}}$ to $mKW_f \circledast mKW_g$ by describing appropriate functions R'_A, R'_B, and R'_{out}.

Given an input $X \in \Lambda^{m \times \ell}$ and $a \in f^{-1}(1)$ for Alice in $mKW_f \circledast S_{\text{gd}}$, the function R'_A constructs an input $X' \in \{0,1\}^{m \times n}$ for Alice in $mKW_f \circledast mKW_g$ as follows: for every $i \in [m]$, we set X'_i to $R_A(X_i)$ if $a_i = 1$ and to the all-zeroes string otherwise. The function R'_B is defined similarly on an input $Y \in \Lambda^{m \times \ell}$ and $b \in f^{-1}(0)$, by setting Y'_i to be $R_B(Y_i)$ if $b_i = 0$ and to the all-ones string otherwise. Observe that if we apply g to the rows of X' and Y' we get the column vector a and b respectively. Finally, the function R'_{out} takes a solution (i,j) for $mKW_f \circledast mKW_g$ and translates it to an output (i,o) for $mKW_f \circledast S_{\text{gd}}$ by keeping i intact and setting $o = R_{\text{out}}(j)$.

To prove that the reduction works, we show that if (i,j) is a solution for $mKW_f \circledast mKW_g$ on (X', Y'), then (i,o) is also a solution for $mKW_f \circledast mKW_g$ on $((X,a),(Y,b))$. Let (i,j) be a solution for $mKW_f \circledast mKW_g$ on (X', Y'). This implies that j is a solution for mKW_g on (X'_i, Y'_i), and that $a_i > b_i$. Since $a_i > b_i$, it holds that so $a_i = 1$ and $b_i = 0$, and hence, $X'_i = R_A(X_i)$ and $Y'_i = R_B(Y_i)$. It follows that j is a solution for mKW_g on $(R_A(X_i), R_B(Y_i))$, and therefore

$o = R_{\text{out}}(j)$ is a solution for S_{gd} on (X_i, Y_i) by the definition of reduction. Thus, (i, o) is a solution for $mKW_f \circledast S_{\text{gd}}$, as required. ■

3.2 The structure theorem

We turn to proving the desired lower bound on $mKW_f \circledast S_{\text{gd}}$. Let $q \stackrel{\text{def}}{=} \mathsf{Q}(S)$ and $\Lambda = \{0,1\}^t$. We prove that

$$\log \mathsf{L}(mKW_f \circledast S_{\text{gd}}) \geq \log \mathsf{L}(mKW_f) + \Omega(q \cdot t). \tag{7}$$

Observe that there is an obvious protocol for solving $mKW_f \circledast S_{\text{gd}}$: The players first solve mKW_f on the column vectors a, b, thus obtaining a coordinate $i \in [m]$ such that $a_i > b_i$. Then, they solve S_{gd} on X_i, Y_i and obtain a solution o for S_{gd}. Finally, they output the pair (i, o). The communication complexity of this protocol is $\mathsf{CC}(KW_f) + \mathsf{CC}(S_{\text{gd}})$, and the logarithm of its size is

$$\begin{aligned}\log \mathsf{L}(mKW_f) + \log \mathsf{L}(S_{\text{gd}}) &\leq \log \mathsf{L}(mKW_f) + \mathsf{CC}(S_{\text{gd}}) \\ &\leq \log \mathsf{L}(mKW_f) + q \cdot t.\end{aligned}$$

Thus, our goal is to prove that the obvious protocol is optimal in terms of size, up to the constant factor of the $q \cdot t$ term.

We prove this bound by showing that every protocol must behave like the obvious protocol, in the sense that it must solve mKW_f on a, b before it starts solving S_{gd} on the rows X_i, Y_i. A bit more formally, our result says that for every protocol Π for $mKW_f \circledast S_{\text{gd}}$ the following holds: at any given point during the execution of Π in which the players have not solved mKW_f yet, the protocol must transmit at least another $\Omega(q \cdot t)$ bits in order to solve $mKW_f \circledast S_{\text{gd}}$. We refer to this result as *the structure theorem*. We state it formally below in Section 3.2.1, and show how to use it to prove Equation (7) in Section 3.2.2. Then, we prove it based on two lemmas in Section 3.2.3.

3.2.1 Statement of the structure theorem

In order to formalize the structure theorem, we need to define what we mean when we say "the players have not solved mKW_f yet" at a given point in time. To this end, we show that the protocol Π contains, in a sense, a protocol for mKW_f. Specifically, for a fixed matrix $W \in \Lambda^{m \times \ell}$, we define the following protocol Π_W for mKW_f: On inputs a, b for mKW_f, the protocol Π_W invokes the protocol Π on inputs (W, a) and (W, b), thus obtaining a pair (i, o) such that $a_i > b_i$ and o is a solution for S_{gd} on (W_i, W_i). Then, the protocol Π_W outputs i as its solution for mKW_f. It is not hard to see that Π_W is indeed a protocol for mKW_f.

Now, let π be a partial transcript of Π, and observe that π can also be viewed as a partial transcript of Π_W for every $W \in \Lambda^{m \times \ell}$. Informally, we say that the protocol Π has not solved mKW_f yet at the transcript π if for an average matrix $W \in \Lambda^{m \times \ell}$, the protocol Π_W has not solved mKW_f yet at π. For short, we say also that such a transcript is *alive.*

We proceed to formalize this intuition. Let π be a partial transcript of the protocol, and let $W \in \Lambda^{m \times \ell}$ be a matrix. We denote by $\mathcal{X}_\pi \times \mathcal{Y}_\pi$ the rectangle of inputs that is associated with π, and denote

$$\begin{aligned}\mathcal{A}_{\pi,W} &= \left\{a \in f^{-1}(1) : (W, a) \in \mathcal{X}_\pi\right\} \\ \mathcal{B}_{\pi,W} &= \left\{b \in f^{-1}(0) : (W, b) \in \mathcal{Y}_\pi\right\}.\end{aligned}$$

In other words, $\mathcal{A}_{\pi,W} \times \mathcal{B}_{\pi,W}$ is the rectangle of inputs that is associated with π *when viewed as a transcript of* Π_W. We measure how close Π_W is to solving mKW_f using the complexity measure

$$\mathsf{mL}(\mathcal{A}_{\pi,W} \times \mathcal{B}_{\pi,W}) = \mathsf{L}(mKW_{\mathcal{A}_{\pi,W} \times \mathcal{B}_{\pi,W}}).$$

We measure how close Π is to solving mKW_f by averaging this measure over all matrices W. Formally,

Definition 3.6. Fix a protocol Π for $mKW_f \circledast S_{\mathrm{gd}}$. For a transcript π of Π, we denote

$$\gamma(\pi) \stackrel{\mathrm{def}}{=} \frac{1}{|\Lambda^{m\times\ell}|} \cdot \sum_{W\in\Lambda^{m\times\ell}} \mathsf{mL}(\mathcal{A}_{\pi,W} \times \mathcal{B}_{\pi,W}).$$

We say that π is *alive* if $\gamma(\pi) \geq 4m^2$.

We are finally ready to state the structure theorem. Informally, it says that if the protocol Π is currently at a live transcript, then it must transmit at least another $\Omega(q \cdot t)$ bits in order to solve $mKW_f \circledast S_{\mathrm{gd}}$. Formally, we have the following result.

Theorem 3.7 (Structure theorem for $mKW_f \circledast S_{\mathrm{gd}}$)**.** *Fix a protocol Π for $mKW_f \circledast S_{\mathrm{gd}}$. For every live transcript π_1 of Π, there exists a suffix π_2 of length at least $\Omega(q \cdot t)$ such that the concatenation $\pi_1 \circ \pi_2$ is a transcript of Π.*

Remark 3.8. It may seem odd that in the definition of the protocol Π_W above, we give the matrix W to both players as an input, since there is no particular reason to give the players an identical matrix. Indeed, this requirement is made solely for convenience: We could have worked with two matrices — a matrix X for Alice and a matrix Y for Bob — but that would have been more cumbersome. The same goes for the definition of the measure γ: we could have averaged over all pairs of matrices $X, Y \in \Lambda^{m\times\ell}$ and look at the rectangle $\mathcal{A}_{\pi,X} \times \mathcal{B}_{\pi,Y}$, but using a single matrix W simplifies the presentation.

3.2.2 The lower bound on $mKW_f \circledast S_{\mathrm{gd}}$

We now prove the lower bound on $mKW_f \circledast S_{\mathrm{gd}}$ using the structure theorem. Fix a protocol Π that solves $mKW_f \circledast S_{\mathrm{gd}}$.

Communication complexity lower bound. As a warm-up, we start by proving a lower bound on the communication complexity of Π, namely,

$$\mathsf{CC}(\Pi) \geq \log \mathsf{L}(mKW_f) + \Omega(q \cdot t). \tag{8}$$

To this end, we use the following lemma, which establishes the existence of a relatively long live transcript.

Lemma 3.9. *Π has either a live transcript of length $\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor$, or a live transcript that is a leaf.*

Proof. The idea of the proof is the following: At the beginning of the protocol, the complexity of solving mKW_f is $\log \mathsf{L}(mKW_f)$. After the protocol transmits $\log \mathsf{L}(mKW_f) - 2\log m - 2$ bits, we expect the complexity to go down to $2\log m + 2$. This means that we expect the measure γ to become $2^{2\log m+2} = 4m^2$, which implies that the corresponding transcript is alive.

This intuition is formalized using the fact that the measure $\gamma(\pi)$ of Definition 3.6 is a subadditive measure on the protocol tree of Π. To see it, note that each of the individual terms $\mathsf{mL}(\mathcal{A}_{\pi,W} \times \mathcal{B}_{\pi,W})$ is a subadditive measure (see Section 2.3), and therefore their sum is also a subadditive measure. Next, let M be the set of vertices of Π that are

- either of depth exactly $\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor$;

- or a leaf of depth at most $\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor$.

It is not hard to see that M is a separating set of Π (see Definition 2.11), and that

$$|M| \leq 2^{\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor} \leq \mathsf{L}(mKW_f)/2^{2\log m + 2}.$$

Observe that γ assigns the root of Π a value of $\mathsf{L}(mKW_f)$. By Claim 2.12, there exists a vertex $\pi_1 \in M$ such that

$$\gamma(\pi_1) \geq \frac{\mathsf{L}(mKW_f)}{|M|} \geq \frac{\mathsf{L}(mKW_f)}{\mathsf{L}(mKW_f)/2^{2\log m+2}} \geq 4m^2.$$

This means that π_1 is a live transcript of Π, as required. ■

By combining Lemma 3.9 with the structure theorem, we immediately obtain the desired lower bound on the communication complexity of Π. Indeed, Lemma 3.9 says that Π has a live transcript π_1 that is either of length $\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor$ or a leaf. The structure theorem says that there is a suffix π_2 of length at least $\Omega(q \cdot t)$ such that the concatenation $\pi_1 \circ \pi_2$ is a transcript of Π. This implies in particular that π_1 is not a leaf (or otherwise $\pi_1 \circ \pi_2$ would not be a legal transcript of Π), and hence π_1 is a partial transcript of length exactly $\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor$. It follows that $\pi_1 \circ \pi_2$ is a full transcript of Π of length at least

$$\lfloor \log \mathsf{L}(mKW_f) - 2\log m - 2 \rfloor + \Omega(q \cdot t) \geq \log \mathsf{L}(mKW_f) + \Omega(q \cdot t),$$

where the inequality uses the fact that $t \gg \log(m)$. Hence, the communication complexity of Π is at least $\log \mathsf{L}(mKW_f) + \Omega(q \cdot t)$ as required.

Protocol size lower bound. While the above argument proves a lower bound on $\mathsf{CC}(mKW_f \circledast S_{\mathrm{gd}})$, our actual goal is to obtain a lower bound on the *protocol size* of $mKW_f \circledast S_{\mathrm{gd}}$, which is a stronger statement. That is, we would like to prove that

$$\log \mathsf{L}(\Pi) \geq \log \mathsf{L}(mKW_f) + \Omega(q \cdot t).$$

We stress that we cannot derive this lower bound from Equation (8) directly using protocol balancing (Fact 2.6), since that would lose a constant factor in the term $\log \mathsf{L}(mKW_f)$ and we cannot afford that loss. Nevertheless, we can afford to apply protocol balancing to the structure theorem, since we can afford to lose a constant factor in the $\Omega(q \cdot t)$ term. This leads to the following corollary, which will be used to prove the lower bound on $\mathsf{L}(\Pi)$.

Corollary 3.10. *For every live transcript π_1 of Π, there exist at least $2^{\Omega(q \cdot t)}$ suffixes π_2 of such that the concatenation $\pi_1 \circ \pi_2$ is a full transcript of Π.*

Proof. Let π_1 be a live transcript of Π, and let Π_2 be the sub-tree of Π that is rooted in π_1. We prove that $\mathsf{L}(\Pi_2) \geq 2^{\Omega(q \cdot t)}$, and this implies the desired claim. By Fact 2.6, there exists a protocol Π_2' that is equivalent to Π_2 and has communication complexity at most $2\log \mathsf{L}(\Pi_2)$. Let Π' be the protocol obtained from Π by replacing Π_2 with Π_2'.

Now, Π' is a protocol that solves $mKW_f \circledast S_{\mathrm{gd}}$, and π_1 is a live transcript of Π', so by Theorem 3.7 there exists a suffix π_2 of length at least $\Omega(q \cdot t)$ such that the concatenation $\pi_1 \circ \pi_2$ is a transcript of Π'. This means that π_2 is a transcript of Π_2' that has length at least $\Omega(q \cdot t)$, and therefore $\mathsf{CC}(\Pi_2') \geq \Omega(q \cdot t)$. It follows that

$$\begin{aligned} 2\log \mathsf{L}(\Pi_2) &\geq \mathsf{CC}(\Pi_2') \geq \Omega(q \cdot t) \\ \log \mathsf{L}(\Pi_2) &\geq \Omega(q \cdot t), \end{aligned}$$

as required. ■

We now prove the lower bound on $\mathsf{L}(\Pi)$. Ideally, we would have liked to prove that if Π did not have many leaves, then there would have to be at least one live transcript π_1 that does not have many leaves in its rooted sub-tree. Since the existence of such π_1 contradicts Corollary 3.10, this would prove that Π must have many leaves.

The latter "ideal claim" about Π is not true in general. However, [KM18] observed that Π can be transformed into an equivalent protocol Π' that does satisfy that claim, and is not much larger than Π. We can therefore use the foregoing argument to show that Π' has many leaves, and then argue that since Π' is not much larger than Π, the protocol Π must have many leaves as well. The transformation of Π is done by the following lemma of [KM18].

Lemma 3.11 ([KM18], following [Tal14])**.** *Let Π be a protocol, and let $s \in \mathbb{N}$ be a parameter such that $s \leq \mathsf{L}(\Pi)$. Then there exists an equivalent protocol Π' that satisfies the following: the protocol tree Π' has a separating set $\pi_1, \ldots, \pi_k$ where $k \leq \frac{36 \cdot \mathsf{L}(\Pi)}{s}$, such that for every $i \in [k]$, the subtree rooted at π_i has at most s leaves.*

By Corollary 3.10, there exists some some $L = 2^{\Omega(q \cdot t)}$ such that every live transcript π_1 has at least L suffixes. We prove that

$$\log \mathsf{L}(\Pi) \geq \log \mathsf{L}(mKW_f) + \log L - 2 \log m - 9, \tag{9}$$

and this would imply that $\log \mathsf{L}(mKW_f \circledast S_{\mathrm{gd}}) \geq \log \mathsf{L}(mKW_f) + \Omega(q \cdot t)$, as required. Suppose for the sake of contradiction that Equation (9) does not hold, that is, we assume that

$$\mathsf{L}(\Pi) < \frac{\mathsf{L}(mKW_f) \cdot L}{512 \cdot m^2}.$$

Let Π' be the protocol that is obtained by applying Lemma 3.11 to Π with $s = L/2$. Then, the protocol tree Π' has a separating set $\pi_1, \ldots, \pi_k$ such that

$$k \leq \frac{36 \cdot \mathsf{L}(\Pi)}{L/2} < \frac{\mathsf{L}(mKW_f)}{4 \cdot m^2},$$

and such that for every $i \in [k]$, the subtree rooted at π_i has at most $L/2$ leaves. Now, recall that the measure $\gamma(\pi)$ is a subadditive measure on the protocol tree of Π'. Moreover, recall that γ assigns to the root of Π' a value of $\mathsf{L}(mKW_f)$. Thus, by Claim 2.12, there exists a transcript π_i in the separating set such that

$$\gamma(\pi_i) \geq \frac{\mathsf{L}(mKW_f)}{k} > \frac{\mathsf{L}(mKW_f)}{\mathsf{L}(mKW_f)/4m^2} = 4m^2.$$

This means that π_i is alive, and therefore by Corollary 3.10, there are at least L leaves in the sub-tree of Π' that is rooted in π_i. However, this contradicts the fact that there are at most $L/2$ such leaves. We reached a contradiction, and hence Equation (9) holds.

3.2.3 Proof of structure theorem from lemmas

Let Π be a protocol that solves $mKW_f \circledast S_{\mathrm{gd}}$. Our goal is to prove that if the protocol reaches a live transcript π_1, then it still has to transmit at least $\Omega(q \cdot t)$ bits in order to solve $mKW_f \circledast S_{\mathrm{gd}}$. The intuition for the proof is the following: The goal of the players is to solve S_{gd} on some row i where $a_i > b_i$. By assumption, it is necessary to transmit $\Omega(q \cdot t)$ bits in order to solve S_{gd} *from scratch*. However, it could be the case that the transcript π_1 contains information that helps in solving S_{gd}

on some rows, which means that the players may need to transmit less than $\Omega(q \cdot t)$ bits in order to solve S_{gd} on those rows. The crucial point is that since at π_1 the players have not solved KW_f on a, b yet, they do not know on which row of X, Y they should be solving S_{gd}. Thus, the information that the players communicated in π_1 is likely to be wasted on irrelevant rows where $a_i \leq b_i$. Hence, we might as well assume that the players have not made progress toward solving S_{gd} in π_1, so they still have to transmit $\Omega(q \cdot t)$ bits in order to solve S_{gd} on some row.

This intuition is formalized as follows. Given a live transcript π_1, we partition the rows of the matrices X, Y into two types:

- "Revealed rows", about which the transcript π_1 reveals much information (i.e., more than two bits of information).
- "Unrevealed rows", about which the transcript π_1 reveals only a little information (i.e., at most two bits of information).

Intuitively, if the protocol chooses to solve S_{gd} on an *unrevealed* row, then it has to send $\Omega(q \cdot t)$ additional bits, since it barely made any progress on this row in in π_1. Thus, it suffices to show that we can prevent the protocol from solving S_{gd} on the revealed rows. This corresponds to our previous intuition that if the players communicate about some rows before solving mKW_f, then this communication will be wasted.

In order to force the protocol to solve S_{gd} on the unrevealed rows, we show that we can find a subset of the inputs that is consistent with π_1, and that satisfies that $a_i \leq b_i$ holds for every revealed row i. This means that on those inputs, the protocol is not allowed to output an entry in any revealed row. Basically, we can find such a subset of inputs since we assumed that at π_1 the players have not solved mKW_f yet, and hence at this point they do not know any row i for which $a_i > b_i$. Therefore, when the protocol is invoked on this subset of inputs, it must solve S_{gd} on an unrevealed row, and therefore must transmit about $\Omega(q \cdot t)$ additional bits, as required. More formally, the proof consists of two steps:

- In the first step of the proof, we show that there exists a collection $\mathcal{W}$ of matrices W and corresponding column vectors a, b that are consistent with π_1, such that $a_i \leq b_i$ holds for each revealed row i.
- In the second step of the proof, we prove that the complexity of solving $mKW_f \circledast S_{\mathrm{gd}}$ on such a collection of inputs is at least $\Omega(q \cdot t)$.

The following definition captures the collection of inputs that we would like to construct.

Definition 3.12. A *collection* consists of a set of matrices $\mathcal{W} \subseteq \Lambda^{m \times \ell}$, and of column vectors $a^W \in f^{-1}(1)$ and $b^W \in f^{-1}(0)$ for each matrix $W \in \mathcal{W}$. We say that a transcript π_1 of Π with a corresponding rectangle $\mathcal{X}_{\pi_1} \times \mathcal{Y}_{\pi_1}$ *supports the collection* if for every matrix $W \in \mathcal{W}$, it holds that $(W, a^W) \in \mathcal{X}_{\pi_1}$ and $(W, b^W) \in \mathcal{Y}_{\pi_1}$. We say that the collection is *hard* if there exists a set $R \subseteq [m]$ of "revealed rows" that satisfies the following:

- For every set $I \subseteq [m] - R$:

$$\mathrm{rAvgDeg}_{I \times [\ell]}\left(\mathcal{W}_{([m]-R) \times [\ell]}\right) \geq \frac{1}{4^{|I|}}$$

  (i.e., at most $2|I|$ bits of information were revealed on every set I of unrevealed rows).
- For every $W, W' \in \mathcal{W}$, it holds that $a^W|_R \leq b^{W'}|_R$.

The following two lemmas now capture the two main steps of the proof, and together imply the structure theorem. They are proved in Sections 3.3 and 3.4 respectively.

Lemma 3.13. *Every live transcript of* Π *supports a hard collection.*

Lemma 3.14. *If a transcript* π_1 *supports a hard collection, then there exists a suffix* π_2 *of length at least* $\Omega(q \cdot t)$ *such that* $\pi_1 \circ \pi_2$ *is a transcript of* Π*.*

The structure theorem follows immediately by combining the two lemmas.

3.3 Proof of Lemma 3.13

Fix a protocol Π that solves $mKW_f \circledast S_{\mathrm{gd}}$, and let π_1 be a live transcript of Π. Our goal is to construct a hard collection that is supported by π_1. To this end, we identify a set of matrices $\mathcal{W}$, a set of revealed rows R, and column vectors $a^W \in \mathcal{A}_{\pi_1,W}$ and $b^W \in \mathcal{B}_{\pi_1,W}$. We then show that $a^W|_R \leq b^{W'}|_R$ holds for every $W, W' \in \mathcal{W}$. Our proof is a straightforward adaptation of an argument of [KM18] to the monotone setting.

Our assumption that π_1 is alive means that $\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W})$ is sufficiently large for the average matrix W. In order to carry out our argument, we need to start from a stronger assumption, namely, that there is a significant number of matrices W for which $\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W})$ is sufficiently large. This can be proved by a standard averaging argument. Formally, in Section 3.3.1 below we prove the following result.

Proposition 3.15. *There exists a number* $p \in \mathbb{N}$ *and a set of matrices* $\mathcal{W}_0 \subseteq \Lambda^{m \times \ell}$ *such that* $\mathrm{density}(\mathcal{W}_0) \geq 2^{-p}$*, and such that for every* $W \in \mathcal{W}_0$*:*

$$\log \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) > p + \log m. \tag{10}$$

Recall that the transcript π_1 is obtained by invoking the protocol Π on inputs of the form (W, a) and (W, b). Intuitively, Proposition 3.15 means that when we restrict ourselves to $\mathcal{W}_0$, the transcript π_1 reveals at most p bits of information about the matrix W, and still it has to transmit more than $p + \log m$ bits to solve mKW_f on (a, b).

Warm-up. Before we explain the construction of the hard collection, we first present a simplified version of the argument. Let $R \subseteq [m]$ denote the set of rows of W on which π_1 reveals more than two bits of information. Since π_1 reveals at most p bits of information about the whole matrix W, it follows that $|R| \leq p/2$.

We would now like to choose column vectors $a^W \in \mathcal{A}_{\pi_1,W}$ and $b^W \in \mathcal{B}_{\pi_1,W}$, such that for every two matrices W, W' in the collection we have that $a^W|_R \leq b^{W'}|_R$. We start by choosing, for every $W \in \mathcal{W}_0$, a pair of column vectors a^W, b^W that satisfy $a^W|_R \leq b^W|_R$ only for W. To see why this is possible, let $W \in \mathcal{W}_0$, and suppose that such column vectors a^W, b^W did not exist for W. We claim that in this case, it is possible to solve mKW_f on the rectangle $\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}$ by communicating at most

$$|R| + \log m < p + \log m \tag{11}$$

bits, contradicting Equation (10). This is done as follows: By our assumption, for every $a \in \mathcal{A}_{\pi_1,W}$ and $b \in \mathcal{B}_{\pi_1,W}$, it holds that $a_i > b_i$ for some $i \in R$. Alice will send a_R to Bob, and Bob will reply with the corresponding coordinate $i \in R$, thus solving mKW_f using at most $|R| + \log m$ bits.

Hence, we can choose for every matrix $W \in \mathcal{W}_0$ a pair of column vectors a^W, b^W such that $a^W|_R \leq b^W|_R$. It remains to enforce the condition $a^W|_R \leq b^{W'}|_R$ for every *two matrices* W, W'. To

this end, let us denote by α_R the most popular value of $a^W|_R$ over all matrices $W \in \mathcal{W}_0$. We take our hard collection $\mathcal{W}$ to be the subset of matrices $W \in \mathcal{W}_0$ for which $a^W|_R = \alpha_R$, and discard all the other matrices. It now holds for every $W, W' \in \mathcal{W}$ that

$$a^W|_R = \alpha_R = a^{W'}_R \leq b^{W'}_R,$$

as required.

It might seem as if the collection $\mathcal{W}$ satisfies our requirements. Indeed, we have a set of revealed rows R, and $a^W|_R \leq b^{W'}|_R$ holds for every $W, W' \in \mathcal{W}$. However, the above reasoning suffers from the following issue: When we moved from $\mathcal{W}_0$ to $\mathcal{W}$, we revealed additional bits of information about the matrices W. This newly leaked information may create new revealed rows that do not belong to R, thus violating the definition of a hard collection.

The actual proof. We resolve the latter issue by repeating the foregoing argument iteratively: We start by setting $\mathcal{W} = \mathcal{W}_0$ and $R = \emptyset$. Then, in each iteration, we identify a set I of revealed rows, add it to R, and move to a subset of $\mathcal{W}$ in which all the column vectors a^W have the same value α_I. The process ends when there are no more revealed rows. In Section 3.3.2 below, we show that this process yields the following.

Proposition 3.16. *There exists a set of matrices $\mathcal{W} \subseteq \mathcal{W}_0$, a set of revealed rows $R \subseteq [m]$, and for each matrix W, a set $\mathcal{A}^W \subseteq f^{-1}(1)$ of candidates for a^W such that properties are satisfied:*

1. *For every $I \subseteq [m] - R$:*
$$\mathrm{rAvgDeg}_{I \times [\ell]}(\mathcal{W}|_{([m]-R)\times[\ell]}) \geq \frac{1}{4^{|I|}}.$$
2. *All the candidate vectors in $\mathcal{A}^W$ for all the matrices $W \in \mathcal{W}$ agree on the coordinates in R.*
3. *For every $W \in \mathcal{W}$, it holds that $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,W}) > m$.*

Let $\mathcal{W}$, R, and $\mathcal{A}^W$ be the sets obtained from Proposition 3.16. We will choose $\mathcal{W}$ to be the set of matrices in our hard collection. At this point, we know that the set $\mathcal{W}$ satisfies the first condition in the definition of a hard collection due to Property 1 above. We now explain how to choose the column vectors a^W and b^W to satisfy $a^W|_R \leq b^{W'}|_R$ for every $W, W' \in \mathcal{W}$, and this will complete the proof of Lemma 3.13.

For every matrix $W \in \mathcal{W}$, we choose a^W arbitrarily from $\mathcal{A}^W$. By Property 2, all the column vectors a^W of all the matrices W agree on the coordinates in R; let us denote this agreed value by α_R. In order to choose the column vectors b^W, we use the following result.

Claim 3.17. *For every matrix $W \in \mathcal{W}$, there exists a column vector $b^W \in \mathcal{B}_{\pi_1,W}$ such that $b^W|_R \geq \alpha_R$.*

Proof. Let $W \in \mathcal{W}$. Suppose for the sake of contradiction that there exists no column vector $b^W \in \mathcal{B}_{\pi_1,W}$ such that $b^W|_R \geq \alpha_R$. We show that in this case there exists a protocol that solves mKW_f on $\mathcal{A}^W \times \mathcal{B}_{\pi_1,W}$ using $\log m$ bits, which contradicts the fact that $\log \mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,W}) > \log m$ by Property 3.

We use the following protocol: Alice gets a column vector $a \in \mathcal{A}^W$, and Bob gets a column vector $b \in \mathcal{B}_{\pi_1,W}$. Note that $a_R = \alpha_R$ by the definition of α_R. Moreover, by our assumption, it does not hold that $b_R \geq \alpha_R$, and therefore there exists some coordinate $i \in R$ such that $(\alpha_R)_i > b_i$. We know that $a_i = (\alpha_R)_i$, so $a_i > b_i$, and therefore i is a solution for mKW_f on $\mathcal{A}^W \times \mathcal{B}_{\pi_1,W}$. Furthermore, Bob knows b, and also knows α_R (since it does not depend on Alice's input), and therefore he can deduce i. Hence, Bob can send i to Alice, thus solving the problem. It is easy to see that this protocol sends at most $\log m$ bits, so we reached the desired contradiction. ■

We conclude by showing that the column vectors a^W, b^W that we chose satisfy that $a^W|_R \leq b^{W'}|_R$ for every $W, W' \in \mathcal{W}$. Let $W, W' \in \mathcal{W}$. Then, by Claim 3.17,

$$a^W|_R = \alpha_R \leq b^{W'}|_R,$$

as required.

3.3.1 The initial set $\mathcal{W}_0$

We now prove Proposition 3.15, which constructs the initial set $\mathcal{W}_0$ for our argument.

Proposition 3.15 (restated). *There exists a number $p \in \mathbb{N}$ and a set of matrices $\mathcal{W}_0 \subseteq \Lambda^{m\times\ell}$ such that* $\text{density}(\mathcal{W}_0) \geq 2^{-p}$*, and such that for every $W \in \mathcal{W}_0$:*

$$\log \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) > p + \log m.$$

Proof. By assumption, the transcript π_1 is alive, and therefore

$$\gamma(\pi_1) = \frac{1}{|\Lambda^{m\times\ell}|} \cdot \sum_{W\in\Lambda^{m\times\ell}} \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) \geq 4 \cdot m^2.$$

In other words,

$$\sum_{W\in\Lambda^{m\times\ell}} \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) \geq 4 \cdot m^2 \cdot \left|\Lambda^{m\times\ell}\right|.$$

We partition the matrices W into $m - \log m$ buckets as follows: the first bucket $\mathcal{V}_1$ consists of all matrices W for which

$$\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) \leq 2m,$$

and for every $k > 1$, the k-th bucket $\mathcal{V}_k$ consists of all matrices W for which

$$2^{k-1} \cdot m < \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) \leq 2^k \cdot m.$$

For every $k \in [m - \log m]$, we define the *weight of a bucket* $\mathcal{V}_k$ to be the sum

$$\sum_{W\in\mathcal{V}_k} \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}).$$

Our assumption that π_1 is alive says that the total weight of all the buckets together is at least $4 \cdot m^2 \cdot \left|\Lambda^{m\times\ell}\right|$. Moreover, it is easy to see that the weight of $\mathcal{V}_1$ is at most $2 \cdot m \cdot \left|\Lambda^{m\times\ell}\right|$. Hence, the total weight of all buckets except the first bucket is at least

$$4 \cdot m^2 \cdot \left|\Lambda^{m\times\ell}\right| - 2 \cdot m \cdot \left|\Lambda^{m\times\ell}\right| \geq 2 \cdot m^2 \cdot \left|\Lambda^{m\times\ell}\right|.$$

By an averaging argument, there exists $k \in [m - \log m] - \{1\}$ such that the weight of $\mathcal{V}_k$ is at least

$$\frac{2 \cdot m^2 \cdot \left|\Lambda^{m\times\ell}\right|}{m - \log m - 1} \geq 2 \cdot m \cdot \left|\Lambda^{m\times\ell}\right|.$$

We choose $\mathcal{W}_0 \stackrel{\text{def}}{=} \mathcal{V}_k$ and $p \stackrel{\text{def}}{=} k - 1$. By definition, for every $W \in \mathcal{W}_0$ we have

$$\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) > 2^{k-1} \cdot m = 2^p \cdot m$$

and hence

$$\log \mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) > p + \log m.$$

It remains to lower bound the size of $\mathcal{W}_0$. To this end, recall that the weight of $\mathcal{W}_0$ is at least $2 \cdot m \cdot \left|\Lambda^{m\times\ell}\right|$. On the other hand, for every $W \in \mathcal{W}_0$:

$$\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) \leq 2^k \cdot m = 2^{p+1} \cdot m.$$

Hence, the weight of $\mathcal{W}_0$ is at most $|\mathcal{W}_0| \cdot 2^{p+1} \cdot m$. It follows that

$$\begin{aligned} |\mathcal{W}_0| \cdot 2^{p+1} \cdot m &\geq 2 \cdot m \cdot \left|\Lambda^{m\times\ell}\right| \\ |\mathcal{W}_0| &\geq \frac{2 \cdot m \cdot \left|\Lambda^{m\times\ell}\right|}{2^{p+1} \cdot m} = 2^{-p} \cdot \left|\Lambda^{m\times\ell}\right|, \end{aligned}$$

as required. ■

3.3.2 The iterative procedure

We conclude the proof of the lemma by proving Proposition 3.16, restated next.

Proposition 3.16 (restated). *There exists a set of matrices $\mathcal{W} \subseteq \mathcal{W}_0$, a set of revealed rows $R \subseteq [m]$, and for each matrix W, a set $\mathcal{A}^W \subseteq f^{-1}(1)$ of candidates for a^W such that properties are satisfied:*

1. *For every $I \subseteq [m] - R$:*
$$\mathrm{rAvgDeg}_{I\times[\ell]}(\mathcal{W}|_{([m]-R)\times[\ell]}) \geq \frac{1}{4^{|I|}}.$$
2. *All the candidate vectors in $\mathcal{A}^W$ for all the matrices $W \in \mathcal{W}$ agree on the coordinates in R.*
3. *For every $W \in \mathcal{W}$, it holds that $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,W}) > m$.*

In order to streamline the presentation, we denote the set of unrevealed rows by $U \stackrel{\text{def}}{=} [m] - R$. For convenience, throughout the procedure we will maintain the property that every partial matrix $W' \in \mathcal{W}|_{U\times[\ell]}$ has a unique extension to a matrix $W \in \mathcal{W}$. Intuitively, this property is convenient since only the value of the unrevealed rows of a matrix matters. We refer to this invariant as the *unique extension property*.

Let $\mathcal{W}_0$ be the set of matrices obtained from Proposition 3.15. The procedure starts by setting $\mathcal{W} = \mathcal{W}_0$, $R = \emptyset$, and $\mathcal{A}^W = \mathcal{A}_{\pi_1,W}$ for every $W \in \mathcal{W}$. Now, as long as there exists a non-empty set $I \subseteq U$ such that

$$\mathrm{rAvgDeg}_{I\times[\ell]}(\mathcal{W}|_{U\times[\ell]}) < \frac{1}{4^{|I|}},$$

we perform the following steps:

1. We add I to R (and remove I from U).
2. We restore the unique extension invariant by choosing for every partial matrix $W' \in \mathcal{W}|_{U\times[\ell]}$ a single extension $W \in \mathcal{W}$, and removing all the other extensions of W' from $\mathcal{W}$.
3. For every $W \in \mathcal{W}$, we make sure that all column vectors in $\mathcal{A}^W$ agree on the coordinates in I as follows:

(a) For each $W \in \mathcal{W}$, we partition $\mathcal{A}^W$ to buckets $\left\{\mathcal{A}^{W,v}\right\}_{v\in\{0,1\}^I}$, such that the bucket $\mathcal{A}^{W,v}$ contains the column vectors $a \in \mathcal{A}^W$ that satisfy $a_I = v$.

(b) Let v_W be the value that maximizes $\mathsf{mL}(\mathcal{A}^{W,v} \times \mathcal{B}_{\pi_1,W})$.

(c) We replace $\mathcal{A}^W$ with the bucket $\mathcal{A}^{W,v_W}$.

4. Finally, we make sure that all column vectors *of all matrices* agree on the coordinates in I as follows:

(a) Let α_I be the most popular value among all the v_W's.

(b) We replace $\mathcal{W}$ with the subset of matrices W for which $v_W = \alpha_I$.

By definition, when the procedure ends, Property 1 of Proposition 3.16 is satisfied. Moreover, it is easy to see that Property 2 is satisfied.

It remains to show that Property 3 is satisfied. To this end, recall that when the procedure starts, every $W \in \mathcal{W}$ satisfies $\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) > 2^p \cdot m$ by the definition of $\mathcal{W}_0$. Next, observe in every iteration, Step 3 decreases $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,\mathcal{W}})$ by a factor of at most $2^{|I|}$ by the subadditivity of $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,\mathcal{W}})$. All the other steps of the procedure do not affect $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,\mathcal{W}})$ at all. Hence, by the time the procedure halts, the value $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,\mathcal{W}})$ has decreased by a factor of at most $2^{|R|}$, so $\mathsf{mL}(\mathcal{A}_{\pi_1,W} \times \mathcal{B}_{\pi_1,W}) > 2^{p-|R|} \cdot m$. Thus, to prove that $\mathsf{mL}(\mathcal{A}^W \times \mathcal{B}_{\pi_1,W}) > m$, it suffices to show that $|R| \leq p$, which we establish next.

Claim 3.18. *When the procedure halts, $|R| \leq p$.*

Proof. We upper bound the size of R using a potential argument. Intuitively, the potential function is the amount of information the players know about the rows in U. At the beginning of the process, $U = [m]$, and the players know p bits of information about all the rows together. For every revealed row i that is added to R, the potential is decreased by at least two, since the two bits that the players knew about the row i are discarded. Then, when the value a_i is fixed to a constant α_i, it reveals at most one bit of information, thus increasing the potential by at most one. All in all, each revealed row that is added to R decreases the potential function by at least one. Since the potential starts from p and is always non-negative, it follows that the number of revealed rows will never surpass p, which is what we wanted to prove.

Formally, our potential function is the density of $\mathcal{W}|_{U\times[\ell]}$. Recall that at the beginning of this procedure, this density is at least 2^{-p} by the definition of $\mathcal{W}_0$. We prove that in every iteration, the density of $\mathcal{W}|_{U\times[\ell]}$ increases by a factor of at least $2^{|I|}$, where I is the set of rows that is added to R at the iteration. Note that this implies the claim, since the density of a set can never exceed 1, and R consists of the union of all the sets I.

Fix a single iteration. By assumption, at the beginning of the iteration we have

$$\mathrm{rAvgDeg}_{I\times[\ell]}(\mathcal{W}|_{U\times[\ell]}) < \frac{1}{4^{|I|}}.$$

In Step 1, the procedure removes I from U. To see how this step affects the density of $\mathcal{W}|_{U\times[\ell]}$, observe that Fact 2.37 implies that

$$\mathrm{density}\left(\mathcal{W}|_{(U-I)\times[\ell]}\right) \geq \frac{1}{\mathrm{rAvgDeg}_{I\times[\ell]}(\mathcal{W}|_{U\times[\ell]})} \cdot \mathrm{density}(\mathcal{W}|_{U\times[\ell]}) > 4^{|I|} \cdot \mathrm{density}(\mathcal{W}|_{U\times[\ell]}).$$

Thus, Step 1 increases the density by a factor of $4^{|I|}$. Steps 2 and 3 do not affect the density of $\mathcal{W}|_{U\times[\ell]}$ at all. Finally, it is not hard to see that Step 4 decreases the size of $\mathcal{W}|_{U\times[\ell]}$ by a factor

of at most $2^{|I|}$. All in all, at the end of the iteration, the density of $\mathcal{W}|_{U\times[\ell]}$ is increased by at least a factor of $2^{|I|}$, as required. ■

This concludes the proof of Lemma 3.13.

3.4 Proof of Lemma 3.14

In this section, we prove Lemma 3.14. Let π_1 be a transcript that supports a hard collection $\mathcal{W}$, and let $\mathcal{X}_{\pi_1} \times \mathcal{Y}_{\pi_1}$ be its associated rectangle. Our goal is to prove that the communication complexity of solving $mKW_f \circledast S_{\mathrm{gd}}$ on the inputs in $\mathcal{X}_{\pi_1} \times \mathcal{Y}_{\pi_1}$ is at least $\Omega(q \cdot t)$. We use the following proof strategy: We observe that solving $mKW_f \circledast S_{\mathrm{gd}}$ on $\mathcal{X}_{\pi_1} \times \mathcal{Y}_{\pi_1}$ amounts to solving sub-problem H of some lifted problem $S' \diamond \mathrm{gd}$. Then, we apply to H our generalized lifting theorem, which deals with sub-problems of lifted search problems, thus obtaining lower bound on $mKW_f \circledast S_{\mathrm{gd}}$. More details follow.

Let R be the set of revealed rows of the hard collection $\mathcal{W}$, and let $U \stackrel{\mathrm{def}}{=} [m] - R$ denote the set of unrevealed rows. Let $\mathcal{W}'$ denote the projection of the matrices in $\mathcal{W}$ to the rows in U. We consider the following communication problem.

Definition 3.19. The communication problem H is defined as follows: Alice gets a matrix $X' \in \mathcal{W}'$, Bob gets a matrix $Y' \in \mathcal{W}'$, and their goal is to output $(i, o) \in U \times \mathcal{O}$ such that $o \in S_{\mathrm{gd}}(X'_i, Y'_i)$.

Proposition 3.20. *H reduces to solving $mKW_f \circledast S_{\mathrm{gd}}$ on the inputs in $\mathcal{X}_{\pi_1} \times \mathcal{Y}_{\pi_1}$.*

Proof. We define the functions $R_A, R_B, R_{\mathrm{out}}$ of the reduction. Given an input $X' \in \mathcal{W}'$ of Alice in H, the function R_A translates it to an input (X, a^X) of Alice in $mKW_f \circledast S_{\mathrm{gd}}$, where $X \in \mathcal{W}$ is an arbitrary fixed extension of X' to a matrix in $\mathcal{W}$. We define $R_B(Y') \stackrel{\mathrm{def}}{=} (Y, b^Y)$ similarly. Finally, we set R_{out} to be the identity function.

Observe that the outputs (X, a^X) and (Y, b^Y) of this reduction are indeed inputs in $\mathcal{X}_{\pi_1} \times \mathcal{Y}_{\pi_1}$, since π_1 supports the collection $\mathcal{W}$. It remains to show that if (i, o) is a solution for $mKW_f \circledast S_{\mathrm{gd}}$ on inputs (X, a^X) and (Y, b^Y), then it is a solution for H on (X', Y'). To see it, recall that the assumption that (i, o) is a solution for $mKW_f \circledast S_{\mathrm{gd}}$ implies that $a^X_i > b^Y_i$ and that $o \in S_{\mathrm{gd}}(X_i, Y_i)$. In particular, it must hold that $i \in U$, since by assumption $a^X_i \leq b^Y_i$ for every $i \in R$. Therefore, (i, o) is a solution for H on (X', Y'). as required. ■

It remains to prove a lower bound of $\Omega(q \cdot t)$ on $\mathsf{CC}(H)$. To this end, we show that H is (a sub-problem of) a lifted search problem $S' \diamond \mathrm{gd}$. Consider the following search problem S': given a matrix $Z \in \{0,1\}^{U\times[\ell]}$, we would like to find a pair (i, o) such that o is a solution for S on Z_i (i.e., $o \in S(Z_i)$). Now, consider the corresponding lifted search problem $S'_{\mathrm{gd}} \stackrel{\mathrm{def}}{=} S' \diamond \mathrm{gd}$, and observe that it can be described as follows: Alice gets a matrix $X' \in \Lambda^{U\times[\ell]}$, Bob gets a matrix $Y' \in \Lambda^{U\times[\ell]}$, and their goal is to find a pair $(i, o) \in U \times \mathcal{O}$ such that $o \in S_{\mathrm{gd}}(X_i, Y_i)$. Hence, the problem H is simply the restriction of the lifted search problem $S' \diamond \mathrm{gd}$ to input matrices that come from the set $\mathcal{W}'$.

It is not hard to see that the query complexity of the problem S' is $q \stackrel{\mathrm{def}}{=} \mathsf{Q}(S)$: indeed, if we had a decision tree T that solves S' using less than q queries, we could have used T to solve S with less than q queries by invoking T on matrices whose rows are all equal. The lifting theorem of [CFK+19] (Theorem 2.26) implies that $\mathsf{CC}(S' \diamond \mathrm{gd}) \geq \Omega(q \cdot t)$. In order to prove a similar lower bound for H, we use our generalized lifting theorem. This generalization applies to lifted search problems when restricted to sets of inputs that have sufficiently large average degree. Formally, the theorem says the following.

Theorem 5.1 (restated). *For every $\eta > 0$ and $d \in \mathbb{N}$ there exist $c \in \mathbb{N}$ and $\kappa > 0$ such that the following holds: Let S be a search problem that takes inputs from $\{0,1\}^\ell$, and let* $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ *be an arbitrary function such that* $\mathrm{disc}(\mathrm{gd}) \leq 2^{-\eta \cdot t}$ *and such that* $t \geq c \cdot \log \ell$. *Let $\mathcal{X}, \mathcal{Y} \subseteq \left(\{0,1\}^t\right)^\ell$ such that for every $I \subseteq [\ell]$ both* $\mathrm{rAvgDeg}_I(\mathcal{X})$ *and* $\mathrm{rAvgDeg}_I(\mathcal{Y})$ *are at least* $1/(d \cdot \ell^d)^{|I|}$. *Then the communication complexity of solving $S \diamond \mathrm{gd}$ on inputs from $\mathcal{X} \times \mathcal{Y}$ is at least $\kappa \cdot \mathsf{Q}(S) \cdot t$.*

In order to apply Theorem 5.1 to H, we need to lower bound the average degree of every set of entries $K \subseteq U \times [\ell]$ in $\mathcal{W}'$. This is done in the following claim.

Claim 3.21. *For every set of entries $K \subseteq U \times [\ell]$, it holds that* $\mathrm{rAvgDeg}_K(\mathcal{W}') \geq \frac{1}{4^{|K|}}$.

Before proving the claim, we show how to use it to our lower bound on H. We apply Theorem 5.1 with $S = S'$, $\mathcal{X} = \mathcal{Y} = \mathcal{W}$, $\eta = \eta$, and $d = 4$. We choose the constant c to be the corresponding constant that are obtained from the application of Theorem 5.1. It now follows that $\mathsf{CC}(H) \geq \kappa \cdot q \cdot t$ for some constant $\kappa > 0$, which completes the proof of Lemma 3.14.

Proof of Claim 3.21. Intuitively, we need to prove that for every set $K \subseteq U \times [\ell]$ of *entries*, the players know at most $2|K|$ bits of information. By the assumption that $\mathcal{W}$ is a hard collection, we know that on any set $I \subseteq U$ of *rows*, the players know at most $2|I|$ bits of information. Since every set of entries K in contained in at most $|K|$ rows, the claim follows. We now formalize this intuition.

Let $K \subseteq U \times [\ell]$ be a set of entries, and let $I \subseteq U$ be the set of rows that contain entries from K. By the assumption that $\mathcal{W}$ is a hard collection,

$$\mathrm{rAvgDeg}_{I \times [\ell]}(\mathcal{W}') \geq \frac{1}{4^{|I|}}.$$

By the "chain rule" for average degree (Fact 2.38) it holds that

$$\mathrm{rAvgDeg}_{I \times [\ell]}(\mathcal{W}') = \mathrm{rAvgDeg}_K(\mathcal{W}') \cdot \mathrm{rAvgDeg}_{I \times [\ell] - K}(\mathcal{W}'_{U \times [\ell] - K}),$$

and since relative average degree is always at most 1 it follows that

$$\mathrm{rAvgDeg}_K(\mathcal{W}') \geq \mathrm{rAvgDeg}_{I \times [\ell]}(\mathcal{W}') \geq \frac{1}{4^{|I|}} \geq \frac{1}{4^{|K|}},$$

as required. ■

4 The semi-monotone composition theorem

In this section we prove our semi-monotone composition theorem. Recall that this theorem can be stated informally as follows.

Theorem 1.5. *Let $g : \{0,1\}^n$ be a non-constant monotone function, and let* eq *be the equality function on strings of length t. Suppose there exists a lifted search problem $S_\phi \diamond \mathrm{eq}$ that reduces to mKW_g via an injective reduction and satisfies the conditions of the theorem of [dRMN+19]. Then*

$$\mathsf{CC}(U_m \diamond mKW_g) \geq m + \Omega(NS_{\mathbb{F}_2}(\phi) \cdot t).$$

Formally, the theorem says the following.

Theorem 4.1 (semi-monotone composition theorem)**.** *Let $m \in \mathbb{N}$ and let $g : \{0,1\}^n \to \{0,1\}$ be a non-constant monotone function, and let* eq *be the equality function on strings of length t. Suppose that there exists a q-CNF contradiction ϕ over ℓ variables, such that the lifted search problem $S_\phi \diamond \mathrm{eq}$ reduces to mKW_g via an injective reduction and such that $t \geq 2\log \ell$. Then,*

$$\log \mathsf{L}(U_m \diamond mKW_g) \geq m + \Omega(NS_{\mathbb{F}_2}(\phi) \cdot t) - \log q. \tag{12}$$

The rest of this section is organized as follows. We start by setting up by some notation. Then, we define a subproblem of $U_m \diamond mKW_g$, denoted $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$. Finally, we prove the desired lower bound on $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$ using three propositions, which are proved in turn in Sections 4.1, 4.2, and 4.3 below.

Let $m, g, \mathrm{eq}, \phi, S_\phi$ be as in the theorem. For simplicity of notation, we denote $\Lambda \stackrel{\mathrm{def}}{=} \{0,1\}^t$, so the domain of the lifted search problem $S_\phi \diamond \mathrm{eq}$ is $\Lambda^\ell \times \Lambda^\ell$. Let $R_A : \Lambda^\ell \to g^{-1}(1)$, $R_B : \Lambda^\ell \to g^{-1}(0)$, and $R_{\mathrm{out}} : [n] \to \mathcal{O}$ denote the assumed reduction from $S_{\mathrm{gd}} \diamond \mathrm{eq}$ to mKW_g, and recall that the functions R_A and R_B are injective. Let $\mathcal{X} \stackrel{\mathrm{def}}{=} R_A(\Lambda^\ell)$ and $\mathcal{Y} \stackrel{\mathrm{def}}{=} R_B(\Lambda^\ell)$ denote the images of R_A and R_B respectively, and observe that $|\mathcal{X}| = |\mathcal{Y}| = \left|\Lambda^\ell\right|$. We denote the latter size by $K \stackrel{\mathrm{def}}{=} \left|\Lambda^\ell\right|$. For every $p \in \mathbb{N}$, we denote by I_p the identity matrix of order p, and we denote by $I \stackrel{\mathrm{def}}{=} I_K$ the identity matrix of order K. Finally, we let $\mathcal{W} \subseteq \{0,1\}^{m\times n}$ be the set of $m \times n$ matrices all of whose rows belong to $\mathcal{X} \cup \mathcal{Y}$.

We turn to define the subproblem $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$. Recall that in the introduction, the communication problem $U_m \diamond mKW_g$ was defined as follows: Alice and Bob get matrices $X, Y \in \{0,1\}^{m\times n}$, and denote by a and b the column vectors that are obtained by applying g to the rows of X and Y respectively. The players are promised that $a \neq b$, and they should either solve mKW_g on a row where $a_i \neq b_i$ or find (i,j) such that $a_i = b_i$ and $X_{i,j} \neq Y_{i,j}$.

In the subproblem $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$, we restrict the input matrices of the players to come from the set $\mathcal{W}$. We also change the problem a bit as follows: we do not promise the players that $a \neq b$, but rather, if the players find that $a = b$ they are allowed to declare failure. It is not hard to see that this modification changes the complexity of the problem by at most two bits (see [HW93]), and it makes the problem easier to analyze since it ensures that the domain of the problem is a combinatorial rectangle. Formally, the sub-problem $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$ is defined as follows.

Definition 4.2. The communication problem $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$ is defined as follows: The inputs of Alice and Bob are matrices $X, Y \in \mathcal{W}$ respectively. Let a and b denote the column vectors that are obtained by applying g to the rows of X and Y respectively. The goal of the players is to find an entry (i,j) that satisfies one of the following three options:

- $a_i > b_i$ and $X_{i,j} > Y_{i,j}$.
- $a_i < b_i$ and $X_{i,j} < Y_{i,j}$.
- $a_i = b_i$ and $X_{i,j} \neq Y_{i,j}$.

In addition, if $a = b$ then players are allowed to output the failure symbol $\perp$ instead of an entry (i,j).

Proof of Theorem 4.1. We prove the theorem by establishing a lower bound on the Razborov rank measure of $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$ (see Section 2.5). To this end, we construct a matrix $M \in \mathbb{F}_2^{\mathcal{W}\times\mathcal{W}}$, and show that

$$\log \mu_{\mathbb{F}_2}(U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}, M) \geq m + \Omega(NS_{\mathbb{F}_2}(\phi) \cdot t) - \log q.$$

We start with the following proposition, which constructs a matrix $A \in \mathbb{F}_2^{\mathcal{X}\times\mathcal{Y}}$ that will be used as a building block in the construction of M, and which is proved in Section 4.3 below.

Proposition 4.3. *There exists a symmetric matrix $A \in \mathbb{F}_2^{\mathcal{X}\times\mathcal{Y}}$ such that*

$$\log \mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A) \geq \Omega(NS_{\mathbb{F}_2}(\phi)\cdot t) - \log q,$$

and such that $A^2 = I$.

We now construct the matrix M is constructed as follows. Recall that the rows and columns of M are indexed by matrices $X, Y \in \mathcal{W}$. We can associate them with corresponding column vectors $a, b \in \{0,1\}^m$. The matrix M is a block matrix, which consists of $2^m \cdot 2^m$ blocks — one for each pair (a, b). The blocks that correspond to pairs where $a = b$ are all-zeroes. For every other block, we take the Kronecker product of m matrices, where the i-th matrix is A (if $a_i \neq b_i$) or I (if $a_i = b_i$). More formally, for any two bits $\gamma, \delta \in \{0,1\}$ we denote

$$A^{\gamma,\delta} \stackrel{\text{def}}{=} \begin{cases} A & \text{if } \gamma \neq \delta \\ I & \text{otherwise.} \end{cases}$$

Then, for every $a, b \in \{0,1\}^m$, the block of M that correspond to the pair (a, b) is

$$\begin{cases} A^{a_1,b_1} \otimes A^{a_2,b_2} \otimes \cdots \otimes A^{a_m,b_m} & a \neq b \\ \text{all zeroes} & a = b \end{cases}.$$

Intuitively, on rows where $a_i \neq b_i$, the players should solve $mKW_{\mathcal{X}\times\mathcal{Y}}$, so we put the matrix A which is "hard" for $mKW_{\mathcal{X}\times\mathcal{Y}}$. On the other hand, on rows where $a_i = b_i$, the players should verify the inequality of strings from $\mathcal{X} \cup \mathcal{Y}$, so we put the matrix I which is "hard" for this task.

We turn to prove the lower bound on $\mu_{\mathbb{F}_2}(U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}, M)$. To this end, we prove a lower bound on the ratio $\text{rank}_{\mathbb{F}_2}(M)/\text{rank}_{\mathbb{F}_2}(M|_R)$ over all the monochromatic rectangles R of $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$. This is done in the following two propositions, which bound the the numerator and denominator in the latter ratio, and are proved in Sections 4.1 and 4.2 respectively.

Proposition 4.4. *The matrix M has full rank, i.e.,* $\log \text{rank}_{\mathbb{F}_2}(M) = \log |\mathcal{W}|$.

Proposition 4.5. *For every monochromatic rectangle R of $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$,*

$$\log \text{rank}_{\mathbb{F}_2}(M|_R) \leq \log |\mathcal{W}| - m - \log \mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A).$$

Together, the above two propositions immediately imply the desired lower bound on $\mu_{\mathbb{F}_2}(U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}, M)$, and hence, Theorem 4.1. ■

We now establish some notation that will be used in the proofs of both Proposition 4.4 and Proposition 4.5. First, we define an auxiliary matrix $M' \in \mathbb{F}_2^{\mathcal{W}}$ as follows: the matrix M' that is defined similarly to M, except that the blocks where $a = b$ are not treated differently. In other words, M' is a block matrix that, for every $a, b \in \{0,1\}^m$, has the block $A^{a_1,b_1} \otimes A^{a_2,b_2} \otimes \cdots \otimes A^{a_m,b_m}$. Observe that the blocks where $a = b$ are equal to I, and that those blocks are placed along the main diagonal of M'. Thus, $M' = M + I_{|\mathcal{W}|}$.

We denote by $M_{(m-1)}$ and $M'_{(m-1)}$ the versions of M and M' that are defined for $m-1$ rather than m — in other words, those are the matrices M and M' that we would define for $U_{m-1} \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$.

4.1 The rank of M

We start by proving Proposition 4.4, which says that M has full rank. We first claim that

$$M = \begin{pmatrix} I \otimes M_{(m-1)} & A \otimes M'_{(m-1)} \\ A \otimes M'_{(m-1)} & I \otimes M_{(m-1)} \end{pmatrix}. \tag{13}$$

The equality holds for the following reason: The upper and lower halves of M correspond to the cases where $a_1 = 0$ and $a_1 = 1$ respectively, and the left and right halves of M correspond to the cases where $b_1 = 0$ and $b_1 = 1$. By Fact 2.44, the matrix $I \otimes M_{(m-1)}$ is[1] the block matrix that is obtained by taking the Kronecker product of I with each block of $M_{(m-1)}$, and these are exactly the blocks of M that correspond to $a_1 = b_1$. Similarly, the matrix $A \otimes M'_{(m-1)}$ is the block matrix that is obtained by taking the Kronecker product of A with each block of $M'_{(m-1)}$, and these are exactly the blocks of M that correspond to $a_1 \neq b_1$: here, we used $M'_{(m-1)}$ rather than $M_{(m-1)}$ since all those blocks satisfy $a \neq b$, and therefore we do not want to zero out the blocks when $a_{-1} = b_{-1}$ (where a_{-1}, b_{-1} denote the column vectors a, b without the first coordinate).

We prove that M has full rank by applying row and column operations to Equation (13). Let I' be the identity matrix of the same order as $M_{(m-1)}$, and recall that $M'_{(m-1)} = M_{(m-1)} + I'$. Since we are working over $\mathbb{F}_2$, the latter equality can also be written as $M_{(m-1)} = M'_{(m-1)} + I'$. By substituting the latter equality in Equation (13), we obtain the matrix

$$\begin{pmatrix} I \otimes (M'_{(m-1)} + I') & A \otimes M'_{(m-1)} \\ A \otimes M'_{(m-1)} & I \otimes (M'_{(m-1)} + I') \end{pmatrix} = \begin{pmatrix} I \otimes M'_{(m-1)} + I \otimes I' & A \otimes M'_{(m-1)} \\ A \otimes M'_{(m-1)} & I \otimes M'_{(m-1)} + I \otimes I' \end{pmatrix}.$$

Next, we subtract the product of $A \otimes I'$ and the right half from the left half, and obtain the matrix

$$\begin{pmatrix} I \otimes M'_{(m-1)} + I \otimes I' - (A \otimes I') \cdot (A \otimes M'_{(m-1)}) & A \otimes M'_{(m-1)} \\ A \otimes M'_{(m-1)} - (A \otimes I') \cdot (I \otimes M'_{(m-1)}) - (A \otimes I') \cdot (I \otimes I') & I \otimes M'_{(m-1)} + I \otimes I' \end{pmatrix}. \tag{14}$$

We now use Fact 2.41 to determine each of the matrix products that appear in the last expression. Recall that $A^2 = I$ by Proposition 4.3. Then,

$$\begin{aligned} (A \otimes I') \cdot (A \otimes M'_{(m-1)}) &= (A \cdot A) \otimes (I' \cdot M'_{(m-1)}) = I \otimes M'_{(m-1)} \\ (A \otimes I') \cdot (I \otimes M'_{(m-1)}) &= (A \cdot I) \otimes (I' \cdot M'_{(m-1)}) = A \otimes M'_{(m-1)} \\ (A \otimes I') \cdot (I \otimes I') &= (A \cdot I) \otimes (I' \cdot I') = A \otimes I'. \end{aligned} \tag{15}$$

By substituting the latter equalities in the matrix of Equation (14), we obtain the matrix

$$\begin{aligned} &\begin{pmatrix} I \otimes M'_{(m-1)} + I \otimes I' - I \otimes M'_{(m-1)} & A \otimes M'_{(m-1)} \\ A \otimes M'_{(m-1)} - A \otimes M'_{(m-1)} - A \otimes I' & I \otimes M'_{(m-1)} + I \otimes I' \end{pmatrix} \\ =&\begin{pmatrix} I \otimes I' & A \otimes M'_{(m-1)} \\ A \otimes I' & I \otimes M'_{(m-1)} + I \otimes I' \end{pmatrix} \end{aligned}$$

where in the last equality we used the fact that we are working over $\mathbb{F}_2$. We now subtract the product of $A \otimes I'$ and the upper half from the lower half, and obtain the matrix

$$\begin{pmatrix} I \otimes I' & A \otimes M'_{(m-1)} \\ A \otimes I' - (A \otimes I) \cdot (I \otimes I') & I \otimes M'_{(m-1)} + I \otimes I' - (A \otimes I') \cdot (A \otimes M'_{(m-1)}) \end{pmatrix}.$$

[1] Here, we apply Fact 2.44 with I being the "block matrix" that has a single block.

By substituting the equalities of Equation (15) in the latter expression, we obtain the matrix

$$\begin{pmatrix} I \otimes I' & A \otimes M'_{(m-1)} \\ A \otimes I' - A \otimes I & I \otimes M'_{(m-1)} + I \otimes I' - I \otimes M'_{(m-1)} \end{pmatrix}$$
$$= \begin{pmatrix} I \otimes I' & A \otimes M'_{(m-1)} \\ 0 & I \otimes I' \end{pmatrix}.$$

The latter matrix is an upper triangular matrix that has ones on its main diagonal, and therefore has full rank, as required.

4.2 The rank of monochromatic rectangles

We turn to prove Proposition 4.5, which upper bounds the rank of monochromatic rectangles. Let $R \subseteq \mathcal{W} \times \mathcal{W}$ be a monochromatic rectangle of $U_m \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$. We prove that

$$\text{rank}_{\mathbb{F}_2}(M|_R) \leq \frac{|\mathcal{W}|}{2^m \cdot \mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A)}.$$

Recall that R can be one of four types:

1. It could correspond to a solution (i, j) where $a_i > b_i$ and $X_i > Y_i$.
2. It could correspond to a solution (i, j) where $a_i < b_i$ and $X_i < Y_i$.
3. It could correspond to a solution (i, j) where $a_i = b_i$ and $X_{i,j} \neq Y_{i,j}$.
4. It could correspond to the failure symbol $\perp$, which means that $a = b$.

We consider each of the types separately, starting with the simpler Types 3 and 4. If R is of Type 4, every entry $(X, Y) \in R$ satisfies $a = b$, and by the definition of M, this implies that $M_{X,Y} = 0$. Hence, $M|_R$ is the all-zeroes matrix and therefore $\text{rank}_{\mathbb{F}_2}(M|_R) = 0$.

If R is of Type 3, there exist some $i \in [m]$ and $j \in [n]$ such that every entry $(X, Y) \in R$ satisfies $a_i = b_i$ and $X_{i,j} \neq Y_{i,j}$. We show that in this case, $M|_R$ is again the all-zeroes matrix. Without loss of generality, assume that $i = 1$. If $a = b$, then again $M_{X,Y} = 0$. Otherwise, by the definition of M, the block that corresponds to (a, b) is equal to

$$I \otimes A^{a_2,b_2} \otimes \cdots \otimes A^{a_m,b_m},$$

and thus the entry that corresponds to (X, Y) is equal to

$$M_{X,Y} = I_{X_1,Y_1} \cdot \prod_{i=2}^{m} (A^{a_i,b_i})_{X_i,Y_i}.$$

Since $X_1 \neq Y_1$, we have $I_{X_1,Y_1} = 0$ and thus $M_{X,Y} = 0$ as well. Hence, $M|_R$ is the all-zeroes matrix and therefore $\text{rank}_{\mathbb{F}_2}(M|_R) = 0$.

The bulk of the proof is devoted to the case where R is of Type 1 (the case where R is of Type 2 can be dealt with similarly since A is symmetric). Assume that R corresponds to a solution (i, j) where $a_i > b_i$ and $X_{i,j} > Y_{i,j}$. Without loss of generality, assume that $i = 1$. Moreover, without loss of generality, we may assume that R is maximal, since extending R can only increase

the rank of $M|_R$. This implies that R can be assumed to contain *all* inputs that satisfy $a_i > b_i$ and $X_{i,j} > Y_{i,j}$. In other words, R can be written as $R = \mathcal{U} \times \mathcal{V}$ where:

$$\mathcal{U} \stackrel{\text{def}}{=} \{X \in \mathcal{W} : a_1 = 1, X_{1,j} = 1\} = \{X \in \mathcal{W} : X_1 \in \mathcal{X}, X_{1,j} = 1\}$$
$$\mathcal{V} \stackrel{\text{def}}{=} \{Y \in \mathcal{W} : b_1 = 0, Y_{1,j} = 0\} = \{Y \in \mathcal{W} : Y_1 \in \mathcal{Y}, Y_{1,j} = 0\},$$

where the second equality in each line holds since $\mathcal{X} \subseteq g^{-1}(1)$ and $\mathcal{Y} \subseteq g^{-1}(0)$. Now, define a rectangle $R^* \subseteq \mathcal{X} \times \mathcal{Y}$ by

$$R^* \stackrel{\text{def}}{=} \{x \in \mathcal{X} : x_j = 1\} \times \{y \in \mathcal{Y} : y_j = 0\}.$$

Then, we can write

$$R = \{(X, Y) \in \mathcal{W} \times \mathcal{W} | (X_1, Y_1) \in R^*\}.$$

Recall that we denote by $M_{(m-1)}$ and $M'_{(m-1)}$ the versions of M and M' for $U_{m-1} \diamond mKW_{\mathcal{X}\times\mathcal{Y}}$. It follows that

$$M|_R = A|_{R^*} \otimes M'_{(m-1)},$$

where we use $M'_{(m-1)}$ rather than $M_{(m-1)}$ since $a \neq b$ for all the entries in R. In order to bound the rank of this matrix, we use the following proposition, whose proof is deferred to the end of this section.

Proposition 4.6. *It holds that* $\text{rank}_{\mathbb{F}_2}(M') = K^m$.

Observe that $|\mathcal{W}| = (2K)^m$: to see why, recall that $\mathcal{W}$ consists of all $m \times n$ matrices whose rows come from $\mathcal{X} \cup \mathcal{Y}$. The sets $\mathcal{X}, \mathcal{Y}$ are disjoint and satisfy $|\mathcal{X}| = |\mathcal{Y}| = K$, and hence $|\mathcal{W}| = (|\mathcal{X} \cup \mathcal{Y}|)^m = (2K)^m$. Moreover, observe that $\text{rank}_{\mathbb{F}_2}(A) = K$, since $A^2 = I$ and so A has full rank. It follows that

$$\begin{aligned}
\text{rank}_{\mathbb{F}_2}(M|_R) &= \text{rank}_{\mathbb{F}_2}(A|_{R^*}) \cdot \text{rank}_{\mathbb{F}_2}(M'_{(m-1)}) \\
&= \text{rank}_{\mathbb{F}_2}(A|_{R^*}) \cdot K^{m-1} \\
&\leq \frac{\text{rank}_{\mathbb{F}_2}(A)}{\mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A)} \cdot K^{m-1} && \text{(By definition of } \mu_{\mathbb{F}_2}) \\
) &= \frac{K}{\mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A)} \cdot K^{m-1} && (\text{rank}_{\mathbb{F}_2}(A) = K) \\
&= \frac{K^m}{\mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A)} \\
&= \frac{|\mathcal{W}|}{2^m \cdot \mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A)}. && (|\mathcal{W}| = (2K)^m)
\end{aligned}$$

The last equality holds since This concludes the proof.

Proof of Proposition 4.6. Let B denote the block matrix

$$B \stackrel{\text{def}}{=} \begin{pmatrix} I & A \\ A & I \end{pmatrix}.$$

We claim that $M' = \underbrace{B \otimes \cdots \otimes B}_{m \text{ times}}$. To see why, note that the upper and lower halves of B correspond to the cases where $a_i = 0$ and $a_i = 1$ respectively, and the left and right halves correspond to the

cases where $b_i = 0$ and $b_i = 1$. Hence, by Fact 2.44, when we take the Kronecker product of m copies of B we get all the possible blocks of the form $A^{a_1,b_1} \otimes A^{a_2,b_2} \otimes \cdots \otimes A^{a_m,b_m}$..

It therefore suffices to prove that $\text{rank}_{\mathbb{F}_2}(B) = K$, since that will imply that $\text{rank}_{\mathbb{F}_2}(M') = K^m$ by Fact 2.43. To this end, we subtract the product of A with the upper half of B from the lower half of B, and obtain the matrix

$$\begin{pmatrix} I & A \\ A - A \cdot I & I - A^2 \end{pmatrix} = \begin{pmatrix} I & A \\ A - A & I - I \end{pmatrix} = \begin{pmatrix} I & A \\ 0 & 0 \end{pmatrix},$$

where the first equality holds since $A^2 = I$ by Proposition 4.3. The matrix on the right-hand size clearly has rank K (since $I \stackrel{\text{def}}{=} I_K$ is the identity matrix of order K). This implies $\text{rank}_{\mathbb{F}_2}(B) = K$, as required. ■

4.3 The existence of the matrix A

Finally, we prove Proposition 4.3, restated next.

Proposition 4.3 (restated). *There exists a symmetric matrix $A \in \mathbb{F}_2^{\mathcal{X}\times\mathcal{Y}}$ such that*

$$\log \mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A) \geq \Omega(NS_{\mathbb{F}_2}(\phi) \cdot t) - \log q, \tag{16}$$

and such that $A^2 = I$.

To this end, we use the lifting theorem of [dRMN+19] (Theorem 2.27). By applying that theorem to the lifted search problem $S_\phi \diamond \text{eq}$, we obtain a matrix $A \in \mathbb{F}_2^{\Lambda^\ell \times \Lambda^\ell}$ that satisfies the lower bound of Equation (16) for $S_\phi \diamond \text{eq}$. Our goal is to prove that A satisfies this lower bound for $mKW_{\mathcal{X}\times\mathcal{Y}}$, and to prove that A is symmetric and satisfies $A^2 = I$.

We start by tackling the following minor technical issue: By its definition, the rows and columns of A are indexed by Λ^ℓ, whereas in order to lower bound $\mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}})$, we need a matrix whose rows and columns are indexed by $\mathcal{X}$ and $\mathcal{Y}$ respectively. To this end, recall that $\mathcal{X} \stackrel{\text{def}}{=} R_A(\Lambda^\ell)$ and $\mathcal{Y} \stackrel{\text{def}}{=} R_B(\Lambda^\ell)$, where R_A and R_B are the injective functions of the reduction from $S_\phi \diamond \text{eq}$ to mKW_g. Thus, R_A and R_B are bijections from Λ^ℓ to $\mathcal{X}$ and $\mathcal{Y}$ respectively. It follows that we can view the rows and columns of A as being indexed by $\mathcal{X}$ and $\mathcal{Y}$ respectively by using R_A and R_B to translate the indices.

Now, in order to prove that A gives the desired lower bound on $\mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}})$, we show that every monochromatic rectangle $T \subseteq \mathcal{X} \times \mathcal{Y}$ of $mKW_{\mathcal{X}\times\mathcal{Y}}$ is also a monochromatic rectangle of $S_\phi \diamond \text{eq}$ (when interpreted as a rectangle in $\Lambda^\ell \times \Lambda^\ell$ via R_A^{-1}, R_B^{-1}). Let $T \subseteq \mathcal{X} \times \mathcal{Y}$ be a monochromatic rectangle of $mKW_{\mathcal{X}\times\mathcal{Y}}$, and suppose that it is labeled with a solution $j \in [n]$. Let $o \stackrel{\text{def}}{=} R_{\text{out}}(j)$, where R_{out} is the function of the reduction from $S_\phi \diamond \text{eq}$ to mKW_g. Then, by the definition of R_{out}, for every $(x, y) \in T$ it holds that o is a solution for $S_\phi \diamond \text{eq}$ on $\left(R_A^{-1}(x), R_B^{-1}(y)\right)$. Thus, T can be

viewed as an o-monochromatic rectangle of $S_\phi \diamond \mathrm{eq}$. It follows that

$$\begin{aligned}
\log \mu_{\mathbb{F}_2}(mKW_{\mathcal{X}\times\mathcal{Y}}, A) &\stackrel{\text{def}}{=} \log \mathrm{rank}_{\mathbb{F}_2}(A) - \max_{\substack{\text{monochromatic rectangle} \\ T \text{ of } mKW_{\mathcal{X}\times\mathcal{Y}}}} \log \mathrm{rank}_{\mathbb{F}_2}(A|_T) \\
&\geq \log \mathrm{rank}_{\mathbb{F}_2}(A) - \max_{\substack{\text{monochromatic rectangle} \\ T \text{ of } S_\phi \diamond \mathrm{eq}}} \log \mathrm{rank}_{\mathbb{F}_2}(A|_T) \\
&\stackrel{\text{def}}{=} \log \mu_{\mathbb{F}_2}(S_\phi \diamond \mathrm{eq}, A) \\
&\geq d \cdot \log\left(\frac{d \cdot 2^t}{e \cdot \ell}\right) - \frac{6 \cdot \ell \cdot \log e}{2^t} - \log q,
\end{aligned}$$

as required.

It remains to prove that A is symmetric and satisfies $A^2 = I$. To this end, we take a closer look at how the matrix A is constructed. The proof of [dRMN+19] (following [She11, RPRC16, PR17]) chooses the matrix A to be a *pattern matrix*, that is: there exists a multi-linear polynomial $p : \mathbb{F}_2^\ell \to \mathbb{F}_2$ of degree ℓ such that, for every two inputs $x, y \in \Lambda^\ell$,

$$A_{x,y} \stackrel{\text{def}}{=} p(\mathrm{eq}(x_1, y_1), \ldots, \mathrm{eq}(x_\ell, y_\ell)). \tag{17}$$

This immediately implies that A is symmetric, since it is easy to see that the right-hand side of Equation (17) remains the same if we swap x and y. In order to show that $A^2 = I$, we write A as a sum of Kronecker products: For every set $T \subseteq [\ell]$, we denote by $\hat{p}(T)$ the coefficient of p at the monomial $\prod_{i\in T} x_i$. Let $\mathbb{1}_{|\Lambda|}$ denote the all-ones matrix of order $|\Lambda| \times |\Lambda|$, and for every $T \subseteq [\ell]$ and $i \in [\ell]$, denote the matrix

$$Q_{i,T} = \begin{cases} I_{|\Lambda|} & \text{if } i \in T \\ \mathbb{1}_{|\Lambda|} & \text{if } i \notin T. \end{cases}$$

Robere [Rob18] showed that A can be written as follows:

$$A = \sum_{T\subseteq[\ell]} \hat{p}(T) \cdot Q_{1,T} \otimes \cdots \otimes Q_{\ell,T}.$$

Essentially, the latter identity holds since for every $i \in T$, the value of $I_{|\Lambda|}$ at the entry x_i, y_i is $\mathrm{eq}(x_i, y_i)$, whereas for every $i \notin T$, multiplying by $\mathbb{1}_{|\Lambda|}$ does not change the value of the product. It follows that

$$\begin{aligned}
A^2 &= \left(\sum_{T\subseteq[\ell]} \hat{p}(T) \cdot Q_{1,T} \otimes \cdots \otimes Q_{\ell,T}\right)^2 \\
&= \left(\sum_{T\subseteq[\ell]:\hat{p}(T)=1} Q_{1,T} \otimes \cdots \otimes Q_{\ell,T}\right)^2 && \text{(we are working over } \mathbb{F}_2) \\
&= \sum_{T,T'\subseteq[\ell]:\hat{p}(T)=\hat{p}(T')=1} (Q_{1,T} \otimes \cdots \otimes Q_{\ell,T}) \cdot \left(Q_{1,T'} \otimes \cdots \otimes Q_{\ell,T'}\right) \\
&= \sum_{T,T'\subseteq[\ell]:\hat{p}(T)=\hat{p}(T')=1} \left(Q_{1,T} \cdot Q_{1,T'}\right) \otimes \cdots \otimes \left(Q_{\ell,T} \cdot Q_{\ell,T'}\right) && \text{(Fact 2.41).}
\end{aligned}$$

Next, observe that for every two distinct sets $T, T' \subseteq [\ell]$, the last sum contains two terms:

$$(Q_{1,T} \cdot Q_{1,T'}) \otimes \cdots \otimes (Q_{\ell,T} \cdot Q_{\ell,T'}) \text{ and } (Q_{1,T'} \cdot Q_{1,T}) \otimes \cdots \otimes (Q_{\ell,T'} \cdot Q_{\ell,T})$$

We now claim that those two terms are equal and therefore cancel each other. To this end, we claim that for every $i \in [\ell]$ the matrices $Q_{i,T}$ and $Q_{i,T'}$ commute: the reason is that either one of those matrices is $I_{|\Lambda|}$ (and then they clearly commute) or both matrices are equal to $\mathbb{1}_{|\Lambda|}$ (and then again they clearly commute). It follows that for every two distinct sets $T, T' \subseteq [\ell]$, the above terms are equal and thus cancel each other. Hence, we remain only with the terms that correspond to $T = T'$, so

$$A^2 = \sum_{T \subseteq [\ell] : \hat{p}(T)=1} Q_{1,T}^2 \otimes \cdots \otimes Q_{\ell,T}^2.$$

Finally, observe that $|\Lambda| = 2^t$ is even, and thus $\left(\mathbb{1}_{|\Lambda|}\right)^2$ is the all-zeroes matrix. Hence, every term in the above sum in which one of the matrices $Q_{i,T}$ is equal to $\mathbb{1}_{|\Lambda|}$ zeroes out. The only term that remains is therefore the term that corresponds to $T = [\ell]$. Furthermore, the degree of p is ℓ, and therefore $\hat{p}([\ell]) = 1$. It follows that

$$\begin{aligned} A^2 &= Q_{1,[\ell]}^2 \otimes \cdots Q_{\ell,[\ell]}^2 \\ &= \underbrace{I_{|\Lambda|}^2 \otimes \cdots \otimes I_{|\Lambda|}^2}_{\ell \text{ times}} \\ &= \underbrace{I_{|\Lambda|} \otimes \cdots \otimes I_{|\Lambda|}}_{\ell \text{ times}} \\ &= I_{\left|\Lambda^\ell\right|} \stackrel{\text{def}}{=} I. \end{aligned}$$

Hence, we have shown that A is symmetric and that $A^2 = I$, as required.

5 A generalized lifting theorem

In this section, we prove our generalization of the lifting theorem of [CFK+19] (Theorem 2.26). The latter theorem says that if a search problem $S \subseteq \{0,1\}^\ell \times \mathcal{O}$ is lifted with an appropriate gadget $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$, then $\mathsf{CC}(S \diamond \mathrm{gd}) = \Omega(\mathsf{Q}(S) \cdot t)$. Essentially, our theorem says that this lower bound remains intact even if we restrict the inputs of $S \diamond \mathrm{gd}$ to a rectangle $\mathcal{X} \times \mathcal{Y}$, as long as the relative average degree of any coordinate in $\mathcal{X}$ and $\mathcal{Y}$ is at least $\frac{1}{\mathrm{poly}(\ell)}$. Formally,

Theorem 5.1. *For every $\eta > 0$ and $d \in \mathbb{N}$ there exist $c \in \mathbb{N}$ and $\kappa > 0$ such that the following holds: Let S be a search problem that takes inputs from $\{0,1\}^\ell$, and let $\mathrm{gd} : \{0,1\}^t \times \{0,1\}^t \to \{0,1\}$ be an arbitrary function such that $\mathrm{disc}(\mathrm{gd}) \leq 2^{-\eta \cdot t}$ and such that $t \geq c \cdot \log \ell$. Let $\mathcal{X}, \mathcal{Y} \subseteq \left(\{0,1\}^t\right)^\ell$ such that for every $I \subseteq [\ell]$ both $\mathrm{rAvgDeg}_I(\mathcal{X})$ and $\mathrm{rAvgDeg}_I(\mathcal{Y})$ are at least $1/(d \cdot \ell^d)^{|I|}$. Then the communication complexity of solving $S \diamond \mathrm{gd}$ on inputs from $\mathcal{X} \times \mathcal{Y}$ is at least $\kappa \cdot \mathsf{Q}(S) \cdot t$.*

We believe that it is possible to prove similar generalizations of the lifting theorems of [RM99, GPW15, CKLM17, WYY17], which in turn would extend our monotone composition theorem to work with those theorems.

Let η, d, S, gd be as in the theorem. We will choose the constants c and κ at the end of the proof to be sufficiently large such that the various inequalities hold. For convenience, for every set of coordinates $I \subseteq [\ell]$ we denote by gd^I the function that takes $|I|$ independent inputs to gd and computes gd on all of them. In particular, we denote $G \stackrel{\text{def}}{=} \mathrm{gd}^{[\ell]}$, so we can write $S \diamond \mathrm{gd} = S \circ G$.

As in previous works, we prove the theorem by showing that every protocol that solves $S \diamond \mathrm{gd}$ using C bits can be transformed into a decision tree that solves S using $\frac{C}{O(t)}$ queries, which implies the desired result. Let Π be a protocol that solves $S \diamond \mathrm{gd}$ using C bits of communication. We construct a decision tree T that solves S using $\frac{C}{O(t)}$ queries. The rest of this section is organized as follows: In Section 5.1, we provide an overview of the proof. In Section 5.2, we state the background that we need from the lifting literature. Then, in Section 5.3, we describe the decision tree T and prove its correctness. Finally, in Section 5.4, we upper bound the query complexity of T.

5.1 Proof overview

We start with an overview of the proof of [CFK+19]. Their proof works by a simulation argument: Given an input $z \in \{0,1\}^\ell$, the tree T constructs a full transcript π of Π, such that the rectangle $\mathcal{X}_\pi \times \mathcal{Y}_\pi$ contains an input $(x,y) \in G^{-1}(z)$, and returns the output of π. Clearly, the transcript π must output the correct solution for z, since $S \circ G(x,y) = S(z)$.

The tree T constructs the transcript π by simulating Π message-by-message. Throughout the simulation, the tree T maintains random variables $\boldsymbol{x}, \boldsymbol{y}$ be that are distributed over $\mathcal{X}_\pi \times \mathcal{Y}_\pi$. Let $\boldsymbol{z} \stackrel{\text{def}}{=} G(\boldsymbol{x}, \boldsymbol{y})$. The goal of the tree T is to make sure that when the simulation of Π halts, the input z is in the support of $\boldsymbol{z}$.

When the simulation starts, we set $\boldsymbol{x}, \boldsymbol{y}$ to be uniformly distributed over all inputs, and therefore $\boldsymbol{z}$ is uniformly distributed over $\{0,1\}^\ell$. As the simulation progresses, the transcript π reveals more and more information about $\boldsymbol{x}, \boldsymbol{y}$, until at some point there are coordinates $I \subseteq [\ell]$ about which a lot of information has been revealed. At this point, there is a danger that the value of $\boldsymbol{z}_I$ might get fixed to a value different than z_I. Before this happens, the tree T queries z_I, and conditions the random variables $\boldsymbol{x}, \boldsymbol{y}$ on the event $\boldsymbol{z}_I = z_I$. This conditioning is repeated whenever a significant amount of information is revealed about some coordinates, where "a significant amount" is $\alpha \cdot t$ bits of information per coordinate in I for some constant $\alpha > 0$.

Eventually, the simulation halts. At this point, we know that $\boldsymbol{z}$ is consistent with z in all its fixed coordinates. Moreover, we can show that since only a little information has been revealed about all the other coordinates, the value of $\boldsymbol{z}$ in the rest of the coordinates is uniformly distributed. Hence, z must be in the support of $\boldsymbol{z}$, as required.

It remains to upper bound the query complexity of T. As noted above, the tree T queries z once for each coordinate on which the transcript revealed $\alpha \cdot t$ bits of information. On the other hand, we know that the transcript π reveals at most C bits of information about $\boldsymbol{x}, \boldsymbol{y}$, since this is the communication complexity of Π. Thus, there are at most $\frac{C}{\alpha \cdot t}$ coordinates about which π reveals $\alpha \cdot t$ bits of information, so the query complexity of T is $O(\frac{C}{t})$, as required.

We now give some more details on how the query complexity is bounded, since we will need those details shortly. We bound the query complexity of T using a potential argument. Let U be the set of unfixed coordinates. Our potential function is the sum $H_\infty(\boldsymbol{x}_U) + H_\infty(\boldsymbol{y}_U)$. At the beginning of the simulation, $\boldsymbol{x}, \boldsymbol{y}$ are uniformly distributed over all inputs and $U = [\ell]$, so the potential is $2 \cdot t \cdot \ell$. After C bits were transmitted and q queries have been made, it is possible to show that the potential is decreased by at most $C + (2-\alpha) \cdot t \cdot q$. On the other hand, the potential is always upper bounded by $2 \cdot t \cdot |U|$, and since $|U| = \ell - q$ it follows that

$$2 \cdot t \cdot \ell - C - (2-\alpha) \cdot t \cdot q \leq 2 \cdot t \cdot |U| = 2 \cdot t \cdot (\ell - q). \tag{18}$$

from which we obtain the bound $q = O(C/t)$ after rearranging.

Our contribution. Our proof follows a similar outline, but at the beginning of the simulation, we set $\boldsymbol{x}, \boldsymbol{y}$ to be uniformly distributed over $\mathcal{X}$ and $\mathcal{Y}$ respectively. This difference results in two

issues. The first issue is that if some coordinate i of $\boldsymbol{x}, \boldsymbol{y}$ starts with relatively low min-entropy, then there is a danger that $\boldsymbol{z}_i$ will be fixed too early. Fortunately, such a situation can never happen since we assumed that $\mathcal{X}, \mathcal{Y}$ have high average degrees, which lower bounds the min-entropy (by Fact 2.39).

The second issue is that the foregoing potential argument becomes slightly more complicated. Specifically, the initial potential is now $\log|\mathcal{X}| + \log|\mathcal{Y}|$ rather than $2 \cdot t \cdot \ell$, and the upper bound on the potential is now $\log|\mathcal{X}_U| + \log|\mathcal{Y}_U|$ rather than $2 \cdot t \cdot |U|$. Thus, Equation (18) is replaced with the equation

$$\log|\mathcal{X}| + \log|\mathcal{Y}| - C - (2-\alpha) \cdot t \cdot q \leq \log|\mathcal{X}_U| + \log|\mathcal{Y}_U|\,.$$

In order to derive a bound on q from the latter equation, we need to lower bound the difference

$$(\log|\mathcal{X}| + \log|\mathcal{Y}|) - (\log|\mathcal{X}_U| + \log|\mathcal{Y}_U|)\,.$$

To this end, we observe that

$$\log(|\mathcal{X}|) - \log(|\mathcal{X}_U|) = \log\left(\frac{|\mathcal{X}|}{|\mathcal{X}_U|}\right) = \log\left(\mathrm{AvgDeg}_{[\ell]-U}(\mathcal{X})\right),$$

and a similar equality holds for $\mathcal{Y}$. We now get the desired lower bound by using our assumed bound on the average degrees of $\mathcal{X}$ and $\mathcal{Y}$.

5.2 Lifting machinery

As explained above, a key part of the simulation is keeping track of the coordinates on which the protocol did *not* transmit much information. We model a string about which not much information has been revealed using the following notion of a *dense* random variable (not to be confused with the notion of *density* from Section 2.10).

Definition 5.2 ([GLM+16])**.** Let $n \in \mathbb{N}$ and $\delta > 0$, and let $\boldsymbol{x}$ be a random variable taking values in Λ^n. We say that $\boldsymbol{x}$ is δ-dense if for every set of coordinates $I \subseteq [n]$ it holds that $H_\infty(\boldsymbol{x}_I) \geq \delta \cdot t \cdot |I|$.

We will keep track of which coordinates of $\boldsymbol{z}$ have been fixed and which are still free using the standard notion of restriction.

Definition 5.3. A *restriction* ρ is a string in $\{0,1,*\}^\ell$. We say that a coordinate $i \in [\ell]$ is *free* in ρ if $\rho_i = *$, and otherwise we say that i is *fixed*. Given a restriction $\rho \in \{0,1,*\}^n$, we denote by $\mathrm{free}(\rho)$ and $\mathrm{fix}(\rho)$ the sets of free and fixed coordinates of ρ respectively. We say that a string $z \in \{0,1\}^\ell$ is *consistent* with ρ if $z_{\mathrm{fix}(\rho)} = \rho_{\mathrm{fix}(\rho)}$.

Our decision tree will maintain the following invariant, which captures the idea that $\boldsymbol{z} = G(\boldsymbol{x}, \boldsymbol{y})$ is fixed in some coordinates, and not too much information has been revealed on the other coordinates.

Definition 5.4 ([GLM+16, GPW17])**.** Let $\rho \in \{0,1,*\}^\ell$ be a restriction, let $\tau > 0$, and let $\boldsymbol{x}, \boldsymbol{y}$ be independent random variables taking values in Λ^ℓ. We say that $\boldsymbol{x}$ and $\boldsymbol{y}$ are (ρ, τ)-*structured* if there exist $\delta_x, \delta_y > 0$ such that $\boldsymbol{x}_{\mathrm{free}(\rho)}$ and $\boldsymbol{y}_{\mathrm{free}(\rho)}$ are δ_x-dense and δ_y-dense respectively, $\delta_x + \delta_y \geq \tau$, and

$$\mathrm{gd}^{\mathrm{fix}(\rho)}\left(\boldsymbol{x}_{\mathrm{fix}(\rho)}, \boldsymbol{y}_{\mathrm{fix}(\rho)}\right) = \rho_{\mathrm{fix}(\rho)}.$$

The following results use the assumption that gd has input length $t \geq c \cdot \log \ell$ and discrepancy at least $2^{-\eta \cdot t}$. A key property of structured variables $\boldsymbol{x}, \boldsymbol{y}$ is that in all the free coordinates, the random variable $\boldsymbol{z}_{\text{free}(\rho)} = G(\boldsymbol{x}, \boldsymbol{y})$ has full support. This property is formalized by the following result.

Proposition 5.5 (special case of [CFK+19, Prop 3.10])**.** *There exists a universal constant h such that the following holds: Let $\boldsymbol{x}, \boldsymbol{y}$ be random variables that are (ρ, τ)-structured for $\tau > 2 + \frac{h}{c} - \eta$. Then, the support of the random variable $\text{gd}^{\text{free}(\rho)}(\boldsymbol{x}_{\text{free}(\rho)}, \boldsymbol{y}_{\text{free}(\rho)})$ is $\{0,1\}^{\text{free}(\rho)}$.*

Whenever the protocol transmits so much information that $\boldsymbol{x}$ or $\boldsymbol{y}$ cease to be dense, we wish to fix some coordinates in order to restore their density. This is done by the following folklore fact.

Proposition 5.6 (see, e.g., [GPW17])**.** *Let $n \in \mathbb{N}$, let $\delta > 0$, and let $\boldsymbol{x}$ be a random variable taking values in Λ^n. Let $I \subseteq [n]$ be a maximal subset of coordinates such that $H_\infty(\boldsymbol{x}_I) < \delta \cdot t \cdot |I|$, and let $x_I \in \Lambda^I$ be a value such that $\Pr[\boldsymbol{x}_I = x_I] > 2^{-\delta \cdot t \cdot |I|}$. Then, the random variable $\boldsymbol{x}_{[n]-I} | \boldsymbol{x}_I = x_I$ is δ-dense.*

Proposition 5.6 allows us to restore the density of $\boldsymbol{x}$ by fixing some set of coordinates I. In order to maintain the invariant that $\boldsymbol{x}$ and $\boldsymbol{y}$ are structured, we also need to ensure that $\text{gd}^I(x_I, \boldsymbol{y}_I) = \rho_I$. To this end, we condition $\boldsymbol{y}$ on the latter event. However, this conditioning reveals information about $\boldsymbol{y}$, which may have two harmful effects:

- **Leaking:** As discussed in Section 5.1, our analysis of the query complexity assumes that the transcript π reveals at most $O(C)$ bits of information. It is important not to reveal more information than that, or otherwise our query complexity may increase arbitrarily. On average, we expect that conditioning on the event $\text{gd}^I(x_I, \boldsymbol{y}_I) = \rho_I$ would reveal only $|I|$ bits of information, which is sufficiently small for our purposes. However, there could be values of x_I and ρ_I for which much more information is leaked. In this case, we say that the conditioning is *leaking*.
- **Sparsifying:** Even if the conditioning reveals only $|I|$ bits of information about $\boldsymbol{y}$, this could still ruin the density of $\boldsymbol{y}$ if the set I is large. In this case, we say that the conditioning is *sparsifying*.

We refer values of $\boldsymbol{x}$ that may lead to those effects as *dangerous*, and define them as follows.

Definition 5.7 ([CFK+19])**.** Let $n \in \mathbb{N}$ and let $\boldsymbol{y}$ be a random variable taking values from Λ^n. We say that a value $x \in \Lambda^n$ is *leaking* if there exists a set $I \subseteq [\ell]$ and an assignment $z_I \in \{0,1\}^I$ such that

$$\Pr\left[\text{gd}^I(x_I, \boldsymbol{y}_I) = z_I\right] < 2^{-|I|-1}.$$

Let $\delta, \varepsilon > 0$, and suppose that $\boldsymbol{y}$ is δ-dense. We say that a value $x \in \Lambda^n$ is *ε-sparsifying* if there exists a set $I \subseteq [n]$ and an assignment $z_I \in \{0,1\}^I$ such that the random variable

$$\boldsymbol{y}_{[n]-I} | \text{gd}^I(x_I, \boldsymbol{y}_I) = z_I$$

is not $(\delta - \varepsilon)$-dense. We say that a value $x \in \Lambda^n$ is *ε-dangerous* if it is either leaking or ε-sparsifying.

Chattopadhyay et. al. [CFK+19] deal with this issue by upper bounding the probability of dangerous values:

Lemma 5.8 (special case of [CFK+19, Lemma 3.9])**.** *There exists a universal constant h such that the following holds: Let $0 < \gamma, \varepsilon, \tau \leq 1$ be such that $\tau \geq 2 + \frac{h}{c \cdot \varepsilon} - \eta$ and $\varepsilon \geq \frac{4}{t}$, and let $\boldsymbol{x}, \boldsymbol{y}$ be (ρ, τ)-structured random variables. Then, the probability that $\boldsymbol{x}_{\text{free}(\rho)}$ takes a value that is ε-dangerous for $\boldsymbol{y}_{\text{free}(\rho)}$ is at most $\frac{1}{2}$.*

5.3 The construction of the decision tree T

Let h be the maximum among the universal constants of Proposition 5.5 and Lemma 5.8. Let $\varepsilon \stackrel{\text{def}}{=} \frac{2h}{c\cdot\eta}$, let $\delta \stackrel{\text{def}}{=} 1 - \frac{\eta}{4} + \frac{\varepsilon}{2}$, and let $\tau \stackrel{\text{def}}{=} 2\cdot\delta - \varepsilon$. The tree T constructs a transcript π by simulating the protocol Π round-by-round, each time adding a single message to π. Throughout the simulation, the tree maintains two independent random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ that are distributed over $\mathcal{X}_\pi$ and $\mathcal{Y}_\pi$ respectively. The tree will maintain the invariant that $\boldsymbol{x}$ and $\boldsymbol{y}$ are (ρ,τ)-structured, where ρ is a restriction that keeps track of the queries the tree has made to z so far. In fact, the tree will maintain a more specific invariant: whenever it is Alice's turn to speak, $\boldsymbol{x}_{\text{free}(\rho)}$ is $(\delta-\varepsilon)$-dense and $\boldsymbol{y}_{\text{free}(\rho)}$ is δ-dense, and whenever it is Bob's turn to speak, the roles of $\boldsymbol{x}$ and $\boldsymbol{y}$ are reversed.

When the tree T starts the simulation, the tree sets the transcript π to be the empty string, the restriction ρ to $\{*\}^\ell$, and the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ to be uniformly distributed over $\mathcal{X}$ and $\mathcal{Y}$ respectively. We show that at this point, $\boldsymbol{x}$ and $\boldsymbol{y}$ are both δ-dense, and thus satisfy the invariant. Let $I \subseteq [\ell]$ be a set of coordinates. We show that $H_\infty(\boldsymbol{x}_I) \geq \delta\cdot t\cdot|I|$, and the proof for $\boldsymbol{y}$ is similar. Recall that by Fact 2.39, the logarithm of average degree is a lower bound on min-entropy. Thus, our assumption on the relative average degrees of $\mathcal{X}$ implies that

$$\begin{aligned} H_\infty(\boldsymbol{x}_I) &\geq t\cdot|I| - \log\frac{1}{\text{rAvgDeg}_I(\mathcal{X})} \\ &\geq (t - d\log\ell - \log d)\cdot|I| \\ &= (1 - \frac{d\log\ell}{t} - \frac{\log d}{t})\cdot t\cdot|I| \\ &\geq (1 - \frac{d+\log d}{c})\cdot t\cdot|I| \qquad\qquad (t \geq c\cdot\log\ell). \end{aligned} \tag{19}$$

By choosing c to be sufficiently large, we can ensure that the last expression is at least $\delta\cdot t\cdot|I|$, as required. We now explain how T simulates a single round of the protocol while maintaining the invariant. Suppose that the invariant holds at the beginning of the current round, and assume without loss of generality that it is Alice's turn to speak. The tree T performs the following steps:

1. The tree conditions $\boldsymbol{x}_{\text{free}(\rho)}$ on not taking a value that is ε-dangerous for $\boldsymbol{y}_{\text{free}(\rho)}$
2. The tree T chooses an arbitrary message M of Alice with the following property: the probability of Alice sending M on input $\boldsymbol{x}$ is at least $2^{-|M|}$ (the existence of M will be justified soon). The tree adds M to the transcript π, and conditions $\boldsymbol{x}$ on the event of sending M.
3. Let $I \subseteq \text{free}(\rho)$ be a maximal set that violates the δ-density of $\boldsymbol{x}_{\text{free}(\rho)}$ (i.e., $H_\infty(\boldsymbol{x}_I) < \delta\cdot t\cdot|I|$), and let $x_I \in \Lambda^I$ be a value that satisfies $\Pr[\boldsymbol{x}_I = x_I] > 2^{-\delta\cdot t\cdot|I|}$. The tree conditions $\boldsymbol{x}$ on $\boldsymbol{x}_I = x_I$. By Proposition 5.6, the variable $\boldsymbol{x}_{\text{free}(\rho)-I}$ is now δ-dense.
4. The tree queries z_I, and sets $\rho_I = z_I$.
5. The tree conditions $\boldsymbol{y}$ on $\text{gd}^I(x_I, \boldsymbol{y}_I) = \rho_I$. Due to Step 1, the variable $\boldsymbol{x}_{\text{free}(\rho)}$ must take a value that is not ε-dangerous, and therefore $\boldsymbol{y}_{\text{free}(\rho)}$ is necessarily $(\delta-\varepsilon)$-dense.

After those steps take place, it becomes Bob's turn to speak, and indeed, $\boldsymbol{x}_{\text{free}(\rho)}$ and $\boldsymbol{y}_{\text{free}(\rho)}$ are δ-dense and $(\delta-\varepsilon)$-dense respectively. Thus, the invariant is maintained. In order for the foregoing steps to be well-defined, it remains to explain three points:

- First, we should explain why Step 1 conditions $\boldsymbol{x}$ on an event with a non-zero probability. To this end, we note that τ is larger than $2 + \frac{h'}{c\cdot\varepsilon} - \eta$ (see Equation (21) below for a detailed calculation). Hence, by Lemma 5.8, the variable $\boldsymbol{x}_{\text{free}(\rho)}$ has a non-zero probability to take a value that is not ε-dangerous for $\boldsymbol{y}_{\text{free}(\rho)}$.

- Second, we should explain why the message M in Step 2 exists. To see why, observe that the set of possible messages of Alice forms a prefix-free code — otherwise, Bob would not be able to tell when Alice finished speaking and his turn starts. Hence, by Fact 2.32, it follows that there exists a message M with probability at least $2^{-|M|}$.

- Third, we should explain why Step 5 conditions $\boldsymbol{y}$ on an event with a non-zero probability. To this end, recall that $\boldsymbol{x}$ must take a value that is not ε-dangerous for $\boldsymbol{y}$, and in particular, the value of $\boldsymbol{x}$ is necessarily not leaking. This means that that the string $\mathrm{gd}^I(x_I, \boldsymbol{y}_I)$ has non-zero probability to be equal to ρ_I.

Finally, when the protocol halts, the tree T outputs the solution of the transcript π. We claim that this solution is a correct solution for z. To this end, recall that since $\boldsymbol{x}$ and $\boldsymbol{y}$ are consistent with π, the transcript π outputs a solution for $S \diamond \mathrm{gd}$ that is correct for every pair (x, y) in the support of $(\boldsymbol{x}, \boldsymbol{y})$. Thus, it suffices to show that there exists some pair (x, y) in the support of $(\boldsymbol{x}, \boldsymbol{y})$ such that $G(x, y) = z$. In other words, it suffices to show that $\Pr[G(\boldsymbol{x}, \boldsymbol{y}) = z] > 0$.

Since $\boldsymbol{x}$ and $\boldsymbol{y}$ are (ρ, τ)-structured and ρ is consistent with z, it holds that $\mathrm{gd}^{\mathrm{fix}(\rho)}(\boldsymbol{x}_{\mathrm{fix}(\rho)}, \boldsymbol{y}_{\mathrm{fix}(\rho)}) = z_{\mathrm{fix}(\rho)}$ with probability 1. It remains to deal with the free coordinates of ρ. To this end, we note that τ is larger than $2 + \frac{h'}{c \cdot \varepsilon} - \eta$ (see Equation (21) below for a detailed calculation). Hence, Proposition 5.5 implies that $z_{\mathrm{free}(\rho)}$ is in the support of $\mathrm{gd}^{\mathrm{free}(\rho)}(\boldsymbol{x}_{\mathrm{free}(\rho)}, \boldsymbol{y}_{\mathrm{free}(\rho)})$. It follows that $\Pr[G(\boldsymbol{x}, \boldsymbol{y}) = z] > 0$, as required.

5.4 The query complexity of T

Let z be an arbitrary input for T, and let q be the number of queries that T makes on input z. We show that for some constant κ that depends only on η and d, it holds that $C \geq \kappa \cdot q \cdot t$, and this will conclude the proof of the lifting theorem. To this end, we prove that when the tree T halts,

$$H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)}) + H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)}) \geq \log|\mathcal{X}| + \log|\mathcal{Y}| - 3 \cdot C - (1 + \delta + \frac{1}{c}) \cdot t \cdot q. \tag{20}$$

We claim that Equation (20) implies the desired bound on C. To see why, observe that by Fact 2.29 it holds that

$$H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)}) + H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)}) \leq \log\left|\mathcal{X}_{\mathrm{free}(\rho)}\right| + \log\left|\mathcal{Y}_{\mathrm{free}(\rho)}\right|.$$

By combining the two bounds, it follows that

$$\begin{aligned}
3 \cdot C &\geq \log|\mathcal{X}| + \log|\mathcal{Y}| - \log\left|\mathcal{X}_{\mathrm{free}(\rho)}\right| - \log\left|\mathcal{Y}_{\mathrm{free}(\rho)}\right| - (1 + \delta + \frac{1}{c}) \cdot t \cdot q \\
&= \log\frac{|\mathcal{X}|}{\left|\mathcal{X}_{\mathrm{free}(\rho)}\right|} + \log\frac{|\mathcal{Y}|}{\left|\mathcal{Y}_{\mathrm{free}(\rho)}\right|} - (1 + \delta + \frac{1}{c}) \cdot t \cdot q \\
&= \log \mathrm{AvgDeg}_{\mathrm{fix}(\rho)}(\mathcal{X}) + \log \mathrm{AvgDeg}_{\mathrm{fix}(\rho)}(\mathcal{Y}) - (1 + \delta + \frac{1}{c}) \cdot t \cdot q
\end{aligned}$$

Next, by applying our assumed lower bound on the relative average degrees and noting that $q = |\mathrm{fix}(\rho)|$, we obtain that

$$\begin{aligned}
\log \mathrm{AvgDeg}_{\mathrm{fix}(\rho)}(\mathcal{X}) &= t \cdot |\mathrm{fix}(\rho)| - \log\frac{1}{\mathrm{rAvgDeg}_{\mathrm{fix}(\rho)}(\mathcal{X})} \\
&\geq t \cdot |\mathrm{fix}(\rho)| - (d \cdot \log\ell + \log d) \cdot |\mathrm{fix}(\rho)| \\
&= (t - d \cdot \log\ell - \log d) \cdot q \qquad (q = |\mathrm{fix}(\rho)|)
\end{aligned}$$

and the same lower bound holds for $\log \mathrm{AvgDeg}_{\mathrm{fix}(\rho)}(\mathcal{Y})$. By combining the inequalities, it follows that

$$\begin{aligned}
3 \cdot C &\geq 2 \cdot (t - d \cdot \log \ell - \log d) \cdot q - (1 + \delta + \frac{3}{c}) \cdot t \cdot q \\
&\geq 2 \cdot (1 - \frac{d + \log d}{c}) \cdot t \cdot q - (1 + \delta + \frac{3}{c}) \cdot t \cdot q \qquad \text{(See Equation (19) for the calculation)} \\
&\geq (1 - \delta - \frac{2d + 2\log d + 3}{c}) \cdot t \cdot q \\
&= (\frac{\eta}{4} - \frac{h}{2 \cdot c \cdot \eta} - \frac{2d + 2\log d + 3}{c}) \cdot t \cdot q \qquad (\delta \stackrel{\text{def}}{=} 1 - \frac{\eta}{4} + \frac{h}{2 \cdot c \cdot \eta}).
\end{aligned}$$

We now choose $\kappa \stackrel{\text{def}}{=} \frac{\eta}{4} - \frac{h}{2 \cdot c \cdot \eta} - \frac{2d + 2\log d + 3}{c}$ and observe that we can make sure that $\kappa > 0$ by choosing c to be sufficiently large.

It remains to prove Equation (20). Observe that when the tree starts the simulation, $\mathrm{free}(\rho) = [\ell]$ and $\boldsymbol{x}, \boldsymbol{y}$ are uniformly distributed over $\mathcal{X}, \mathcal{Y}$ respectively, and hence

$$H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)}) + H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)}) = \log |\mathcal{X}| + \log |\mathcal{Y}| .$$

We will show that in every round of the simulation, the sum $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)}) + H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)})$ decreases by at most $3 \cdot |M| + (1 + \delta + \frac{1}{c}) \cdot t \cdot |I|$, where M is the message that the protocol sends and I is the set of queries that the tree makes at that round. Since the sum of the lengths of all the messages M is at most C, and the sum of the sizes of all sets I is q, this will imply Equation (20).

Fix a round of the simulation, and assume without loss of generality that the message is sent by Alice. We analyze the effect of each of the steps on $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)}) + H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)})$ individually:

- In Step 1, the tree conditions $\boldsymbol{x}_{\mathrm{free}(\rho)}$ on taking dangerous values that are not ε-dangerous for $\boldsymbol{y}_{\mathrm{free}(\rho)}$. We show that this step decreases $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)})$ by at most one bit. Recall that at this point $\boldsymbol{x}$ and $\boldsymbol{y}$ are (ρ, τ)-structured, where

$$\begin{aligned}
\tau &\stackrel{\text{def}}{=} 2 \cdot \delta - \varepsilon \qquad (21) \\
&= 2 \cdot \left(1 - \frac{\eta}{4} + \frac{\varepsilon}{2}\right) - \varepsilon \qquad \text{(by definition of } \delta) \\
&= 2 - \frac{\eta}{2} \\
&= 2 + \frac{\eta}{2} - \eta \\
&= 2 + \frac{h}{c \cdot \varepsilon} - \eta \qquad (\text{since } \varepsilon \stackrel{\text{def}}{=} \frac{2h}{c \cdot \eta}).
\end{aligned}$$

  Therefore, by applying Lemma 5.8, it follows that the probability that $\boldsymbol{x}_{\mathrm{free}(\rho)}$ is ε-dangerous is at most $\frac{1}{2}$. By Fact 2.30, conditioning on that event decreases $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)})$ by at most one bit.

- In Step 2, the tree conditions $\boldsymbol{x}$ on the event of sending the message M, which has probability at least $2^{-|M|}$. By Fact 2.30, this decreases $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)})$ by at most $|M|$ bits.

- In Step 3, the tree conditions on $\boldsymbol{x}$ on the event $\boldsymbol{x}_I = x_I$, which has probability greater than $2^{-\delta \cdot t \cdot |I|}$. By Fact 2.30, this decreases $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)})$ by at most $\delta \cdot t \cdot |I|$ bits.

- In Step 4, the tree removes I from $\mathrm{free}(\rho)$. By Fact 2.31, this removal decreases $H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)})$ by at most $t \cdot |I|$ bits. Moreover, this removal does not affect $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)})$, since at this point $\boldsymbol{x}_I$ is fixed.

- Finally, in Step 5, the tree conditions $\boldsymbol{y}$ on the event $\mathrm{gd}^I(x_I, \boldsymbol{y}_I) = \rho_I$. Due to Step 1, the value x_I is not dangerous and hence not leaking, so the latter event has probability at least $2^{-|I|-1}$. It follows that this conditioning decreasing $H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)})$ by at most $|I|+1$ bits.

Summing up, the round decreases $H_\infty(\boldsymbol{x}_{\mathrm{free}(\rho)}) + H_\infty(\boldsymbol{y}_{\mathrm{free}(\rho)})$ by at most

$$\begin{aligned}
&|M| + 1 + \delta \cdot t \cdot |I| + t \cdot |I| + |I| + 1 \\
=&|M| + (1 + \delta + \frac{1}{t}) \cdot t \cdot |I| + 2 \\
\leq&3 \cdot |M| + (1 + \delta + \frac{1}{t}) \cdot t \cdot |I| \\
\leq&3 \cdot |M| + (1 + \delta + \frac{1}{c}) \cdot t \cdot |I| \,,
\end{aligned}$$

as required.

6 Open questions

An obvious question that arises from this work is whether we can strengthen our semi-monotone composition theorem (Theorem 1.5) to work for every non-constant outer function f. As a starting point, can we prove such a semi-monotone composition theorem that holds when the inner function g is the s-t-connectivity function? We note that proving such a result would likely require new ideas, since our techniques seem to be insufficient:

- On the one hand, we cannot prove such a result along the lines of our monotone composition theorem, since in the semi-monotone setting we cannot assume that the protocol outputs an entry (i, j) for which $a_i \neq b_i$ (as in the observation of [KRW95] in the monotone case).
- On the other hand, we cannot prove such a result along the lines of our semi-monotone composition theorem, since the Razborov rank measure cannot prove interesting lower bounds for non-monotone KW relations [Raz92]. In particular, we would not be able to analyze the complexity of a non-monotone outer relation KW_f using this technique.

Another interesting question is whether we can strengthen our monotone composition theorem (Theorem 3.1) even further: Although this theorem holds for many choices of the inner functions g, there are still a few "classic" functions that it does not cover — most notably the matching function [RW92]. Can we prove a monotone composition theorem where f can be any non-constant monotone function, and g is the matching function?

Finally, recall that in the long run, our goal is to prove the KRW conjecture for the composition $KW_f \diamond MUX$ (for every f), since this would imply that $\mathbf{P} \not\subseteq \mathbf{NC}^1$. To this end, it seems reasonable to try to prove first the monotone and semi-monotone versions of this conjecture. The monotone version might be within reach (see [Mei19] for the statement of this conjecture). Can we prove it?

A Composition theorems for classic functions

In this appendix, we show that our composition theorems can be applied to three classic functions, namely: s-t-connectivity [KW90], clique [GH92, RW92], and generation [RM99]. Recall that if ϕ is a CNF contradiction, we denote by S_ϕ its corresponding search problem. We prove our results

by showing that for each of the above above functions, there is an injective reduction from the lifted search problem $S_\phi \diamond \mathrm{gd}$ to mKW_g for some appropriate formula ϕ and gadget gd. Specifically, for our monotone composition theorem we will choose the gadget gd to be inner product mod 2 function ip. For our semi-monotone composition theorem we will choose the gadget to be the equality function eq. In both cases, we will denote the input length of the gadget by t.

Preliminaries

Following [GP18, Oli15, Rob18], we will construct our reductions from $S_\phi \diamond \mathrm{gd}$ to mKW_g in two steps: first, we will reduce $S_\phi \diamond \mathrm{gd}$ to to the monotone KW relation $mKW_{\textsc{CspSat}}$ for a certain constraint satisfaction problem CspSat, and then we will reduce the latter relation to mKW_g. We now define the constraints satisfaction problem and the related notions.

Definition A.1. Let $H = (L \cup R, E)$ be a bipartite graph, and let Λ be a finite alphabet. For every vertex $r \in R$, we denote by $N(r) \subseteq L$ the set of neighbors of r. The constraint satisfaction problem $\textsc{CspSat}_{H,\Lambda}$ is the following decision problem: The input consists of a set of predicates $P_r : \Lambda^{N(r)} \to \{0,1\}$ for every $r \in R$. The answer on an input is "yes" if and only if there exists an assignment $\alpha : L \to \Lambda$ that satisfies all the predicates.

Definition A.2. Let ϕ be a CNF formula. The *graph of* ϕ, denoted $\mathrm{graph}(\phi)$, is the bipartite graph whose left and right vertices are the variables and clauses of ϕ respectively, and whose edges connect each clause with its variables.

We reduce $S_\phi \diamond \mathrm{gd}$ to CspSat using the following generic technique, due to [RM99, GP18, Oli15] (see also [Rob18, Sec. 6.1]). We note that the "moreover" part in the following theorem is implicit in those works, and that its condition is is satisfied by the gadgets that we use.

Theorem A.3. *For every CNF contradiction ϕ and gadget function $\mathrm{gd} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, the lifted search problem $S_\phi \diamond \mathrm{gd}$ reduces to the monotone KW relation of $\textsc{CspSat}_{\mathrm{graph}(\phi),\mathcal{X}}$. Moreover, the reduction is injective if for every $y \in \mathcal{Y}$, the function $\mathrm{gd}(\cdot, y) : \mathcal{X} \to \{0,1\}$ is non-constant and determines y.*

In order to reduce $mKW_{\textsc{CspSat}}$ to mKW_g, we reduce the function CspSat to g using the following special type of reduction.

Definition A.4. We say that a function $\rho : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$ is a *monotone projection* if for every $j \in [n_2]$, it either holds that the j-th output is a constant (i.e., always 0 or always 1), or there exists an input coordinate $i \in [n_1]$ such that for every $x \in \{0,1\}^{n_1}$ it holds that $\rho(x)_j = x_i$. Given two monotone functions $g_1 : \{0,1\}^{n_1} \to \{0,1\}$ and $g_2 : \{0,1\}^{n_2} \to \{0,1\}$, we say that there is *monotone projection* from g_1 to g_2 if $g_1 = g_2 \circ \rho$ for some monotone projection $\rho : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$.

It is not hard to see that if there is a monotone projection from g_1 to g_2, then there is an injective reduction from mKW_{g_1} to mKW_{g_2}. Finally, we will use the following fact to lower bound the query complexity of search problems.

Fact A.5 (see, e.g., [dRMN+19, Appx. C])**.** *Let ϕ be a CNF contradiction. Then $\mathsf{Q}(S_\phi) \geq NS_{\mathbb{F}_2}(\phi)$.*

The s-t-connectivity function

The s-t-connectivity function $\textsc{stConn}_n$ takes as input the adjacency matrix of a directed graph over n vertices with two distinguished vertices s, t, and says whether s and t are connected in the

graph. Karchmer and Wigderson [KW90] proved that $\mathsf{CC}(mKW_{\text{stConn}_n}) = \Theta(\log^2 n)$ for the case of undirected graphs, and alternative proofs were given by [GS91, Pot17, Rob18] for the case of directed graphs.

Below, we apply our main results to derive composition theorems with the inner function being stConn_n. Following [Rob18], we do this using the induction principle of [BP98], which is the CNF contradiction defined as follows:

$$\text{Ind}_\ell(z_1, \ldots, z_\ell) \stackrel{\text{def}}{=} z_1 \wedge (\neg z_1 \vee z_2) \wedge (\neg z_2 \vee z_3) \wedge \ldots \wedge (\neg z_{\ell-1} \vee z_\ell) \wedge \neg z_\ell.$$

Buss and Pitassi [BP98] showed that $NS_{\mathbb{F}_2}(\text{Ind}_\ell) = \Theta(\log \ell)$. We now reduce $S_{\text{Ind}_\ell} \diamond \text{gd}$ to mKW_{stConn_n} by constructing a monotone projection from $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ to stConn_n.

Proposition A.6. *For every $\ell \in \mathbb{N}$ and every finite set Λ, there is a monotone projection from $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ to stConn_n for $n \stackrel{\text{def}}{=} \ell \cdot |\Lambda| + 2$.*

Proof. We construct a projection that maps an input of $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ to an input of stConn_n. The input of stConn_n will be a layered graph G that has $\ell + 2$ layers. The first layer will contains only the distinguished vertex s, and the last layers will contain only the distinguished vertex t. Each of the ℓ middle layers will consist of $|\Lambda|$ vertices, which we label with the elements of Λ.

The edges of G are determined by the input of $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ as follows. Recall that an input to $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ consists of the following predicates: a predicate $P_{z_1} : \Lambda \to \{0,1\}$, predicates of the form $P_{\neg z_i \vee z_{i+1}} : \Lambda^2 \to \{0,1\}$ for every $i \in [\ell - 1]$, and a predicate $P_{\neg z_\ell} : \Lambda \to \{0,1\}$. Now,

- For every vertex $v \in \Lambda$ of the second layer, we put an edge (s, v) in G if and only if $P_{z_1}(v) = 1$.
- For every vertex $v \in \Lambda$ of the second-to-last layer, we put an edge (v, t) in G if and only if $P_{\neg z_\ell}(v) = 1$.
- For every two middle layers i and $i + 1$, we put in G an edge between a vertex $u \in \Lambda$ of the layer i and a vertex $v \in \Lambda$ of the layer $i + 1$ if and only if $P_{\neg z_i \vee z_{i+1}}(u, v) = 1$.

It can be verified that this mapping from inputs of $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ to inputs of stConn_n is a monotone projection. To see that it maps "yes" inputs of $\text{CspSAT}_{\text{graph}(\text{Ind}_\ell),\Lambda}$ to "yes" instances of stConn_n and vice versa, observe that every satisfying assignment specifies a path from s to t in G and vice versa. ■

We now apply our main results to obtain a monotone and semi-monotone composition theorems with the inner function being $g = \text{stConn}_n$.

Theorem A.7. *For every non-constant monotone function $f : \{0,1\}^m \to \{0,1\}$, it holds that for almost all $n \in \mathbb{N}$,*

$$\log \mathsf{L}(mKW_f \diamond mKW_{\text{stConn}_n}) = \log \mathsf{L}(mKW_f) + \Omega(\log \mathsf{L}(mKW_{\text{stConn}_n}))$$
$$\mathsf{CC}(U_m \diamond mKW_{\text{stConn}_n}) \geq m + \Omega\left(\mathsf{CC}(mKW_{\text{stConn}_n})\right).$$

Proof. Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant monotone function, and let c be the maximum between 2 and the constant obtained from our monotone composition theorem for $\eta = \frac{1}{2}$ (since the discrepancy of ip is $2^{-\frac{1}{2}t}$). Let $\ell \in \mathbb{N}$ be such that $\ell \geq m$, and let $t \stackrel{\text{def}}{=} \lceil c \cdot \log(m \cdot \ell) \rceil$. We show that the theorem holds for $n \stackrel{\text{def}}{=} \ell \cdot 2^t + 2$ and this will imply the theorem for almost all $n \in \mathbb{N}$ by padding.

By combining Proposition A.6 with Theorem A.3, we obtain injective reductions from $S_{\mathrm{Ind}_\ell} \diamond \mathrm{ip}$ and $S_{\mathrm{Ind}_\ell} \diamond \mathrm{eq}$ to mKW_{STCONN_n}. By the aforementioned result of [BP98] it holds that $NS_{\mathbb{F}_2}(\mathrm{Ind}_\ell) = \Theta(\log \ell)$, and this implies that $\mathsf{Q}(\mathrm{Ind}_\ell) \geq \Omega(\log \ell)$ by Fact A.5. It now follows by our monotone composition theorem that

$$\begin{aligned}
\log \mathsf{L}(mKW_f \diamond mKW_{\mathrm{STCONN}_n}) &\geq \log \mathsf{L}(mKW_f) + \Omega(\mathsf{Q}(\mathrm{Ind}_\ell) \cdot t) \\
&= \log \mathsf{L}(mKW_f) + \Omega\left(\log \ell \cdot \log(m \cdot \ell)\right) \\
&= \log \mathsf{L}(mKW_f) + \Omega\left(\log^2 n\right) \\
&= \log \mathsf{L}(mKW_f) + \Omega\left(\log \mathsf{L}(mKW_{\mathrm{STCONN}_n})\right).
\end{aligned}$$

Similarly, our semi-monotone composition theorem implies that

$$\begin{aligned}
\mathsf{CC}(U_m \diamond mKW_{\mathrm{STCONN}_n}) &\geq m + \Omega(NS_{\mathbb{F}_2}(\mathrm{Ind}_\ell) \cdot t) \\
&= m + \Omega\left(\mathsf{CC}(mKW_{\mathrm{STCONN}_n})\right),
\end{aligned}$$

as required. ■

The clique function

We denote by $\textsc{Clique}_{n,k}$ the function that takes as an input the adjacency matrix of an n-vertex graph and says whether it contains a k-clique. Observe that for every $k, n \in \mathbb{N}$ it holds that $\mathsf{CC}(mKW_{\textsc{Clique}_{n,k}}) \leq k \log n$, which is witnessed by the circuit that checks all $\binom{n}{k}$ potential cliques by brute force. Goldmann and Håstad [GH92] proved that $\mathsf{CC}(mKW_{\textsc{Clique}_{n,k}}) \geq \Omega(\sqrt{k})$ for every $k \leq (n/2)^{2/3}$, and Raz and Wigderson [RW92] improved this bound to $\mathsf{CC}(mKW_{\textsc{Clique}_{n,k}}) = \Omega(k)$ that for every $k \leq \frac{2}{3}n + 1$. In what follows, we use lifting to improve the latter lower bound to $\Omega(k \log k)$ for small values of k, namely, $k \leq 2^{O(\sqrt{\log n})}$. We then apply our main results to obtain corresponding compositions theorems with the inner function being $g = \textsc{Clique}_{n,k}$.

To this end, we choose our the CNF contradiction to be the bitwise pigeonhole principle, defined as follows: Let $d \in \mathbb{N}$. The *bitwise pigeonhole principle* bitPHP_d is a $2(d-1)$-CNF contradiction over $\ell \stackrel{\text{def}}{=} 2^d \cdot (d-1)$ variables. The variables are partitioned into 2^d blocks of $(d-1)$ variables each, and we view each block as encoding a number in $\left[2^{d-1}\right]$. The formula bitPHP_d contains $\binom{2^d}{2}$ clauses that check that every two blocks encode different numbers. Informally, this formula encodes the statement that 2^d pigeons can not be mapped injectively into 2^{d-1} pigeonholes.

It follows from a recent work of de Rezende et. al. [dRGN$^+$] that $NS_{\mathbb{F}_2}(\mathrm{bitPHP}_d) \geq \Omega(2^d) = \Omega(\ell / \log \ell)$. We have the following monotone projection from $\textsc{CspSat}_{\mathrm{graph}(\mathrm{bitPHP}_d),\Lambda}$ to $\textsc{Clique}_{n,k}$.

Proposition A.8. *For every $d \in \mathbb{N}$ and every finite set Λ, there is a monotone projection from $\textsc{CspSat}_{\mathrm{graph}(\mathrm{bitPHP}_d),\Lambda}$ to $\textsc{Clique}_{n,k}$ for $n \stackrel{\text{def}}{=} 2^d \cdot |\Lambda|^{d-1}$ and $k \stackrel{\text{def}}{=} 2^d$.*

Proof. We construct a monotone projection that maps an input of $\textsc{CspSat}_{\mathrm{graph}(\mathrm{bitPHP}_d),\Lambda}$ to an input of $\textsc{Clique}_{n,k}$. The input of $\textsc{Clique}_{n,k}$ will be a graph G that consists of 2^d classes of $|\Lambda|^{d-1}$ vertices each. Within each class, we label the vertices with strings in Λ^{d-1}. All the edges of G will connect different classes, so a clique will contain at most one vertex from each class.

The edges between the classes are determined by the input of $\textsc{CspSat}_{\mathrm{graph}(\mathrm{bitPHP}_d),\Lambda}$ as follows. Recall that an input of $\textsc{CspSat}_{\mathrm{graph}(\mathrm{bitPHP}_d),\Lambda}$ is a constraint satisfaction problem over of ℓ variables, which are partitioned to 2^d blocks of $(d-1)$ variables each. Moreover, the input to $\textsc{CspSat}_{\mathrm{graph}(\mathrm{bitPHP}_d),\Lambda}$ consists, for every two distinct blocks i, j, of a predicate $P_{i,j} : \Lambda^{d-1} \times \Lambda^{d-1} \to$

$\{0,1\}$. Now, for every distinct i,j, we put in G an edge between a vertex $u \in \Lambda$ of the i-th class and a vertex $v \in \Lambda$ of j-th class if and only if $P_{i,j}(u,v) = 1$.

It can be verified that this mapping from inputs of $\text{CSPSAT}_{\text{graph}(\text{bitPHP}_d),\Lambda}$ to inputs of $\text{CLIQUE}_{n,k}$ is a monotone projection. To see that it maps "yes" inputs of $\text{CSPSAT}_{\text{graph}(\text{bitPHP}_d),\Lambda}$ to "yes" instances of $\text{CLIQUE}_{n,k}$ and vice versa, observe that every satisfying assignment specifies a clique of size 2^d in G and vice versa. ■

We use Proposition A.8 together with the lifting theorem of [dRMN+19] to obtain the following lower bound on $\mathsf{CC}(mKW_{\text{CLIQUE}_{n,k}})$.

Theorem A.9. *For almost all $n \in \mathbb{N}$ it holds that $\mathsf{CC}(mKW_{\text{CLIQUE}_{n,k}}) = \Omega(k \log k)$ for every $k \leq 2^{O(\sqrt{\log n})}$.*

Proof. Let $d \in \mathbb{N}$, let $\ell \stackrel{\text{def}}{=} 2^d \cdot (d-1)$, and let $t \stackrel{\text{def}}{=} \lceil 2 \cdot \log \ell \rceil = \Theta(d)$. We prove the theorem for $n \stackrel{\text{def}}{=} 2^{d+(d-1)\cdot t} = 2^{\Theta(d^2)}$ and $k \stackrel{\text{def}}{=} 2^d = 2^{\Theta(\sqrt{\log n})}$, and this will imply the theorem for almost all $n \in \mathbb{N}$ and $k \leq 2^{O(\sqrt{\log n})}$ by padding. By applying the lifting theorem of [dRMN+19] (specifically, Corollary 2.28) together with the bound $NS_{\mathbb{F}_2}(\text{bitPHP}_d) \geq \Omega(\frac{\ell}{\log \ell})$ of [dRGN+], it holds that

$$\mathsf{CC}(S_{\text{bitPHP}_d} \diamond \text{eq}) \geq \Omega(\frac{\ell}{\log \ell} \cdot t) - \log\left(2(d-1)\right) = \Omega(\ell).$$

By combining Proposition A.8 with Theorem A.3, we obtain a reduction from $S_{\text{bitPHP}_d} \diamond \text{eq}$ to $mKW_{\text{CLIQUE}_{n,k}}$, and therefore.

$$\mathsf{CC}(mKW_{\text{CLIQUE}_{n,k}}) = \Omega(\ell) = \Omega(2^d \cdot d) = \Omega(k \log k),$$

as required. ■

We now apply our main results to obtain a monotone and semi-monotone composition theorems with the inner function being $g = \text{CLIQUE}_{n,k}$.

Theorem A.10. *For every non-constant monotone function $f : \{0,1\}^m \to \{0,1\}$ the following holds: For almost all $n \in \mathbb{N}$ and for $k = 2^{\Theta(\sqrt{\log n})}$,*

$$\log \mathsf{L}(mKW_f \diamond mKW_{\text{STCONN}_n}) = \log \mathsf{L}(mKW_f) + \tilde{\Omega}(\log \mathsf{L}(\text{CLIQUE}_{n,k}))$$
$$\mathsf{CC}(U_m \diamond mKW_{\text{CLIQUE}_{n,k}}) \geq m + \tilde{\Omega}\left(\mathsf{CC}(mKW_{\text{CLIQUE}_{n,k}})\right).$$

Proof. Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant monotone function, and let c be the maximum between 2 and the constant obtained from our monotone composition theorem for $\eta = \frac{1}{2}$ (since the discrepancy of ip is $2^{-\frac{1}{2}t}$). Let $d \in \mathbb{N}$ be such that $d \geq \log m$, let $\ell \stackrel{\text{def}}{=} 2^d \cdot (d-1)$, and let $t \stackrel{\text{def}}{=} \lceil c \cdot \log(m \cdot \ell) \rceil$. We show that the theorem holds for $n \stackrel{\text{def}}{=} 2^{d+(d-1)\cdot t} = 2^{\Theta(d^2)}$ and $k \stackrel{\text{def}}{=} 2^d = 2^{\Theta(\sqrt{\log n})}$, and this will imply the theorem for almost all $n \in \mathbb{N}$ by padding.

By combining Proposition A.8 with Theorem A.3, we obtain injective reductions from $S_{\text{bitPHP}_d} \diamond$ ip and $S_{\text{bitPHP}_d} \diamond \text{eq}$ to $mKW_{\text{CLIQUE}_{n,k}}$. By the aforementioned result of [dRGN+] it holds that $NS_{\mathbb{F}_2}(\text{bitPHP}_d) \geq \Omega(2^d)$, and this implies that $\mathsf{Q}(\text{bitPHP}_d) = \Omega(2^d)$ by Fact A.5. It now follows by our monotone composition theorem that

$$\begin{aligned}
\log \mathsf{L}(mKW_f \diamond mKW_{\text{STCONN}_n}) &\geq \log \mathsf{L}(mKW_f) + \Omega(\mathsf{Q}(\text{bitPHP}_d) \cdot t) \\
&= \log \mathsf{L}(mKW_f) + \Omega\left(2^d \cdot \log(m \cdot \ell)\right) \\
&\geq \log \mathsf{L}(mKW_f) + \Omega\left(2^d \cdot d\right) \qquad \text{(since } d \geq \log m \text{ and } \log \ell = \Theta(d)\text{)} \\
&= \log \mathsf{L}(mKW_f) + \Omega\left(k \log k\right).
\end{aligned}$$

Finally, recall that $\log \mathsf{L}(\textsc{Clique}_{n,k}) = O(k \log n)$. Since $k = 2^{\Theta(\sqrt{\log n})}$, it holds that $k \log k = \tilde{\Omega}(k \log n)$, and therefore

$$\log \mathsf{L}(mKW_f \diamond mKW_{\textsc{stConn}_n}) \geq \log \mathsf{L}(mKW_f) + \tilde{\Omega}(\log \mathsf{L}(\textsc{Clique}_{n,k})).$$

Finally, by applying our semi-monotone composition theorem and using similar calculations as in the proof of Theorem A.9, we obtain that

$$\mathsf{CC}(U_m \diamond mKW_{\textsc{stConn}_n}) \geq m + \Omega(k \log k) \geq m + \tilde{\Omega}\left(\mathsf{CC}(mKW_{\textsc{Clique}_{n,k}})\right),$$

as required. ■

The generation function

Let $n \in \mathbb{N}$. Given a set $\mathcal{T} \subseteq [n]^3$, we say that $\mathcal{T}$ *generates* a point $w \in [n]$ if $w = 1$, or if there is a triplet $(u, v, w) \in \mathcal{T}$ such that $\mathcal{T}$ generates u and v. The generation function $\textsc{Gen}_n$ takes as an input a set $\mathcal{T} \subseteq [n]^3$ and says whether $\mathcal{T}$ generates n or not. This function was introduced by Raz and McKenzie [RM99] in order to separate the monotone **NC** hierarchy.

Raz and McKenzie [RM99] showed that $\mathsf{CC}(mKW_{\textsc{Gen}_n}) \geq \Omega(n^{\varepsilon})$ for some constant $\varepsilon > 0$ by using their lifting theorem for query complexity. Specifically, they considered a certain 3-CNF contradiction Peb_{Δ_h} (namely, the pebbling contradiction of the pyramid graph) and reduced the lifted search problem $S_{\text{Peb}_{\Delta_h}} \diamond \text{gd}$ to $mKW_{\textsc{Gen}_n}$. Recently, Robere [Rob18] applied their method with the lifting theorem for Nullstellensatz degree of [RPRC16] and obtained a bound of $\mathsf{CC}(mKW_{\textsc{Gen}_n}) \geq \Omega(n^{1/6})$. Below, we use our main results to obtain corresponding composition theorems with the inner function being $g = \textsc{Gen}_n$.

For every $h \in \mathbb{N}$, the formula Peb_{Δ_h} has $\ell \stackrel{\text{def}}{=} \frac{h(h+1)}{2}$ variables. It can be shown that $NS_{\mathbb{F}_2}(\text{Peb}_{\Delta_h}) = \Theta(h)$ by combining the results of Cook [Coo74] and Buresh-Oppenheim et. al. [BCIP02] (see [Rob18, Sec 6.3] for details). We use the following result due to Robere [Rob18].

Proposition A.11 (implicit in the proof of [Rob18, Thm. 6.3.3])**.** *For every $h \in \mathbb{N}$ and every finite set Λ, there is a monotone projection from $\textsc{CspSat}_{\text{graph}(\text{Peb}_{\Delta_h}),\Lambda}$ to $\textsc{Gen}_n$ for $n \stackrel{\text{def}}{=} \ell \cdot |\Lambda| + 2$.*

Remark A.12. We note that the proof of Proposition A.11 in [Rob18] only states this claim for $\Lambda = [\ell^2]$, but it actually works for every finite set Λ.

We now apply our main results to obtain a monotone and semi-monotone composition theorems with the inner function being $g = \textsc{Gen}_n$ that match the lower bounds of [RM99] and [Rob18] respectively.

Theorem A.13. *There exists $\varepsilon > 0$ such that the following holds. For every non-constant monotone function $f : \{0,1\}^m \to \{0,1\}$, it holds that for almost all $n \in \mathbb{N}$,*

$$\log \mathsf{L}(mKW_f \diamond mKW_{\textsc{Gen}_n}) = \log \mathsf{L}(mKW_f) + \Omega(n^{\varepsilon}). \tag{22}$$

Proof. Let $f : \{0,1\}^m \to \{0,1\}$ be a non-constant monotone function, and let c be the constant obtained from our monotone composition theorem for $\eta = \frac{1}{2}$ (since the discrepancy of ip is $2^{-\frac{1}{2}t}$). Let $h \in \mathbb{N}$ and $\ell \stackrel{\text{def}}{=} \frac{h(h+1)}{2}$ be such that $\ell \geq m$, and let $t \in \mathbb{N}$ be such that $t \geq c \cdot \log(m \cdot \ell)$. We show that Equation (22) holds for $n \stackrel{\text{def}}{=} \ell \cdot 2^t + 2 \leq h^{4c+2}$ and this will imply the required result for almost all $n \in \mathbb{N}$ by padding.

By combining Proposition A.11 with Theorem A.3, we obtain a reduction from $S_{\mathrm{Peb}_{\Delta_h}} \diamond \mathrm{ip}$ to $mKW_{\mathrm{Peb}_{\Delta_h}}$. By the foregoing discussion $NS_{\mathbb{F}_2}(\mathrm{Peb}_{\Delta_h}) \geq \Omega(h)$, and this implies that $\mathsf{Q}(\mathrm{Ind}_\ell) \geq \Omega(h)$ by Fact A.5. It now follows by our monotone composition theorem that

$$\begin{aligned}\log \mathsf{L}(mKW_f \diamond mKW_{\mathrm{STCONN}_n}) &\geq \log \mathsf{L}(mKW_f) + \Omega(\mathsf{Q}(\mathrm{Ind}_\ell) \cdot t) \\ &\geq \log \mathsf{L}(mKW_f) + \Omega\left(h\right) \\ &= \log \mathsf{L}(mKW_f) + \Omega\left(n^{\frac{1}{4c+2}}\right).\end{aligned}$$

By choosing $\varepsilon = \frac{1}{4c+2}$, we obtain the required result. ■

Theorem A.14. *For every $m \in \mathbb{N}$ and for almost all $n \in \mathbb{N}$ it holds that $\mathsf{CC}(U_m \diamond mKW_{\mathrm{GEN}_n}) \geq m + \Omega\left(n^{1/6}\right)$.*

Proof. Let $m \in \mathbb{N}$. Let $h \in \mathbb{N}$, let $\ell \stackrel{\mathrm{def}}{=} \frac{h(h+1)}{2}$ and let $t = \lceil 2 \log \ell \rceil$. We prove the theorem for $n \stackrel{\mathrm{def}}{=} \ell \cdot 2^t + 2 = \Theta(h^6)$ and this will imply the theorem for almost all $n \in \mathbb{N}$ by padding. By combining Proposition A.6 with Theorem A.3, we obtain an injective reduction from $S_{\mathrm{Peb}_{\Delta_h}} \diamond \mathrm{eq}$ to mKW_{GEN_n}. Moreover, by the foregoing discussion $NS_{\mathbb{F}_2}(\mathrm{Peb}_{\Delta_h}) \geq \Omega(h)$. It now follows by our semi-monotone composition theorem that

$$\begin{aligned}\mathsf{CC}(U_m \diamond mKW_{\mathrm{STCONN}_n}) &\geq m + \Omega(NS_{\mathbb{F}_2}(\mathrm{Peb}_{\Delta_h}) \cdot t) \\ &\geq m + \Omega(h) \\ &= m + \Omega(n^{1/6}).\end{aligned}$$

as required. ■

Acknowledgment

This work was partly carried out while the authors were visiting the Simons Institute for the Theory of Computing in association with the DIMACS/Simons Collaboration on Lower Bounds in Computational Complexity, which is conducted with support from the National Science Foundation.

References

[And87] Alexander E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Moscow University Mathematics Bulletin*, 42(1):24–29, 1987.

[BCIP02] Josh Buresh-Oppenheim, Matthew Clegg, Russell Impagliazzo, and Toniann Pitassi. Homogenization and the polynomial calculus. *Comput. Complex.*, 11(3-4):91–108, 2002.

[BIK+97] Samuel R. Buss, Russell Impagliazzo, Jan Krajícek, Pavel Pudlák, Alexander A. Razborov, and Jirí Sgall. Proof complexity in algebraic systems and bounded depth frege systems with modular counting. *Computational Complexity*, 6(3):256–298, 1997.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[BP98] Samuel R. Buss and Toniann Pitassi. Good degree bounds on nullstellensatz refutations of the induction principle. *J. Comput. Syst. Sci.*, 57(2):162–171, 1998.

[CFK+19] Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:103, 2019.

[CKLM17] Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudorandom properties. *CoRR*, abs/1704.06807, 2017.

[Coo74] Stephen A. Cook. An observation on time-storage trade off. *J. Comput. Syst. Sci.*, 9(3):308–316, 1974.

[DM18] Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. *Computational Complexity*, 27(3):375–462, 2018.

[dRGN+] Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is np-hard. Submitted to FOCS 2020.

[dRMN+19] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:186, 2019.

[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 Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 295–304. IEEE Computer Society, 2016.

[EIRS01] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiri Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10(3):210–246, 2001.

[GH92] Mikael Goldmann and Johan Håstad. A simple lower bound for monotone clique using a communication game. *Inf. Process. Lett.*, 41(4):221–226, 1992.

[GLM+16] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM J. Comput.*, 45(5):1835–1869, 2016.

[GMWW17] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: The composition of a function and a universal relation. *SIAM J. Comput.*, 46(1):114–131, 2017.

[GP18] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Proceedings of IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1077–1088, 2015.

[GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *Proceedings of IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 132–143, 2017.

[GS91] Michelangelo Grigni and Michael Sipser. Monotone separation of Logspace from NC. In *Structure in Complexity Theory Conference*, pages 294–298, 1991.

[Hås98] Johan Håstad. The shrinkage exponent of de morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998.

[HHL18] Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. *SIAM J. Comput.*, 47(1):208–217, 2018.

[HW93] Johan Håstad and Avi Wigderson. Composition of the universal relation. In *Advances in computational complexity theory, AMS-DIMACS*, 1993.

[IN93] Russell Impagliazzo and Noam Nisan. The effect of random restrictions on formula size. *Random Struct. Algorithms*, 4(2):121–134, 1993.

[Khr72] V. M. Khrapchenko. A method of obtaining lower bounds for the complexity of π-schemes. *Mathematical Notes Academy of Sciences USSR*, 10:474–479, 1972.

[KM18] Sajin Koroth and Or Meir. Improved composition theorems for functions and relations. In *RANDOM*, 2018.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3/4):191–204, 1995.

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

[Mei17] Or Meir. On derandomized composition of boolean functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:146, 2017.

[Mei19] Or Meir. Toward better depth lower bounds: Two results on the multiplexor relation. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:120, 2019.

[Oli15] Igor Carboni Oliveira. *Unconditional Lower Bounds in Complexity Theory*. PhD thesis, Columbia University, 2015.

[Pot17] Aaron Potechin. Bounds on monotone switching networks for directed connectivity. *J. ACM*, 64(4):29:1–29:48, 2017.

[PR17] Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1246–1255, 2017.

[PR18] Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1207–1219. ACM, 2018.

[PZ93] Mike Paterson and Uri Zwick. Shrinkage of de morgan formulae under restriction. *Random Struct. Algorithms*, 4(2):135–150, 1993.

[Raz90] Alexander A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990.

[Raz92] Alexander A. Razborov. On submodular complexity measures. In *Poceedings of the London Mathematical Society Symposium on Boolean Function Complexity*, pages 76–83, New York, NY, USA, 1992. Cambridge University Press.

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

[Rob18] Robert Robere. *Unified Lower Bounds for Monotone Computation*. PhD thesis, University of Toronto, 2018.

[RPRC16] Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 406–415, 2016.

[RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992.

[She11] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011.

[Sub61] Bella Abramovna Subbotovskaya. Realizations of linear functions by formulas using +,.,-. *Soviet Mathematics Doklady*, 2:110–112, 1961.

[SZ09] Yaoyun Shi and Yufan Zhu. Quantum communication complexity of block-composed functions. *Quantum Information & Computation*, 9(5):444–460, 2009.

[Tal14] Avishay Tal. Shrinkage of de morgan formulae by spectral techniques. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 551–560, 2014.

[TZ97] Gábor Tardos and Uri Zwick. The communication complexity of the universal relation. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 247–259, 1997.

[WYY17] Xiaodi Wu, Penghui Yao, and Henry S. Yuen. Raz-McKenzie simulation with the inner product gadget. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:10, 2017.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il