Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 151 (2023)

An Improved Composition Theorem of a Universal Relation and Most Functions via Effective Restriction

Hao Wu*

November 12, 2023

Abstract

One of the major open problems in complexity theory is to demonstrate an explicit function which requires super logarithmic depth, to tackle this problem Karchmer, Raz and Wigderson proposed the KRW conjecture about composition of two functions. While this conjecture seems out of our current reach, some relaxed conjectures are suggested to be the stepping stone to the original one. One important kind of relaxed forms is composition about universal relation. We already have strong lower bounds for composition of two universal relations as well as composition of a function and a universal relation. The final jigsaw to complete our understanding of composition about universal relation is the composition of a universal relation and a function. Recently, Ivan Mihajlin and Alexander Smal proved a composition theorem of a universal relation and some function via so called xor composition, that is there exists some function $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{CC}(\mathrm{U}_n \diamond \mathrm{KW}_f) \geq 1.5n - o(n)$ where CC denotes the communication complexity of the problem.

In this paper, we significantly improve their result and present an asymptotically tight and much more general composition theorem of a universal relation and most functions, that is for most functions $f : \{0,1\}^n \to \{0,1\}$ we have $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq m + n - O(\sqrt{m})$ when $m = \omega(\log^2 n), n = \omega(\sqrt{m})$. This is done by a direct proof of composition theorem of a universal relation and a multiplexor in the partially half-duplex model avoiding the xor composition. And the proof works even when the multiplexor only contains a few functions. One crucial ingredient in our proof involves a combinatorial problem of constructing a tree of many leaves and every leaf contains a non-overlapping set of functions. For each leaf, there is a set of inputs such that every function in the leaf takes the same value, that is all functions are restricted. We show how to choose a set of good inputs to effectively restrict these functions to force that the number of functions in each leaf is as small as possible while maintaining the total number of functions in all leaves. This results in a large number of leaves.

*College of Information Engineering, Shanghai Maritime University, Shanghai, China. My email is haowu@shmtu.edu.cn, you can also reach me via wealk@outlook.com.

ISSN 1433-8092

Contents

1 Introduction

One of the major open problems in complexity theory is to demonstrate an explicit function which requires super logarithmic depth, a.k.a, the $\mathbf{P}$ versus $\mathbf{NC^1}$ problem. The current best depth lower bound [Hås98, Tal14] is $(3 - o(1)) \cdot \log n$, and we still don't even know how to obtain a lower bound strictly larger than $3 \log n$. One promising approach to tackle this problem was suggested by Karchmer, Raz and Wigderson [KRW95], they proposed that we should understand the complexity of (block)-composition of Boolean functions. Given two functions $f : \{0,1\}^m \to \{0,1\}$, $g : \{0,1\}^n \to \{0,1\}$, we define their composite function $f \diamond g : (\{0,1\}^n)^m \to \{0,1\}$ as: $f \diamond g\,(x_1, \ldots, x_m) = f\,(g\,(x_1), \ldots, g\,(x_m))$. Given any Boolean function f, we denote the depth complexity of f by $\mathsf{D}(f)$, that is the minimal depth of a circuit of AND, OR and NOT gates of fan-in 2 that computes f. And it is easy to see the depth complexity of $f \diamond g$ is upper-bounded by $\mathsf{D}(f) + \mathsf{D}(g)$ and it is natural to ask whether the depth complexity of $f \diamond g$ is far from this upper bound. Karchmer, Raz and Wigderson [KRW95] conjectured that the depth complexity of $f \diamond g$ is not far from its upper bound:

Conjecture 1.1. *Given two arbitrary non-constant Boolean functions $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$, then $\mathsf{D}(f \diamond g) \approx \mathsf{D}(f) + \mathsf{D}(g)$.*

The merit of this conjecture is, if it is proved and the "approximate equality" is instantiated with proper parameters, then by an argument of iterative composition [KRW95], we will obtain an explicit function with super-logarithmic depth, which separates $\mathbf{P}$ from $\mathbf{NC^1}$. The hope to resolve this conjecture lies in a deep and elegant connection between circuit complexity and communication complexity which is captured by the concept of Karchmer-Wigderson relations [KW90]. Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the Karchmer-Wigderson relation (KW relation for short) of function f, denoted by KW_f, is the following communication problem: Alice gets an input $x \in f^{-1}(1)$, and Bob gets an input $y \in f^{-1}(0)$. The goal of Alice and Bob is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$. Note that since $x \neq y$, there always exists at least one such coordinate.

The key observation by Karchmer and Wigderson [KW90] is that the *deterministic* communication complexity of KW_f is exactly equal to $\mathsf{D}(f)$. This allows us to view the original KRW conjecture from the KW relation perspective. Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant functions. In the KW relation $KW_{f \diamond g}$, the inputs to Alice and Bob are viewed as two $m \times n$ Boolean matrices X, Y. Alice gets $X \in (f \diamond g)^{-1}(1)$ and Bob gets $Y \in (f \diamond g)^{-1}(0)$, their task is to find an entry (i, j) such that $X_{i,j} \neq Y_{i,j}$. Moreover, it is convenient to write $\mathrm{KW}_{f \diamond g}$ as $\mathrm{KW}_f \diamond \mathrm{KW}_g$, indicating that these KW relations could be more general KW relation rather than KW relation of functions, now we can rephrase KRW conjecture in terms of communication complexity:

Conjecture 1.2. *Given two arbitrary non-constant Boolean functions $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$, then $\mathsf{CC}\,(\mathrm{KW}_f \diamond \mathrm{KW}_g) \approx \mathsf{CC}\,(\mathrm{KW}_f) + \mathsf{CC}\,(\mathrm{KW}_g)$, where CC means the deterministic communication complexity of a KW relation.*

Current successes towards KRW conjecture are all restricted cases. There are composition theorems when the inner function g satisfies certain property, for example when the inner function is the parity function [Hås98, Tal14, DM18] and when the inner functions are with

a tight unweighted quantum adversary bound [FMT21]. There are composition theorems where the composition itself is restricted such as monotone composition, semi-monotone composition [dRMN+20] and strong composition [Mei23]. There are also some variants [EIRS01, Mei20, MS21] of original conjecture with the similar effect to the **P** versus $\mathbf{NC^1}$ problem, but we don't know how to prove them either. Maybe to prove the general form of KRW conjecture is out of our reach now. Edmonds, Impagliazzo, Rudich and Sgall [EIRS01] suggested we should consider relaxed form of KRW conjecture and hope that any progresses of these relaxed compositions involve ideas and techniques which will be useful to attack the original KRW conjecture. One choice is to relax the KW relation of function to the universal relation. In the universal relation U_n, Alice and Bob get two distinct strings $x, y \in \{0,1\}^n$, their task is to find a coordinate i such that $x_i \neq y_i$. It is perhaps a necessary starting point for us to study composition of KW relations.

The first challenge is to prove lower bound for composition of two universal relations $\mathrm{U}_m \diamond \mathrm{U}_n$, this was met by [EIRS01, HW93]. The next step is to understand the composition of a function and a universal relation $\mathrm{KW}_f \diamond \mathrm{U}_n$. Gavinsky, Meir, Weinstein and Wigderson [GMWW17] showed a lower bound with a small additive loss, then Koroth and Meir [KM18] improved their result and provided an essential optimal lower bound for $\mathrm{KW}_f \diamond \mathrm{U}_n$. After that, the final jigsaw to complete our understanding of composition about universal relation is composition of a universal relation and a function. Let $f : \{0,1\}^n \to \{0,1\}$ be a function. Given any Boolean matrix $X \in \{0,1\}^{m\times n}$, define $f(X) = (f(X_1), \ldots, f(X_m))$. In KW relation $\mathrm{U}_m \diamond \mathrm{KW}_f$, Alice gets a Boolean matrix $X \in \{0,1\}^{m\times n}$, Bob gets a Boolean matrix $Y \in \{0,1\}^{m\times n}$, their goal is to find an entry (i,j) such that $X_{i,j} \neq Y_{i,j}$. If $f(X) = f(Y)$, they can also output $\perp$. It is natural to make following conjecture [GMWW17, DM18].

Conjecture 1.3. *Given a universal relation* U_m *and a function* $f : \{0,1\}^n \to \{0,1\}$*, then* $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \approx m + \mathsf{CC}(\mathrm{KW}_f)$.

Ivan Mihajlin and Alexander Smal [MS21] took a big step towards Conjecture 1.3 and proved a composition theorem of a universal relation and some function via so called xor composition, that is there exists some function $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{CC}(\mathrm{U}_n \diamond \mathrm{KW}_f) \geq 1.5n - o(n)$. But their result is not tight and works only for some function when $m \simeq n$, they asked whether the success of [GMWW17, KM18] can be achieved in the case of $\mathrm{U}_n \diamond \mathrm{KW}_f$. Thus comparing to the optimal lower bound in the case of $\mathrm{KW}_f \diamond \mathrm{U}_n$, following conjecture should not be too ambitious.

Conjecture 1.4. *Given a universal relation* U_m *and a function* $f : \{0,1\}^n \to \{0,1\}$ *with* m, n *in proper range, then* $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) = m + \mathsf{CC}(\mathrm{KW}_f) - o(\min\{m, \mathsf{CC}(\mathrm{KW}_f)\})$.

In this paper, we make progress towards Conjecture 1.4 and show it is almost true.

1.1 Our results

Our main result is for most functions $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq m + n - O(\sqrt{m})$.

Theorem 1.5. *Let* $m = \omega(\log^2 n), n = \omega(\sqrt{m})$*, if we pick a function* $f : \{0,1\}^n \to \{0,1\}$ *randomly, the probability of* $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq m + n - O(\sqrt{m})$ *is* $1 - o(1)$.

This result follows from a composition theorem of a universal relation and a multiplexor in the partially half-duplex model. And it works even when the multiplexor only contains a few functions. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$. In KW relation $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$, Alice gets a function $f \in \mathcal{F}$ and a Boolean matrix $X \in \{0,1\}^{m \times n}$, Bob gets a function $g \in \mathcal{F}$ and a Boolean matrix $Y \in \{0,1\}^{m \times n}$, their goal is to find an entry (i,j) such that $X_{i,j} \neq Y_{i,j}$. If $f \neq g$ or $f(X) = g(Y)$, they can also output $\perp$.

Theorem 1.6. *Let $m = \omega(\log^2 n), n = \omega(\sqrt{m})$, $\epsilon = \frac{\sqrt{m}}{n}$. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$ such that $|\mathcal{F}| \geq 2^{2^n - 2^{(1-\epsilon)n}}$ we have $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \geq m + n - O(\sqrt{m})$ where $\mathsf{CC}^{\mathsf{phd}}$ denotes the communication complexity in partially half-duplex model.*

Comparison with related works. Comparing to the result of Ivan Mihajlin and Alexander Smal, our result is asymptotically tight and much more general. More importantly, we give a direct proof without using the xor composition. We also note recently Meir [Mei23] proved a result about 'strong' composition of a function and a multiplexor, but so-called strong composition is a restricted form of the standard composition while our result is about standard composition, thus our result is incomparable to Meir's.

Our approach. Here we give a simplified description of our proof of Theorem 1.6. For convenience, assume $n \geq m$ and ignore the difference between standard communication model and the partially half-duplex model. We can prove it via a two-stage argument similar to that in [MS21], that is after the protocol has spent approximate m bits, we are able to extract a set $\mathcal{H}$ of size almost 2^{2^n} from the residual problem, then use this set and the protocol to solve the non-equality problem over $\mathcal{H}$ non-deterministically, thus the protocol will require another approximate n bits. Now we give more details. Let c be an integer which depends on m, n, ϵ. Let $t = c + 4$, $s = m - t - 1$, $\mathcal{X}$ be the set $\{0,1\}^{m \times n}$, $\mathcal{D} = \{((f,X),(f,X)) \mid f \in \mathcal{F}, X \in \mathcal{X}\}$ and d be the depth of the protocol.

- In the first stage, there is a (partial) transcript $\tau \in \{0,1\}^s$ and a subset of inputs $\mathcal{D}' \subseteq \mathcal{D}$ such that every input in $\mathcal{D}'$ is consistent with τ. Intuitively, after spending the s bits in the transcript τ, the residual protocol still has to solve all inputs from the set $\mathcal{D}'$. Furthermore, there is a set $\mathcal{S} \subseteq \mathcal{F} \times \mathcal{X}$ such that $\{((f,X),(f,X)) \mid (f,X) \in \mathcal{S}\} \subseteq \mathcal{D}'$ and
  - $|\mathcal{S}| \geq 2^{t-m} \cdot |\mathcal{F}| \cdot |\mathcal{X}|$.
  - Let $\mathcal{U}_{\mathcal{S}} = \{f \mid (f,X) \in \mathcal{S}\}$, for every $f \in \mathcal{U}_{\mathcal{S}}$, $|\mathcal{X}_{\mathcal{S},f}| \geq 2^{t-m} \cdot |\mathcal{X}|$.

  Eventually we can extract a subset $\mathcal{H} \subseteq \mathcal{U}_{\mathcal{S}}$ of size at least $2^{2^{(1-\epsilon)n}}$ such that for all distinct $f, g \in \mathcal{H}$, there exists an $X : (f,X),(g,X) \in \mathcal{S}$, and $f(X) \neq g(X)$.
- In the second stage, recall that the depth of residual protocol is at most $d - s$ and by the rectangle property it must correctly solve every input from set $\{(f,X),(g,X) \mid (f,X),(g,X) \in \mathcal{S}, f, g \in \mathcal{H}\}$. We can leverage this fact to non-deterministically solve the non-equality problem over $\mathcal{H}$ with a witness of size $d - s + O(\sqrt{m})$.

Since the nondeterministic complexity of the non-equality problem over $\mathcal{H}$ is at least $\log\log\mathcal{H}$, we have $d - s + O(\sqrt{m}) \geq \log\log\mathcal{H}$, that is the depth $d \geq m + n - O(\sqrt{m})$.

Let's take a glimpse at how to effectively extract the set $\mathcal{H}$, see more details in Lemma 3.4. The extraction involves a combinatorial problem of constructing a tree and every leaf of the tree contains a non-overlapping set of functions, then the set $\mathcal{H}$ takes one function from each leaf. The tree is constructed recursively. Each node z in the tree is associated with a set $\mathcal{Z} \subseteq \mathcal{S}$, let $\mathcal{U}_{\mathcal{Z}} = \{f \mid (f, X) \in \mathcal{Z}\}$, then every node in the same depth contains a non-overlapping set $\mathcal{U}_{\mathcal{Z}}$ of functions. Assume z is at depth d, from root to node z, its ancestors are $z_0, z_1, \ldots, z_{d-1}$. For every $i \in \{0, 1, \ldots, d-1\}$, z_i is labeled with X_{z_i}, treat every X_{z_i} as a set of its distinct rows, we define $\Psi(z) = \bigcup_{i=0}^{d-1} X_{z_i}$. Then given inputs from $\Psi(z)$, every $f \in \mathcal{U}_{\mathcal{Z}}$ takes the same value thus restricted in $\{0,1\}^m \setminus \Psi(z)$. Now it's turn to choose a good X_z for node z to restrict functions in the children of z as much as possible, meanwhile maintaining the total number of functions in all its children. Fortunately, we can choose a good X_z in each step downward, such that the number of functions in each child decreases by a factor of (at least) 2^{m-c} while the total number of functions in all its children decreases by a smaller (average) factor of (at most) 2^{m+3-t}. The parameters are carefully chosen to make sure that $t = c + 4$, and finally, at depth $h = 2^{\lceil(1-\epsilon)n\rceil}$, the total number of functions in all leaves is $(2^{-(m+3-t)}/2^{-(m-c)})^h = 2^{(m-c-(m+3-t))h} = 2^h$ times bigger than the number of functions in each leaf, thus we obtain a set $\mathcal{H}$ of size at least $2^{2^{(1-\epsilon)n}}$. We have omitted some technicalities in the full proof as follows.

- When $n \ll m$, in the first stage, we can only obtain a short transcript τ such that $|\tau| \approx n \ll m$, thus single shot of two-stage augment is not sufficient. Nevertheless, we can use the two-stage argument multiple times to boost the complexity up until it's done. See the discussion at the beginning of Section 3.

- The second problem is the difference between the standard communication model and the partially half-duplex model, and the argument has to be tuned to be compatible with the partially half-duplex model. Nonetheless, this problem can be overcome in a similar way like that in [MS21], see more details in Section 2.3 and Lemma 3.2.

1.2 Organization of the rest of the paper

The rest of the paper is organized as follows. In Section 2, we provide necessary preliminaries. It is highly recommended not to skip Section 2.2 and 2.3, particularly, we explain how we avoid xor composition in Section 2.2. In Section 3, we prove Theorem 1.6, a composition theorem of a universal relation and a multiplexor in the model of partially half-duplex communication with adversary. In Section 4, we prove Theorem 1.5, a composition theorem of a universal relation and most functions in the standard model of communication. In Section 5, we make some discussion and point out some future directions.

2 Preliminaries and Notations

In this section, we provide some basic notations, definitions and facts. Let $\mathbb{N}^+$ be the set of positive natural numbers, for any $n \in \mathbb{N}^+$, we denote by $[n]$ the set $\{1, \ldots, n\}$. Let

$x \in \{0,1\}^n$ be a Boolean string, we denote the i-th bit of x by x_i. Let $X \in \{0,1\}^{m\times n}$ be an $m \times n$ Boolean matrix, we denote the i-th row of X by X_i and the entry at (i,j) by $X_{i,j}$.

2.1 Communication complexity

We assume the readers are familiar with the basic knowledge of communication complexity, a more detailed introduction to communication complexity can be found in textbooks such as [KN97, RY20].

Definition 2.1 (Two party communication problems)**.** In a two-party communication problem $S \subseteq (X \times Y) \times Z$, there are two involved players, Alice and Bob, who need to solve following task: Alice is given an input $x \in X$ and Bob is given an input $y \in Y$, they need to output a value $z \in Z$ such that $(x, y, z) \in S$.

Deterministic protocol

Definition 2.2. A deterministic protocol $\Pi : X \times Y \to Z$ for a communication problem $S \subseteq (X \times Y) \times Z$ is a rooted binary tree with following structure:

- Every node v in the tree belongs to Alice or Bob and is associated with a rectangle $X_v \times Y_v \subseteq X \times Y$. Particularly, the root of protocol tree is associated with the rectangle $X \times Y$.
- Every internal node v has two outgoing edges labeled with 0 and 1 respectively. These two edges labeled with 0 and 1 lead to v's two children v_0, v_1 respectively.
- Recall v is associated with a rectangle $X_v \times Y_v$, if v is owned by Alice, then v_0 is associated with $X_{v_0} \times Y_v$, v_1 is associated with $X_{v_1} \times Y_v$ where $X_{v_0} \cap X_{v_1} = \emptyset$ and $X_{v_0} \cup X_{v_1} = X_v$; if v is owned by Bob, then v_0 is associated with $X_v \times Y_{v_0}$, v_1 is associated with $X_v \times Y_{v_1}$ where $Y_{v_0} \cap Y_{v_1} = \emptyset$ and $Y_{v_0} \cup Y_{v_1} = Y_v$.
- Every leaf node ℓ is associated with a value $z \in Z$ as the output of the protocol. And for every leaf ℓ, we have $X_\ell \times Y_\ell \times \{z\} \subseteq S$.

Definition 2.3. Given a protocol tree Π and a node v in the tree, the transcript of node v is the string obtained by concatenating the labels of the edges in the path from the root to the node v.

Definition 2.4. Given a protocol tree Π, its depth $\mathsf{D}(\Pi)$ is the length of the longest path from the root to a leaf in the tree. Given a communication problem $S \subseteq (X \times Y) \times Z$, the (deterministic) communication complexity $\mathsf{CC}(S)$ of communication problem S is the minimum $\mathsf{D}(\Pi)$ over all protocol Π for the problem S.

Non-Deterministic protocol

Definition 2.5 (Non-deterministic communication protocol [KN97, MS21])**.** Given a function $f : X \times Y \to \{0,1\}$, we say it has non-deterministic communication protocol of complexity d if there are two functions $A : X \times \{0,1\}^d \to \{0,1\}$ and $B : Y \times \{0,1\}^d \to \{0,1\}$ such that

- $\forall (x,y) \in f^{-1}(1)\ \exists w \in \{0,1\}^d : A(x,w) = B(y,w) = 1$,
- $\forall (x,y) \in f^{-1}(0)\ \forall w \in \{0,1\}^d : A(x,w) \neq 1$ or $B(y,w) \neq 1$.

The non-deterministic communication complexity of f , denoted by $\mathsf{NCC}(f)$, is the minimal complexity over all non-deterministic communication protocols for f .

Definition 2.6 (Privately non-deterministic communication protocol [KN97, MS21])**.** Given a function $f : X \times Y \to \{0,1\}$, we say it has privately non-deterministic communication protocol of complexity d if there is a function $\hat{f} : (X \times \{0,1\}^*) \times (Y \times \{0,1\}^*) \to \{0,1\}$ such that

- $\forall (x,y) \in f^{-1}(1)\ \exists w_x, w_y \in \{0,1\}^* : \hat{f}((x,w_x),(y,w_y)) = 1$,
- $\forall (x,y) \in f^{-1}(0)\ \forall w_x, w_y \in \{0,1\}^* : \hat{f}((x,w_x),(y,w_y)) = 0$,

and (deterministic) communication complexity of $\hat{f}$ is at most d. The privately non-deterministic communication complexity of f, denoted by $\mathsf{NCC}'(f)$, is the minimal complexity over all privately non-deterministic communication protocols for f .

Theorem 2.7 ([MS21])**.** *For any function $f : X \times Y \to \{0,1\}$, we have*

$$\mathsf{NCC}(f) + 2 \geq \mathsf{NCC}'(f) \geq \mathsf{NCC}(f).$$

Non-Deterministic complexity of non-equality problem

Definition 2.8 (The non-equality problem)**.** Given a non-empty finite set S, the non-equality on S is the function $\mathrm{NEQ}_S : S \times S \to \{0,1\}$ defined as follows: $\mathrm{NEQ}_S(x,y) = 1$ if and only if $x \neq y$.

Fact 2.9 ([MS21])**.** Given any non-empty finite set S, $\mathsf{NCC}'(\mathrm{NEQ}_S) \geq \log\log|S|$.

2.2 Karchmer-Wigderson relations and their compositions

We start by defining the universal relation and other involved Karchmer-Wigderson relations, then we define compositions of these relations.

Definition 2.10 (Universal relation U_n)**.** The Universal relation U_n is the following communication problem: Alice and Bob get inputs $x, y \in \{0,1\}^n$ respectively. Their task is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$. If $x = y$, they can also output $\perp$.

Definition 2.11 (KW relation over rectangle)**.** Given two disjoint sets $X, Y \subseteq \{0,1\}^n$, the KW relation over rectangle $X \times Y$, denoted by $\mathrm{KW}_{X\times Y}$ is defined by

$$\mathrm{KW}_{X\times Y} \stackrel{\text{def}}{=} \{(x,y,i) \mid x \in X, y \in Y, x_i \neq y_i\}.$$

Definition 2.12 (KW relation for functions)**.** Given a non-constant function $f : \{0,1\}^n \to \{0,1\}$, its KW relation KW_f is defined by $\mathrm{KW}_f \stackrel{\text{def}}{=} \mathrm{KW}_{f^{-1}(1)\times f^{-1}(0)}$.

Definition 2.13 (The multiplexor relation MUX_n)**.** In KW relation MUX_n, Alice gets a function $f : \{0,1\}^n \to \{0,1\}$ and a Boolean string $x \in \{0,1\}^n$, Bob gets a function $g : \{0,1\}^n \to \{0,1\}$ and a Boolean string $y \in \{0,1\}^n$, their goal is to find an entry i such that $x_i \neq y_i$. If $f \neq g$ or $f(x) = g(y)$, they can also output $\perp$.

Remark 2.14. Here we want to point out in the original version of multiplexor, the inputs to the players are promised to satisfy $f = g$ and $f(x) \neq g(x)$. Here we use the rejectable version of multiplexor, that is when the promise is false, the players are allowed to reject and output $\perp$. The difference of the complexities of two versions is only two bits, for example, Alice can send the i-th bit to Bob, Bob replies with one bit that whether the answer i is correct, if not, they output $\perp$. Thus, we ignore such difference and in the rest of the paper, for problems similar to the multiplexor problem, we all present their rejectable versions.

Definition 2.15 (Composition of two Boolean functions and its KW relation)**.** Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant functions. The (block) composition of f and g, denoted by $f \diamond g : \{0,1\}^{m\times n} \to \{0,1\}$, is defined as follows:

$$f \diamond g(X) = f(g(X_1), \ldots, g(X_m)).$$

In KW relation $\mathrm{KW}_{f\diamond g}$, Alice and Bob get $X \in (f \diamond g)^{-1}(1)$ and $Y \in (f \diamond g)^{-1}(0)$ viewed as $m \times n$ Boolean matrices, and their goal is to find an entry (i, j) such that $X_{i,j} \neq Y_{i,j}$. We also denote $\mathrm{KW}_{f\diamond g}$ by $\mathrm{KW}_f \diamond \mathrm{KW}_g$.

Definition 2.16 (Composition of a universal relation and a Boolean function)**.** Let U_m be the universal relation and $f : \{0,1\}^n \to \{0,1\}$ be a function. Given any Boolean matrix $X \in \{0,1\}^{m\times n}$, define $f(X) = (f(X_1), \ldots, f(X_m))$. In KW relation $\mathrm{U}_m \diamond \mathrm{KW}_f$, Alice gets a Boolean matrix $X \in \{0,1\}^{m\times n}$, Bob gets a Boolean matrix $Y \in \{0,1\}^{m\times n}$, their goal is to find an entry (i, j) such that $X_{i,j} \neq Y_{i,j}$. If $f(X) = f(Y)$, they can also output $\perp$.

Definition 2.17 (Composition of a universal relation and a multiplexor)**.** Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$. In KW relation $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$, Alice gets a function $f \in \mathcal{F}$ and a Boolean matrix $X \in \{0,1\}^{m\times n}$, Bob gets a function $g \in \mathcal{F}$ and a Boolean matrix $Y \in \{0,1\}^{m\times n}$, their goal is to find an entry (i, j) such that $X_{i,j} \neq Y_{i,j}$. If $f \neq g$ or $f(X) = g(Y)$, they can also output $\perp$.

Now we make a detour to show how our idea emerges from the xor composition and finally avoids it. We start with the notion of generalized KW relations. Ivan Mihajlin and Alexander Smal [IMS22] considered a more general form of KW relation including the case of non Boolean functions.

Definition 2.18 (The generalized KW relation)**.** Given a function $f : \{0,1\}^n \to \{0,1\}^k$, its KW relation KW_f is defined by $\mathrm{KW}_f \stackrel{\text{def}}{=} \{(x, y, i) \mid x, y \in \{0,1\}^n, f(x) \neq f(y), x_i \neq y_i\}$. The generalized KW relation KW_f is the following communication problem: Alice and Bob get inputs $x, y \in \{0,1\}^n$ respectively. Their task is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$ with the promise $f(x) \neq f(y)$.

We will focus on a special form of generalized KW relation.

Definition 2.19 (Function bundle)**.** An (m, n) function bundle $F = (F_1, \ldots, F_m)$ is a tuple of m functions $F_1, \ldots, F_m : \{0,1\}^n \to \{0,1\}$. We also treat F as a function of $\{0,1\}^{m\times n} \to \{0,1\}^m$ defined as follows: $F(X) = (F_1(X_1), \ldots, F_m(X_m))$.

Let F be an (m, n) function bundle, the generalized KW relation KW_F is the following communication problem: Alice and Bob get two Boolean matrices $X, Y \in \{0,1\}^{m\times n}$ respectively. Their task is to find an entry (i, j) such that $X_{i,j} \neq Y_{i,j}$ with the promise $F(X) \neq F(Y)$. The merit of studying this special form of generalized KW relation is illustrated in following fact which is implicit in [MS21].

Fact 2.20. Given an (m, n) function bundle F, define a function $h : \{0,1\}^{\log m+n} \to \{0,1\}$ such that $h(i, x) = F_i(x)$ where $i \in \{0,1\}^{\log m}, x \in \{0,1\}^n$. Then, $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_h) \geq \mathsf{CC}(\mathrm{KW}_F)$.

We also can define a multiplexor of function bundles with restricted inputs.

Definition 2.21. Let $\mathcal{F}$ be a set of function bundles and $\mathcal{X} \subseteq \{0,1\}^{m\times n}$. In a communication problem $\mathrm{MUX}_{\mathcal{F},\mathcal{X}}$, Alice gets a function bundle $F \in \mathcal{F}$ and an $X \in \mathcal{X}$, Bob gets a function bundle $G \in \mathcal{F}$ and a $Y \in \mathcal{X}$. Their goal is to find (i, j) such that $X_{i,j} \neq Y_{i,j}$. If $F \neq G$ or $F(X) = G(Y)$, they can output $\perp$.

Ivan Mihajlin and Alexander Smal introduced a so called xor composition which is crucial for their results. They defined the xor composition of a universal relation and a multiplexor as follows.

Definition 2.22 ([MS21])**.** In a communication problem $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$, Alice is given a permutation function $F \in \{0,1\}^n \to \{0,1\}^n$ and two strings $a, x \in \{0,1\}^n$, Bob is given a permutation function $G \in \{0,1\}^n \to \{0,1\}^n$ and two strings $b, y \in \{0,1\}^n$. Let $\circ$ be concatenation of strings and $\oplus$ be bit-wise xor. Their goal is to find $i \in [2n]$ such that $(a \circ x)_i \neq (b \circ y)_i$. If $F \neq G$ or $a \oplus F(x) = b \oplus G(y)$, they can output $\perp$.

Ivan Mihajlin and Alexander Smal [MS21] proved $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_n \boxplus \mathrm{MUX}'_n) \geq 1.5n - o(n)$ where $\mathsf{CC}^{\mathsf{phd}}$ denotes the communication complexity in partially half-duplex model. Let's see that the above xor composition $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ can be viewed as a multiplexor of function bundles which take a restricted form of inputs.

Fact 2.23. Let $\mathcal{P}$ be the set of all permutation functions over n bit strings. Let $\mathcal{F}$ be a set of $(n, n + 1)$ function bundles such that every $F = (F_1, \ldots, F_n) \in \mathcal{F}$ is generated from a permutation $G \in \mathcal{P}$. Given a permutation $G \in \mathcal{P}$, for every $i \in [n]$, define $F_i : \{0,1\}^{n+1} \to \{0,1\}$ to be $F_i(x \circ z) = G(x)_i \oplus z$ where $x \in \{0,1\}^n, z \in \{0,1\}$. Let $\mathcal{X} = \{X \in \{0,1\}^{n\times(n+1)} \mid x, a \in \{0,1\}^n, \forall i, X_i = x \circ a_i\}$. Then the communication problem $\mathrm{MUX}_{\mathcal{F},\mathcal{X}}$ is essentially the same as the communication problem $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$.

Our idea originates in trying to improve the xor composition theorem of Ivan Mihajlin and Alexander Smal, soon we find out it is in fact a special form of multiplexor of function bundles. Then we prove an almost tight lower bound of KW_F for most function bundles. But by Fact 2.20, this only implies a lower bound of $\mathrm{U}_m \diamond \mathrm{KW}_h$ for many functions rather than for most functions. Finally, we manage to prove the almost tight lower bound of $\mathrm{U}_m \diamond \mathrm{KW}_f$ for most functions with a refined restriction technique.

2.3 Half-duplex communication complexity

To handle communication problems like the multiplexor problem, Hoover, Impagliazzo, Mihajlin and Smal [HIMS18] proposed a generalization of the classical communication model, the half-duplex model. Unlike Yao's classical model of communication [Yao79], in each round of the half-duplex model, the players can synchronize their clocks and perform actions simultaneously. At the beginning of one clock cycle, each player takes one of three actions: send 0, send 1, or receive. If the action of receive is taken, the player listens to the communication channel and receives one bit at the end of the clock cycle. Thus at the end of every round, each player will eventually perform one of four actions: receive 0 (`r(0)` for short), receive 1 (`r(1)` for short), send 0 (`s(0)` for short), and send 1 (`s(1)` for short). Let `Action = {r0,r1,s0,s1}` be the set of all such actions. According to the actions taken by the players, intuitively, there are three different kinds of rounds: *classical*, *wasted* and *silent*.

- In a *classical* round, one player sends some bit and the other one receives such bit as the case in the classical model of communication.
- In a *wasted* round, both players send bits but since no one is listening, such bits never get received thus wasted.
- In a *silent* round, both players receive. Since no one is actually speaking, the channel is silent in this round.

Here the tricky thing is about the silent round, since at the end of a silent round, both players eventually receive certain bits those neither of players send. There are different ways [HIMS18] to determine those bits received in a silent round, in this paper, we focus on the model of half-duplex communication with adversary where in silent round both players receive certain bits which are chosen by an adversary. Formally, we have following definition.

Definition 2.24 (Half-duplex protocol with adversary)**.** A deterministic half-duplex protocol with adversary $\Pi : X \times Y \to Z$ for a communication problem $S \subseteq (X \times Y) \times Z$ is a pair of full 4-ary trees (Π_A, Π_B) with the same depth d owned by Alice and Bob respectively. And two trees Π_A, Π_B are with the following structure:

- Every node v in the tree Π_A (respectively Π_B) is associated with a subset $X_v \subseteq X$ (respectively $Y_v \subseteq Y$). Particularly, if the node v is the root of Π_A (respectively Π_B), it is associated with X (respectively Y). Each node v represents certain state in the tree, a state pair (u, v) from two trees Π_A, Π_B represents certain state in the protocol Π, thus we will also treat u, v as states. The subset X_v (respectively Y_v) is the set of inputs to Alice (respectively Bob) that can reach the node v from the root of tree Π_A (respectively Π_B).
- Every internal node v has 4 outgoing edges. Each edge is labeled with one action `ac` from `{r(0),r(1),s(0),s(1)}` respectively. Each edge labeled with the action `ac` leads to v's child $v_{\texttt{ac}}$.
- Recall that each node v of tree Π_A is associated with a subset $X_v \subseteq X$, then X_v is partitioned into three disjoint subsets $X_{v:\texttt{r}}, X_{v:\texttt{s(0)}}, X_{v:\texttt{s(1)}}$. Similarity, every node v in

the tree Π_B is associated with a subset $Y_v \subseteq Y$ and Y_v is partitioned into three disjoint subsets $Y_{v:\mathtt{r}}, Y_{v:\mathtt{s(0)}}, Y_{v:\mathtt{s(1)}}$. The partition of inputs indicates what action the player takes at the beginning of each round.

- Every leaf node ℓ is associated with a value $z_\ell \in Z$ as the output of the protocol.

Now let's see how the protocol find out the answer to any input $(x, y) \in X \times Y$. The protocol maintains a pair of states (u, v) where u, v are nodes at the same depth in the trees Π_A, Π_B. Alice knows the state u and the input x, meanwhile Bob holds the state v and the input y. Initially, (u, v) are the two roots of trees Π_A, Π_B. When u, v are not leaves in the trees Π_A, Π_B, the protocol takes some action from following cases and updates the pair of states until u, v are leaves.

- If $x \in X_{u:\mathtt{s(b)}}$ for some $\mathtt{b} \in \{0, 1\}$ and $y \in Y_{v:\mathtt{r}}$, Alice sends a bit $\mathtt{b}$ and Bob receives such bit $\mathtt{b}$. The protocol updates the state pair (u, v) to new state pair $(u_{\mathtt{s(b)}}, v_{\mathtt{r(b)}})$. This is a *classical* round.

- Similarly, if $x \in X_{u:\mathtt{r}}$ and $y \in Y_{v:\mathtt{s(b)}}$ for some $\mathtt{b} \in \{0, 1\}$, Bob sends a bit $\mathtt{b}$ and Alice receives such bit $\mathtt{b}$. The protocol updates the state pair (u, v) to new state pair $(u_{\mathtt{r(b)}}, v_{\mathtt{s(b)}})$. This is also a *classical* round.

- If $x \in X_{u:\mathtt{s(b)}}$ for some $\mathtt{b} \in \{0, 1\}$ and $y \in Y_{v:\mathtt{s(d)}}$ for some $\mathtt{d} \in \{0, 1\}$, Alice sends a bit $\mathtt{b}$ and Bob sends a bit $\mathtt{d}$. The protocol updates the state pair (u, v) to new state pair $(u_{\mathtt{s(b)}}, v_{\mathtt{s(d)}})$. This is a *wasted* round.

- If $x \in X_{u:\mathtt{r}}$ and $y \in Y_{v:\mathtt{r}}$, the adversary chooses two bits $\mathtt{b,d} \in \{0, 1\}$, Alice receives bit $\mathtt{b}$ and Bob receives bit $\mathtt{d}$. The protocol updates the state pair (u, v) to new state pair $(u_{\mathtt{r(b)}}, v_{\mathtt{r(d)}})$. This is a *silent* round.

When the protocol finally reaches a pair of states (u, v) where u, v are leaves of trees Π_A, Π_B respectively, the protocol outputs the result (z_u, z_v). We say the protocol Π (correctly) solves the problem $S \subseteq X \times Y \times Z$, if for every input (x, y), the protocol Π reaches some pair of states (u, v) where u, v are leaves such that $z_u = z_v = z$ and $(x, y, z) \in S$ no matter what bits are chosen by the adversary in any silent round. The complexity $\mathsf{CC}^{\mathsf{hd}}(\Pi)$ of the protocol Π is the depth d of two trees Π_A, Π_B, recall that we require two trees Π_A, Π_B are of the same depth d. The deterministic communication complexity of S in half-duplex model with adversary, denoted by $\mathsf{CC}^{\mathsf{hd}}(S)$, is the minimal complexity over all deterministic half-duplex protocol with adversary for S.

Now we introduce some useful notations and facts. The first notion is the legal action pair. Recall that in every round, eventually Alice and Bob take some $\mathtt{ac}_A, \mathtt{ac}_B \in \mathtt{Action}$ respectively, those two actions $\mathtt{ac}_A, \mathtt{ac}_B$ form an action pair $(\mathtt{ac}_A, \mathtt{ac}_B)$. But not every action pair from $\mathtt{Action} \times \mathtt{Action}$ is legal, particularly in the classical round, the bit sent must be the same as the bit received, thus action pairs such as

$$(\mathtt{s(1),r(0)}), (\mathtt{s(0),r(1)}), (\mathtt{r(1),s(0)}), (\mathtt{r(0),s(1)})$$

are all *illegal*. Now let σ be a sequence of legal action pairs, let σ_A (respectively σ_B) be a sequence of actions taken by Alice (respectively Bob), then σ determines a unique legal

state pair (u, v) where u, v are determined by σ_A, σ_B in two tree Π_A, Π_B respectively. Indeed, given a sequence σ_A of actions taken by Alice, σ_A defines the unique path from the root to u, the case for v is similar. Now we try to define the transcript in the model of half-duplex communication with adversary and make it compatible to the transcript in the classical model of communication. Given a sequence σ_A of actions taken by Alice, let the $\pi(\sigma_A)$ be the ordered bits involved in the actions, we say $\pi(\sigma_A)$ is Alice's transcript. Similarity, let the $\pi(\sigma_B)$ be the ordered bits involved in the actions taken by Bob, we say $\pi(\sigma_B)$ is Bob's transcript. But Alice's transcript is not always consistent with the one of Bob, thus in general, we can not have transcript for the entire protocol. Nevertheless, if all action pairs in a sequence are classical, we can have a consistent transcript for both players. Formally, We have following definitions.

Definition 2.25. Let $(\mathtt{ac}_A, \mathtt{ac}_B)$ be an action pair taken in a classical round, we say it is a classical action pair. If an action pair sequence σ contains only classical action pairs, we say the sequence σ is classical. Let (u, v) be the state pair determined by a classical sequence σ of action pairs, we say (u, v) is a classical state pair. Let σ be a classical sequence of action pairs and (u, v) be the state pair determined by σ, let $\pi \in \{0, 1\}^*$ be the ordered bits involved in sequence σ, we say π is a protocol's transcript. Furthermore, we say both σ and (u, v) are consistent with protocol's transcript π. Note that due to different choices of the sender, there may be several classical state pairs at given depth such that all of them are consistent with one same protocol's transcript. For simplicity, if we say π is a transcript, we mean it's a protocol's transcript rather than some player's transcript.

In classical model of communication, one important property is the rectangle property. That is there is a rectangle associated with each node v in the protocol tree. But this is not true in half-duplex model with adversary, due to the interference of the adversary. In general, it is not true that for every state pair (u, v) the inputs which reach (u, v) form a rectangle. Nevertheless, if we concern the classical state pair, the rectangle property is true.

Definition 2.26. Given a input (x, y), if the protocol reaches a state pair (u, v) along some sequence σ of action pairs, we say the input (x, y) is consistent with the state pair (u, v). More over if the state pair (u, v) is consistent with a transcript π, we say input (x, y) is also consistent with the transcript π.

Remark 2.27. Note that due to the adversary, one input (x, y) may be consistent with several distinct state pairs at given depth, but one input (x, y) can only be consistent with at most one classical state pair at given depth and one protocol's transcript of given length, since the adversary can not interfere any classical round.

Now we show the rectangle property is true for every classical state pair.

Fact 2.28. Given two input pairs $(x, y), (x', y')$, if both $(x, y), (x', y')$ are consistent with some classical state pair (u, v), then input pairs $(x, y'), (x', y)$ are also consistent with the classical state pair (u, v).

Proof. We prove this fact by induction on the depth of the state pair. Initially, (u, v) are roots of two trees Π_A, Π_B, X(respectively Y) is associated with root u(respectively v), if

both $(x,y),(x',y')$ are consistent with the classical state pair (u,v), $x,x' \in X$ and $y,y' \in Y$, thus $(x,y'),(x',y)$ are also consistent with the classical state pair (u,v). Now assume there are two input pairs $(x,y),(x',y')$ which are consistent with some classical state pair (u',v'), and let u,v be parent nodes of u',v' respectively. W.l.o.g., assume the classical state pair (u,v) transits to (u',v') via action pair $(\mathtt{s(0),r(0)})$. Since $(x,y),(x',y')$ are consistent with (u',v'), they must be also consistent with (u,v) in the first place, by induction hypothesis, $(x,y'),(x',y)$ are consistent with (u,v), we will show $(x,y'),(x',y)$ are also consistent with (u',v'). Now since $(x,y),(x',y')$ are consistent with (u',v'), $x,x' \in X_{u:\mathtt{s(0)}}$ and $y,y' \in Y_{v:\mathtt{r}}$. Therefore, given input pairs $(x,y'),(x',y)$ at state pair (u,v), after Alice and Bob take actions $\mathtt{s(0),r(0)}$ respectively, the protocol also enters state (u',v'), thus $(x,y'),(x',y)$ are also consistent with (u',v') as required. □

Partially half-duplex communication When we handle a communication problem similar to the multiplexor, we consider a more restricted model of half-duplex communication with adversary which is called the *partially half-duplex* communication model. In such model, each player's input contains two parts: Alice gets (f,x) and Bob gets (g,y). They can use a half-duplex protocol for their task but not with its full power, when $f = g$, the protocol is only allowed to perform classical rounds. We use $\mathsf{CC}^{\mathsf{phd}}$ to denote the communication complexity of a problem in partially half-duplex model with adversary.

Fact 2.29. Let Π be a partially half-duplex protocol for some communication problem and the depth of Π is at least d. Let $\mathcal{D}$ be a set of inputs to the protocol and every input to the protocol in $\mathcal{D}$ is of form $((f,x),(f,x'))$, then there is a transcript $\tau \in \{0,1\}^d$ and a subset $\mathcal{D}' \subseteq \mathcal{D}$ such that $|\mathcal{D}'| \geq |\mathcal{D}|/2^d$ and every input in $\mathcal{D}'$ is consistent with the transcript τ.

Proof. Since the depth of Π is at least d, there must be transcripts of length d. Given any fixed input $((f,x),(f,x'))$ in $\mathcal{D}$, since Π is partially half-duplex, $((f,x),(f,x'))$ must be consistent with some transcript τ of length d. Moreover, there are at most 2^d such transcripts, there must be one τ and a subset $\mathcal{D}' \subseteq \mathcal{D}$ of size at least $|\mathcal{D}|/2^d$ such that every input in $\mathcal{D}'$ is consistent with τ. □

3 A Composition Theorem of a Universal Relation and a Multiplexor

In this section, we prove the lower bound for $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$ in the model of partially half-duplex communication with adversary. At first, let's see the overall strategy of the proof. When $n \geq m$, we can use a two-stage argument to show that after the protocol has spent approximate m bits communication, it still needs another approximate n bits to completely solve the problem. After spent approximate m bits, we can extract a set of inputs from the residual problem and use it to solve the non-equality problem of size approximate 2^{2^n} non-deterministically thus the protocol needs another approximate n bits.

But when n is much smaller than m, there is some subtle issue about this argument. In order to apply the two stage argument we must be able to show the protocol needs to spend about m bits in the first stage, but now we are only able to show that the protocol

needs to spend about n bits in the first stage. Nevertheless, we can repeatedly use the two stage argument to boost the complexity of the protocol up until it's done. In general, we can show a boosting theorem that is after the protocol has spent $s \leq m - o(m)$ bits in first stage, it still requires another approximate n bits to compete the task. We can repeatedly use the boosting theorem to add approximate n to s until s is about m, then we add a final approximate n to s and obtain the final complexity which is about $m + n$.

More formally, the boosting theorem depends on two following lemmas: a boosting lemma and an extraction lemma. Let ϵ, c, t be parameters which depend on m, n. Assume a protocol Π has spent $s \leq m - t - 1$ bits, let $\mathcal{S}$ be a subset of $\mathcal{F} \times \mathcal{X}$, the residual protocol has to solve every input of form $((f, X), (f, X))$ where $(f, X) \in \mathcal{S}$. The extraction lemma allows us to extract a set of function $\mathcal{H}$ of size at least $2^{2^{(1-\epsilon)n}}$ such that for all distinct $f, g \in \mathcal{H}$, there exists an $X : (f, X), (g, X) \in \mathcal{S}$, and $f(X) \neq g(X)$. Then the boosting lemma can use the set $\mathcal{H}$ and the protocol Π to solve $\mathrm{NEQ}_{\mathcal{H}}$ with a privately non-deterministic communication protocol, thus the protocol Π will need another (approximate) $\log \log |\mathcal{H}|$ bits communication. To proceed, we need following definition which treats any subset $\mathcal{Z} \subseteq \mathcal{F} \times \mathcal{X}$ as a bipartite graph.

Definition 3.1. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$ and $\mathcal{X}$ be the set $\{0,1\}^{m \times n}$. Given a set $\mathcal{Z} \subseteq \mathcal{F} \times \mathcal{X}$, we define the domain graph $\Gamma_{\mathcal{Z}}$ to be a bipartite graph $(\mathcal{U}_{\mathcal{Z}}, \mathcal{V}_{\mathcal{Z}}, \mathcal{E}_{\mathcal{Z}})$, such that $\mathcal{U}_{\mathcal{Z}} = \{f \mid (f, X) \in \mathcal{Z}\}$, $\mathcal{V}_{\mathcal{Z}} = \{X \mid (f, X) \in \mathcal{Z}\}$, and $(f, X) \in \mathcal{E}_{\mathcal{Z}} \iff (f, X) \in \mathcal{Z}$. Furthermore, for every $f \in \mathcal{U}_{\mathcal{Z}}$, denote $\{X \mid X \in \mathcal{X}, (f, X) \in \mathcal{Z}\}$ by $\mathcal{X}_{\mathcal{Z},f}$.

Now we prove the boosting lemma, its idea is similar to that in [MS21], we adapt their idea to our case and present a more detailed proof.

Lemma 3.2 (The boosting lemma)**.** *Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$, $\mathcal{X}$ be the set $\{0,1\}^{m \times n}$ and Π be a partially half-duplex communication protocol solving $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$. Assume the protocol Π has spent $s \leq m$ rounds communication, let $\tau \in \{0,1\}^s$ be a partial transcript, there is a set $\mathcal{S} \subseteq \mathcal{F} \times \mathcal{X}$ such that every input from $\{((f, X), (f, X)) \mid (f, X) \in \mathcal{S}\}$ is consistent with the transcript τ. Let $\Gamma_{\mathcal{S}} = (\mathcal{U}_{\mathcal{S}}, \mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$ be the domain graph of $\mathcal{S}$, if there is a set $\mathcal{H} \subseteq \mathcal{U}_{\mathcal{S}}$ such that for all distinct $f, g \in \mathcal{H}$, there exists an $X : (f, X), (g, X) \in \mathcal{S}$, and $f(X) \neq g(X)$, let d be the depth of Π, assume $d - s \leq n$, then $d \geq s + \log \log \mathcal{H} - \log m - \log n - 6$.*

Proof. Let $\mathcal{S} \subseteq \mathcal{F} \times \mathcal{X}$ be the set such that every input from $\{((f, X), (f, X)) \mid (f, X) \in \mathcal{S}\}$ is consistent with the transcript $\tau \in \{0,1\}^s$, we will show how to use the protocol to solve $\mathrm{NEQ}_{\mathcal{H}}$ with a privately non-deterministic communication protocol. Given $f, g \in \mathcal{H}$, Alice and Bob check one of following three conditions to be true to make sure $f \neq g$. If $f = g$, all these conditions are false.

- The first condition is there exist $X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}$ such that $((f, X), (f, X)), ((g, Y), (g, Y))$ are consistent with two distinct state pairs at depth d respectively, meanwhile both two distinct state pairs are consistent with the transcript τ.
- The second condition is there exist $X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}$ such that $f(X) \neq g(Y)$ and the residual protocol performs at least one non-classical round to solve $((f, X), (g, Y))$.

- Finally, the third condition is there exist $X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}$ such that $f(X) \neq g(Y)$ and the residual protocol solves $((f,X),(g,Y))$ with only classical rounds and outputs $\perp$.

Now we give a detailed description of the privately non-deterministic communication protocol to solve $\mathrm{NEQ}_{\mathcal{H}}$. When Alice gets a function $f \in \mathcal{H}$ and Bob gets a function $g \in \mathcal{H}$, at first Alice guesses one condition out of the three and tells Bob with 2 bits communication which condition they are going to verify, then they verify that condition as follows.

- For the first condition, Alice guesses an $X \in \mathcal{X}_{\mathcal{S},f}$ then Alice simulates the protocol Π on input $((f,X),(f,X))$ and obtains a sequence σ of classical action pairs which is consistent with τ. Then Alice guesses an index $i \in [s]$; Bob guesses a $Y \in \mathcal{X}_{\mathcal{S},g}$, then simulates the protocol Π on input $((g,Y),(g,Y))$ and obtains a sequence σ' of classical action pairs which is consistent with τ. Alice sends i to Bob and uses another 1 bit to tell Bob who sends in i-th round of σ. If the i-th round of σ is consistent with the i-th round of σ', Bob replies Alice with 0. Otherwise Bob sends 1 to Alice. To check the first condition requires at most $\log m + 2$ bits communication.

- For the second condition, Alice guesses an $X \in \mathcal{X}_{\mathcal{S},f}$, then simulates the protocol Π on input $((f,X),(f,X))$ and obtains a sequence σ of classical action pairs which is consistent with τ, let (u,v) be the state pair the protocol reaches. Bob guesses a $Y \in \mathcal{X}_{\mathcal{S},g}$, then simulates the protocol Π on input $((g,Y),(g,Y))$ and obtains a sequence σ' of classical action pairs which is consistent with τ, let (u',v') be the state pair the protocol reaches. Now Alice guesses a number $s' \in [d-s]$, a string $\tau' \in \{0,1\}^{s'}$, a coordinate $i \in [m]$ and two bits $\mathtt{a} \in \{\mathtt{receive}, \mathtt{send}\}, \mathtt{b} \in \{0,1\}$, then sends all $s', \tau', i, \mathtt{a}, \mathtt{b}$ to Bob, and they verify following to be true.

  - $\mathtt{b} = f(X)_i \neq g(Y)_i = 1 - \mathtt{b}$.
  - Alice simulates the protocol from node u in the tree Π_A according to the string τ', that is each bit involved in each action must be consistent with the corresponding bit in τ'. Similarly, Bob simulates the protocol from node v' in the tree Π_B according to the string τ'. After s' rounds, Alice and Bob verify the actions they take in next round are the same as the bit $\mathtt{a}$ indicates: either both receive or both send.

  After all that, Alice and Bob use two bits communication to tell each other the results. The second condition requires at most $d - s + \log m + \log n + 4$ bits communication.

- For the third condition, similarly, Alice guesses an $X \in \mathcal{X}_{\mathcal{S},f}$, then simulates the protocol Π on input $((f,X),(f,X))$ and obtains a sequence σ of classical action pairs which is consistent with τ, let (u,v) be the state pair the protocol reaches. Bob guesses a $Y \in \mathcal{X}_{\mathcal{S},g}$, then simulates the protocol Π on input $((g,Y),(g,Y))$ and obtains a sequence σ' of classical action pairs which is consistent with τ, let (u',v') be the state pair the protocol reaches. Now Alice guesses a string $\tau' \in \{0,1\}^{d-s}$, a coordinate $i \in [m]$ and a bit $\mathtt{b} \in \{0,1\}$, then sends all $\tau', i, \mathtt{b}$ to Bob, and they verify following to be true.

– $\mathtt{b} = f(X)_i \neq g(Y)_i = 1 - \mathtt{b}$.

– Alice simulates the protocol from node u in the tree Π_A according to the string τ' meanwhile Bob simulates the protocol from node v' in the tree Π_B according to the string τ'. After s' rounds, Alice and Bob verify they both reach leaves labeled with $\perp$.

After all that, Alice and Bob use two bits communication to tell each other the results. The third condition requires $d - s + \log m + 3$ bits communication.

Now we show this privately non-deterministic protocol is correct. Suppose that $f = g$. Then neither of three conditions could be true. Since $f = g$ the protocol behaves as a classical one, any transcript determines who sends in each round. Now the transcript τ is fixed already, the sequence of action pairs is the same for every $((f, X), (f, X)), X \in \mathcal{X}_{\mathcal{S},f}$, thus the first condition is false. By the definition of partially half-duplex protocol and $f = g$, the second condition is also false. For every input $((f, X), (f, Y)), X, Y \in \mathcal{X}_{\mathcal{S},f}, f(X) \neq f(Y)$, the protocol Π should output (i, j) such that $X_{i,j} \neq Y_{i,j}$ rather than $\perp$, it means the third condition also fails.

Suppose that $f \neq g$. If the first or the second condition is true, then we have $f \neq g$ already. If this is not the case, the third condition must be true. Now since the first condition is false, that is for every $X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}$, the protocol takes the same sequence of classical action pairs upon inputs $((f, X), (f, X)), ((g, Y), (g, Y))$ and $((f, X), (f, X)), ((g, Y), (g, Y))$ are consistent with the same classical state pair (u, v). By the rectangle property of classical state pair of Fact 2.28, for every $X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}$, $((f, X), (g, Y))$ is also consistent with (u, v). Let $\mathcal{R}_{f,g}$ be the set $\{((f, X), (g, Y)) \mid X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}\}$, this means every input in $\mathcal{R}_{f,g}$ will be solved correctly by the residual protocol starting at (u, v). Let $\mathcal{R}'_{f,g}$ be the set $\{((f, X), (g, Y)) \mid X \in \mathcal{X}_{\mathcal{S},f}, Y \in \mathcal{X}_{\mathcal{S},g}, f(X) \neq g(Y)\}$ and since for every f, g there exists an $X^\star$ such that $f(X^\star) \neq g(X^\star)$, $\mathcal{R}'_{f,g}$ is not empty. When the second condition is also false, it means the residual protocol solves every input from $\mathcal{R}'_{f,g}$ correctly with only classical rounds. By the definition of $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$, to correctly solve $((f, X^\star), (g, X^\star))$ Alice and Bob must reach leaves labeled with $\perp$ as required.

The total number of bits communicated in the privately non-deterministic protocol is at most $d - s + \log m + \log n + 6$. By Fact 2.9, $d - s + \log m + \log n + 6 \geq \log\log|\mathcal{H}|$, thus $d \geq s + \log\log|\mathcal{H}| - \log m - \log n - 6$. □

Remark 3.3. Note that the string τ' is necessary, Alice and Bob use the common string τ' to make sure in every classical round the bits in their actions are consistent. Without the common string, there may be illegal action pairs.

Lemma 3.4 (The extraction lemma)**.** *Let m, n be integers such that $m \geq 1, n > 2\log m + 2$. Let $\epsilon \in (\frac{\log m+2}{n}, 1 - \frac{\log m}{n})$ be a parameter and c, t be integers satisfying $c \geq \frac{2m+\log m}{\epsilon n - \log m - 2}, t \geq c + 4$. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$ such that $|\mathcal{F}| \geq 2^{-2^{(1-\epsilon)n}} \cdot 2^{2^n}$. Let $\mathcal{X}$ be the set $\{0,1\}^{m \times n}$. Let $\mathcal{S} \subseteq \mathcal{F} \times \mathcal{X}$ be a subset such that $|\mathcal{S}| \geq 2^{t-m} \cdot |\mathcal{F}| \cdot |\mathcal{X}|$, and let $\Gamma_{\mathcal{S}} = (\mathcal{U}_{\mathcal{S}}, \mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$ be the domain graph of $\mathcal{S}$, for every $f \in \mathcal{U}_{\mathcal{S}}$, $|\mathcal{X}_{\mathcal{S},f}| \geq 2^{t-m} \cdot |\mathcal{X}|$. Then there is a set $\mathcal{H} \subseteq \mathcal{U}_{\mathcal{S}}$ of size at least $2^{2^{(1-\epsilon)n}}$ such that for all distinct $f, g \in \mathcal{H}$, there exists an $X : (f, X), (g, X) \in \mathcal{S}$, and $f(X) \neq g(X)$.*

Proof. We extract the $\mathcal{H}$ from $\mathcal{U}_{\mathcal{S}}$ by constructing a tree $T(\mathcal{S}')$ rooted with $\mathcal{S}' \subseteq \mathcal{S}$ such that

- in the tree, each node z is associated with a subset $\mathcal{Z} \subseteq \mathcal{S}$ which viewed as a domain graph $\Gamma_{\mathcal{Z}} = (\mathcal{U}_{\mathcal{Z}}, \mathcal{V}_{\mathcal{Z}}, \mathcal{E}_{\mathcal{Z}})$ and if z is not a leaf, internal node z is also labeled with an $X_{\mathcal{Z}} \in \mathcal{V}_{\mathcal{Z}}$. Sometimes, to emphasize they are associated with node z, we also denote them with subscript z such as $\mathcal{U}_z, \mathcal{V}_z, \mathcal{E}_z$ and X_z.
- For every two distinct leaves ℓ_1, ℓ_2, we have $\mathcal{U}_{\ell_1} \cap \mathcal{U}_{\ell_2} = \emptyset$. Let node v be the lowest common ancestor of these two leaves and v is labeled with X, then for all $f \in \mathcal{U}_{\ell_1}, g \in \mathcal{U}_{\ell_2}$, we have $(f, X), (g, X) \in \mathcal{S}$ and $f(X) \neq g(X)$.

After the tree $T(\mathcal{S}')$ is constructed, the set $\mathcal{H}$ is obtained by taking exact one function from each leaf. Given two distinct elements $f, g \in \mathcal{H}$ such that $f \in \mathcal{U}_{\ell_1}, g \in \mathcal{U}_{\ell_2}$, since $\mathcal{U}_{\ell_1} \cap \mathcal{U}_{\ell_2} = \emptyset$, $f \neq g$. Moreover, let X be the label of the least common ancestor of leaves ℓ_1 and ℓ_2, we have $(f, X), (g, X) \in \mathcal{S}$ and $f(X) \neq g(X)$ as required.

Before construction of the tree, we need to introduce some helpful notations. A trace Ψ is a subset of $\{0,1\}^n$. Particularly, we can view every $X \in \{0,1\}^{m \times n}$ as a trace, for convenience, when the context is clear, we abuse the notation and treat X as a trace set of all its distinct rows $\{x \mid \exists i, x = X_i\}$. Let z be a node at depth d in the tree, from root to node z, its ancestors are $z_0, z_1, \ldots, z_{d-1}$. For every $i \in \{0, 1, \ldots, d-1\}$, z_i is labeled with X_{z_i}, treat every X_{z_i} as a trace, we define $\Psi(z) = \bigcup_{i=0}^{d-1} X_{z_i}$.

The purpose of trace is to record a set of inputs $\Psi(z)$ and all functions in $\mathcal{U}_z$ take the same value given any input in $\Psi(z)$. In another word, all functions in $\mathcal{U}_z$ are restricted to set $\{0,1\}^n \setminus \Psi(z)$. Therefore, the number of functions in $\mathcal{U}_z$ is up bounded by $2^{2^n - |\Psi(z)|}$. For our purpose, we need the size of $\mathcal{U}_z$ to be as small as possible thus the size of trace $\Psi(z)$ to be as large as possible. Given a trace $\Psi(z)$ for some node z, we want to choose an X for node z such that $|\Psi(z) \cup X| - |\Psi(z)| \geq m - c$. The problem is that we can not choose any X freely, to make any remaining X is good for our purpose, we have to remove all bad Xs in advance. Let

$$\Phi(z) = \{X \mid |\Psi(z) \cup X| - |\Psi(z)| < m - c\} = \{X \mid |X \setminus \Psi(z)| < m - c\}$$

be the set of bad Xs for node z, the parent of node z will take the responsibility to remove all the bad Xs against node z, then any X in $\mathcal{V}_z$ is good for z to choose.

Now we show how to construct the tree recursively and lower bound the size of $\mathcal{H}$ which is exactly the number of leaves in the tree. Set $h = 2^{\lceil (1-\epsilon)n \rceil}$. Let z be some node of $T(\mathcal{S})$ at depth $d \leq h$, the node z is associated with a subset $\mathcal{Z} \subseteq \mathcal{S}$. Initially, if z is the root of $T(\mathcal{S}')$, the trace $\Psi(z)$ at root z is the empty set, the set of bad Xs for z is $\Phi(z) = \{X \mid |X \setminus \Psi(z)| < m - c\} = \{X \mid |X| < m - c\}$ where $|X|$ is the number of distinct rows in X, recall that we treat X as a trace of its rows. Since root z has no parent, we have to remove all bad Xs for root z in advance and z is associated with a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that

$$\mathcal{S}' = \{(f, X) \mid (f, X) \in \mathcal{S}, X \notin \{X \mid |X| < m - c\}\}.$$

Let $\Gamma_{\mathcal{Z}} = (\mathcal{U}_{\mathcal{Z}}, \mathcal{V}_{\mathcal{Z}}, \mathcal{E}_{\mathcal{Z}})$ be the domain graph of $\mathcal{Z}$. If z is at depth h, then z is a leaf, otherwise, we recursively construct a tree $T(\mathcal{Z})$ rooted at z by attaching a set of sub-trees to the node z. Now since z's parent has removed all bad Xs against z, all Xs in $\mathcal{V}_z$ are

good. Let X_z be some vertex of maximal degree in $\mathcal{V}_z$, then the trace of each z's children is $\Psi(z) \cup X_z$, now we want to remove all bad Xs against z's children, and the set of bad X against z's children is following set

$$\Phi'(z) = \{X \mid |X \setminus (\Psi(z) \cup X_z)| < m - c\}.$$

After choosing X_z and removing all bad Xs against z's children, let

$$\mathcal{Z}' = \{(f, X) \mid (f, X_z) \in \mathcal{Z}, (f, X) \in \mathcal{Z}, X \notin \Phi'(z)\}.$$

$\mathcal{Z}'$ is obtained from $\mathcal{Z}$ as follows. At first, remove all fs in $\mathcal{U}_\mathcal{Z}$ such that (f, X_z) is not in $\mathcal{Z}$, then for the remaining fs, remove every (f, X) such that $X \in \Phi'(z)$ which is bad against z's children.

Now for every $a \in \{0,1\}^m$, let $\mathcal{Z}'_a = \{(f, X) \mid (f, X) \in \mathcal{Z}', f(X_z) = a\}$. If $\mathcal{Z}'_a$ is not empty, there is a subtree $T(\mathcal{Z}'_a)$ attached to the node z. Given two distinct subtrees $T(\mathcal{Z}'_{a1}), T(\mathcal{Z}'_{a_2})$, let $\Gamma_{\mathcal{Z}'_{a_1}}, \Gamma_{\mathcal{Z}'_{a_2}}$ be domain graphs of $\mathcal{Z}'_{a_1}, \mathcal{Z}'_{a_2}$ respectively, then $\mathcal{U}_{\mathcal{Z}'_{a_1}} \cap \mathcal{U}_{\mathcal{Z}'_{a_2}} = \emptyset$, since for every $f \in \mathcal{U}_{\mathcal{Z}'_{a_1}}, g \in \mathcal{U}_{\mathcal{Z}'_{a_2}}$, $f(X_z) = a_1 \neq a_2 = g(X_z)$. Thus recursively, for every two nodes z_1, z_2 at the same depth, $\mathcal{U}_{z_1} \cap \mathcal{U}_{z_2} = \emptyset$, and let X be the label of the two nodes z_1, z_2' lowest common ancestor, for all $f \in \mathcal{U}_{z_1}, g \in \mathcal{U}_{z_2}, f(X) \neq g(X)$. Finally, for every two leaves with $\mathcal{U}_{\ell_1}$ and $\mathcal{U}_{\ell_2}$, this is also true.

Now we are ready to lower bound the number of leaves in $T(\mathcal{S})$ by lower bounding the number of nodes at depth d. The idea is to show the total number functions in these nodes is large and the number of function in each single node is small. Since for every two nodes z_1, z_2 at the same depth, $\mathcal{U}_{z_1} \cap \mathcal{U}_{z_2} = \emptyset$, there must be many such nodes.

Let z be some node of the tree $T(\mathcal{S})$ at depth $d \leq h$ labeled with X_z corresponding to a root node of a subtree $T(\mathcal{Z})$ for some $\mathcal{Z} \subseteq \mathcal{S}$. Let $\Gamma_\mathcal{Z} = (\mathcal{U}_\mathcal{Z}, \mathcal{V}_\mathcal{Z}, \mathcal{E}_\mathcal{Z})$ be the domain graph of $\mathcal{Z}$. Let $T(\mathcal{Z}_{a_1}), \ldots, T(\mathcal{Z}_{a_k})$ be the subtrees attached to z and $z_{a_1}, \ldots, z_{a_k}$ be the roots of these subtrees respectively. Note that for every $i \in [k]$, trace $\Psi(z_{a_i}) = \Psi(z) \cup X_z$ and $\Phi(z_{a_i}) = \Phi'(z) = \{X \mid |X \setminus (\Psi(z) \cup X_z)| < m - c\}$. Recall that $\mathcal{U}_{\mathcal{Z}_{a_i}} \cap \mathcal{U}_{\mathcal{Z}_{a_j}} = \emptyset$ for all $i \neq j$, let $\Gamma_{\mathcal{Z}'} = (\mathcal{U}_{\mathcal{Z}'}, \mathcal{V}_{\mathcal{Z}'}, \mathcal{E}_{\mathcal{Z}'})$ be the domain graph of $\mathcal{Z}'$, then $\mathcal{U}_{\mathcal{Z}_{a_1}} \cup \ldots \cup \mathcal{U}_{\mathcal{Z}_{a_k}} = \mathcal{U}_{\mathcal{Z}'}$. Now let

$$\mathcal{Z}^\star = \{(f, X) \mid (f, X_z) \in \mathcal{Z}, (f, X) \in \mathcal{Z}\}$$

where $\mathcal{Z}^\star$ is obtained from $\mathcal{Z}$ by collecting all fs in $\mathcal{U}_\mathcal{Z}$ such that (f, X_z) is in $\mathcal{Z}$, then $\mathcal{U}_{\mathcal{Z}^\star} = \mathcal{U}_{\mathcal{Z}'}$. To see why this is true, we have to lower bound the degree of every $f \in \mathcal{U}_{\mathcal{Z}'}$ in the domain graph $\Gamma_{\mathcal{Z}'}$.

Firstly, we show for every node z associated with some set $\mathcal{Z} \subseteq \mathcal{F} \times \mathcal{X}$, $\mathcal{X}_{\mathcal{Z},f} = \mathcal{X}_{\mathcal{S},f} \setminus \Phi(z)$ by induction on the depth of the node. Recall that when z is the root node, z is associated with $\mathcal{S}' = \{(f, X) \mid (f, X) \in \mathcal{S}, X \notin \Phi(z)\}$, that is for every $f \in \mathcal{U}_{\mathcal{S}'}$, $\mathcal{X}_{\mathcal{S}',f} = \mathcal{X}_{\mathcal{S},f} \setminus \Phi(z)$. Assume z is node which is associated with $\mathcal{Z}$, for every $f \in \mathcal{U}_\mathcal{Z}$, $\mathcal{X}_{\mathcal{Z},f} = \mathcal{X}_{\mathcal{S},f} \setminus \Phi(z)$. Let z_a be a child of z and z_a is associated with set $\mathcal{Z}_a$, then for every $f \in \mathcal{U}_{\mathcal{Z}_a}$, we have

$$\begin{aligned}
\mathcal{X}_{\mathcal{Z}_a,f} &= \mathcal{X}_{\mathcal{Z},f} \setminus \Phi(z_a) \\
&= (\mathcal{X}_{\mathcal{S},f} \setminus \Phi(z)) \setminus \Phi(z_a), \text{ by induction hypothesis} \\
&= \mathcal{X}_{\mathcal{S},f} \setminus \Phi(z_a) \text{ , since } \Phi(z) \subseteq \Phi(z_a)
\end{aligned}$$

as required. Now for every $f \in \mathcal{U}_{\mathcal{Z}}$, we have $|\mathcal{X}_{\mathcal{Z},f}| \geq |\mathcal{X}_{\mathcal{S},f}| - |\Phi(z)|$. To proceed, we have to up bound $|\Phi(z)|$ as follows.

$$
\begin{aligned}
|\Phi(z)| &\leq |\{X \mid |X \setminus \Psi(z)| < m - c\}| = \sum_{i=0}^{m-c-1} |\{X \mid |X \setminus \Psi(z)| = i\}| \\
&\leq \sum_{i=0}^{m-c-1} \binom{2^n - |\Psi(z)|}{i} \cdot \binom{m}{i} \cdot (i + |\Psi(z)|)^{m-i} \\
&\leq \sum_{i=0}^{m-c-1} 2^{ni} \cdot 2^m \cdot (m + md)^{m-i}, \text{since } |\Psi(z)| \leq md, i \leq m \\
&\leq \sum_{i=0}^{m-c-1} 2^{ni} \cdot 2^m \cdot 2^{((1-\epsilon)n + \log m + 2)(m-i)}, \text{since } d \leq 2^{\lceil (1-\epsilon)n \rceil} \\
&= \sum_{i=0}^{m-c-1} 2^{mn} \cdot 2^{(-\epsilon n + \log m + 2)(m-i)} \cdot 2^m \\
&\leq \sum_{i=0}^{m-c-1} 2^{mn} \cdot 2^{(-\epsilon n + \log m + 2)(c+1)+m}, \text{since } -\epsilon n + \log m + 2 < 0, m - i \geq c + 1 \\
&\leq 2^{mn} \cdot 2^{(-\epsilon n + \log m + 2)(c+1)+m+\log m} \\
&\leq 2^{-m} \cdot 2^{mn}, \text{since } c \geq \frac{2m + \log m}{\epsilon n - \log m - 2}.
\end{aligned}
$$

Thus, for every node z associated with $\mathcal{Z}$, for every $f \in \mathcal{U}_{\mathcal{Z}}$, we have $|\mathcal{X}_{\mathcal{Z},f}| \geq 2^{t-m} \cdot 2^{mn} - 2^{-m} \cdot 2^{mn} \geq 2^{t-m-1} \cdot 2^{mn} \gg 0$. Now we show $\mathcal{U}_{\mathcal{Z}^\star} = \mathcal{U}_{\mathcal{Z}'}$ where $\mathcal{Z}'$ is obtained from $\mathcal{Z}^\star$ by removing bad Xs, after the removal, for every f in $\mathcal{U}_{\mathcal{Z}^\star}$, $\mathcal{X}_{\mathcal{Z}',f}$ is still not empty and f remains in $\mathcal{U}_{\mathcal{Z}'}$. More formally, for every f in $\mathcal{U}_{\mathcal{Z}^\star}$, $\mathcal{X}_{\mathcal{Z}',f} = \mathcal{X}_{\mathcal{Z}^\star,f} \setminus \Phi'(z) = \mathcal{X}_{\mathcal{Z},f} \setminus \Phi'(z) = \mathcal{X}_{\mathcal{S},f} \setminus \Phi'(z)$. Let z_a be some child of z, recall that $\Phi'(z) = \Phi(z_a)$, thus for every f in $\mathcal{U}_{\mathcal{Z}^\star}$, $\mathcal{X}_{\mathcal{Z}',f} = \mathcal{X}_{\mathcal{S},f} \setminus \Phi(z_a)$. Similarly, $|\mathcal{X}_{\mathcal{Z}',f}| \geq |\mathcal{X}_{\mathcal{S},f}| - |\Phi(z_a)| \geq 2^{t-m-1} \cdot 2^{mn} \gg 0$ since $|\Phi(z_a)|$ is also no larger than $2^{-m} \cdot 2^{mn}$. Particularly, we have $\mathcal{U}_{\mathcal{S}} = \mathcal{U}_{\mathcal{S}'}$.

Given that X_z is a vertex of maximal degree in $\mathcal{V}_{\mathcal{Z}}$ and $|\mathcal{V}_{\mathcal{Z}}| \leq |\mathcal{X}| = 2^{mn}$, the number of functions in the subtrees can be lower bounded as follows

$$
\begin{aligned}
|\mathcal{U}_{Z_{a_1}} \cup \ldots \cup \mathcal{U}_{Z_{a_k}}| = |\mathcal{U}_{\mathcal{Z}'}| = |\mathcal{U}_{\mathcal{Z}^\star}| &\geq \frac{|\mathcal{E}_{\mathcal{Z}}|}{|\mathcal{V}_{\mathcal{Z}}|} \geq \frac{|\mathcal{U}_{\mathcal{Z}}| \cdot \min_{f \in \mathcal{U}_{\mathcal{Z}}} |\mathcal{X}_{\mathcal{Z},f}|}{2^{mn}} \\
&\geq \frac{|\mathcal{U}_{\mathcal{Z}}| \cdot 2^{t-m-1} \cdot 2^{mn}}{2^{mn}} \\
&= \frac{|\mathcal{U}_{\mathcal{Z}}|}{2^{m+1-t}}.
\end{aligned}
$$

Thus by induction the total number of functions that appear in the nodes at depth d is at least

$$\frac{|\mathcal{U}_{\mathcal{S}}|}{2^{(m+1-t)d}}.$$

Now we are ready to lower bound the number of nodes at some depth d. Let z be a node, then for every $f \in \mathcal{U}_z, x \in \Psi(z)$, $f(x)$ is the same, so the number of distinct functions in

$\mathcal{U}_z$ is at most $2^{2^n}/2^{|\Psi(z)|} \le 2^{2^n-(m-c)d}$. The number of nodes at depth d is at least the total number of functions at depth d divided by the upper bound on the number of functions in one node, that is

$$\frac{|\mathcal{U}_{\mathcal{S}}|}{2^{(m+1-t)d} \cdot 2^{2^n-(m-c)d}} = \frac{2^{(t-c-1)d}|\mathcal{U}_{\mathcal{S}}|}{2^{2^n}}.$$

Since by assumption $|\mathcal{S}| \ge 2^{t-m} \cdot |\mathcal{F}| \cdot |\mathcal{X}|$ and $|\mathcal{F}| \ge 2^{-2^{(1-\epsilon)n}} \cdot 2^{2^n} \ge 2^{-h} \cdot 2^{2^n}$, the size of $\mathcal{U}_{\mathcal{S}}$ is at least

$$\frac{|\mathcal{S}|}{|\mathcal{X}|} \ge \frac{2^{t-m} \cdot |\mathcal{F}| \cdot |\mathcal{X}|}{|\mathcal{X}|} \ge 2^{t-m} \cdot 2^{-h} \cdot 2^{2^n},$$

the number of leaves at depth $h = 2^{\lceil (1-\epsilon)n \rceil}$ is at least

$$\begin{aligned}\frac{2^{(t-c-1)h} \cdot 2^{t-m} \cdot 2^{-h} \cdot 2^{2^n}}{2^{2^n}} &\ge 2^{(t-c-3)h}, \text{since } \epsilon < 1 - \frac{\log m}{n}, m \le 2^{(1-\epsilon)n} \\ &\ge 2^h, \text{since } t \ge c+4 \\ &= 2^{2^{\lceil (1-\epsilon)n \rceil}}\end{aligned}$$

as required. □

Theorem 3.5 (The boosting theorem)**.** *Let m, n be integers such that $m \ge 1, n > 2\log m + 2$. Let $\epsilon \in (\frac{\log m+2}{n}, 1 - \frac{\log m}{n})$ be a parameter and c, t, s be integers satisfying $c \ge \frac{2m+\log m}{\epsilon n - \log m - 2}, t \ge c+4, s \le m-t-1$. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$ such that $|\mathcal{F}| \ge 2^{-2^{(1-\epsilon)n}} \cdot 2^{2^n}$, let $\mathcal{X}$ be the set $\{0,1\}^{m \times n}$, if $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \ge s$, $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \ge s + (1-\epsilon)n - \log m - \log n - 6$.*

Proof. Given any partially half-duplex protocol Π for $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$, let d be the depth of protocol Π, since $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \ge s$, $d \ge s$. Let $\mathcal{D} = \{((f,X),(f,X)) \mid f \in \mathcal{F}, X \in \mathcal{X}\}$, by Fact 2.29, there is a transcript $\tau \in \{0,1\}^s$ and a subset of inputs $\mathcal{D}' \subseteq \mathcal{D}$ such that every input in $\mathcal{D}'$ is consistent with τ and $|\mathcal{D}'| \ge |\mathcal{D}|/2^s$, let $\mathcal{T} = \{(f,X) \mid ((f,X),(f,X)) \in \mathcal{D}'\}$ then $|\mathcal{T}| \ge 2^{-s} \cdot |\mathcal{F}| \cdot |\mathcal{X}| \ge 2^{t+1-m} \cdot |\mathcal{F}| \cdot |\mathcal{X}|$. Removing every f such that $|\mathcal{X}_{\mathcal{T},f}| < 2^{t-m} \cdot |\mathcal{X}|$ in $\mathcal{T}$, and obtain $\mathcal{S} = \{(f,X) \mid (f,X) \in \mathcal{T}, |\mathcal{X}_{\mathcal{T},f}| \ge 2^{t-m} \cdot |\mathcal{X}|\}$, then $|\mathcal{S}| \ge |\mathcal{T}| - |\mathcal{F}| \cdot 2^{t-m} \cdot |\mathcal{X}| \ge 2^{t-m} \cdot |\mathcal{F}| \cdot |\mathcal{X}|$, and let $\Gamma_{\mathcal{S}} = (\mathcal{U}_{\mathcal{S}}, \mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$ be the domain graph of $\mathcal{S}$, for every $f \in \mathcal{U}_{\mathcal{S}}$, $|\mathcal{X}_{\mathcal{S},f}| \ge 2^{t-m} \cdot |\mathcal{X}|$.

Apply Lemma 3.4 with $\mathcal{S}$ and parameters m, n, ϵ, c, t, then there is a set $\mathcal{H} \subseteq \mathcal{U}_{\mathcal{S}}$ of size at least $2^{2^{(1-\epsilon)n}}$ such that for all distinct $f, g \in \mathcal{H}$, there exists an $X : (f,X), (g,X) \in \mathcal{S}$, and $f(X) \ne g(X)$. Apply Lemma 3.2 with the transcript τ and the set $\mathcal{H}$, we have $d \ge s + (1-\epsilon)n - \log m - \log n - 6$ as required. □

Now we prove Theorem 1.6 rephrased as follows.

Theorem 3.6. *Let m, n be integers such that $m \ge 1, n > 2\log m + \log n + 9$. Let $\epsilon \in (\frac{\log m+2}{n}, 1 - \frac{\log m + \log n + 7}{n})$. Let $\mathcal{F}$ be a set of functions $f : \{0,1\}^n \to \{0,1\}$ such that $|\mathcal{F}| \ge 2^{-2^{(1-\epsilon)n}} \cdot 2^{2^n}$, then*

$$\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \ge m + n - \left\lceil \frac{2m + \log m}{\epsilon n - \log m - 2} \right\rceil - \epsilon n - \log m - \log n - 11.$$

Furthermore, let $m = \omega(\log^2 n), n = \omega(\sqrt{m})$, $\epsilon = \frac{\sqrt{m}}{n}$, *we have*

$$\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \geq m+n-\left\lceil \frac{2m+\log m}{\sqrt{m}-\log m-2} \right\rceil - \sqrt{m} - \log m - \log n - 11 = m+n-O(\sqrt{m}).$$

Proof. In the beginning, set $c = \lceil \frac{2m+\log m}{\epsilon n - \log m - 2} \rceil, t = c+4, s = 0$ where s is current known lower bound of $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}})$. Repeatedly applying Theorem 3.5, after each application, we have $s \leftarrow s + (1-\epsilon)n - \log m - \log n - 6$. Since $\epsilon \in (\frac{\log m+2}{n}, 1 - \frac{\log m + \log n + 7}{n})$, $(1-\epsilon)n - \log m - \log n - 6 \geq 1$, that is every application will increase the complexity at least one. This repetition will not end until $s = m - t - 1$, when $s = m - t - 1$, apply Theorem 3.5 for the last time, and obtain

$$\begin{aligned} s &\geq m - t - 1 + (1-\epsilon)n - \log m - \log n - 6 \\ &= m - \left\lceil \frac{2m+\log m}{\epsilon n - \log m - 2} \right\rceil - 4 - 1 + (1-\epsilon)n - \log m - \log n - 6 \\ &= m + n - \left\lceil \frac{2m+\log m}{\epsilon n - \log m - 2} \right\rceil - \epsilon n - \log m - \log n - 11 \end{aligned}$$

as required. □

4 A Composition Theorem of a Universal Relation and Most Functions

In this section we prove when m, n are in proper range, for most functions $f : \{0,1\}^n \to \{0,1\}$, the communication complexity of $\mathrm{U}_m \diamond \mathrm{KW}_f$ is at least $m + n - O(\sqrt{m})$. At first, we need the following lemma which transforms the complexity of $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$ in the partially half-duplex model to the complexity of $\mathrm{U}_m \diamond \mathrm{KW}_f$ for some function $f \in \mathcal{F}$ in the standard model of communication. The lemma is proved with the same idea in [MS21].

Lemma 4.1. *Let* $\mathcal{F}$ *be a set of functions* $f : \{0,1\}^n \to \{0,1\}$, *then*

$$\max_{f \in \mathcal{F}} \mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq \mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) - \log mn - 2.$$

Proof. Let $d = \max_{f \in \mathcal{F}} \mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f)$. For every $f \in \mathcal{F}$, Alice and Bob hold the same optimal standard protocol Π_f which depth is no larger than d. Now we can leverage this to construct a partially half-duplex protocol for $\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}$. Given input (f, X), Alice simulates the protocol Π_f on input X. Similarly, given input (g, Y), Bob simulates the protocol Π_g on Y. When Alice performs t rounds and reaches some leaf labeled with (i, j) in protocol Π_f, if $t < d$, Alice performs another $d - t$ round of sending 1. Similarly, when Bob performs t' rounds and reaches some leaf labeled with (i', j') in protocol Π_g, if $t' < d$, Bob performs another $d - t'$ round of receiving.[1] After both players spend exact d rounds, they

[1] In the proof of a similar lemma in [MS21] by Ivan Mihajlin and Alexander Smal, they ask both Alice and Bob to perform the action of receiving after reaching leaves, this is problematic, when Alice and Bob are given the same function, they also perform non-classical rounds after reaching leaves.

start to verify that Alice's answer is correct. Alice sends (i, j) and $X_{i,j}$ to Bob, Bob replies with 1 if $X_{i,j} \neq Y_{i,j}$ and 0 otherwise. Finally, they output (i, j) if Alice's answer is correct and $\perp$ otherwise.

When Alice and Bob are given the same function f, they must perform t classical rounds and reach the same leaf in protocol tree Π_f since they simulate the same protocol Π_f. In the next $d-t$ round, Alice sends 1 and Bob receives 1, after that they perform classical rounds to verify Alice's answer is correct. Thus, above protocol is indeed a correct partially half-duplex protocol and it spends $d + \log mn + 2$ bits communication. That is $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \leq d + \log mn + 2$ as required. □

Now we prove Theorem 1.5 rephrased as follows.

Theorem 4.2. *Let $m = \omega(\log^2 n), n = \omega(\sqrt{m}), \epsilon = \frac{\sqrt{m}}{n}$, there are at least $2^{2^n}(1 - 2^{-2^{(1-\epsilon)n}})$ distinct functions $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq m + n - O(\sqrt{m})$.*

Proof. In the beginning, let $\mathcal{F}$ be the set of all functions $f : \{0,1\}^n \to \{0,1\}$ and $|\mathcal{F}| = 2^{2^n}$, apply Theorem 3.6 with $\mathcal{F}$, we have $\mathsf{CC}^{\mathsf{phd}}(\mathrm{U}_m \diamond \mathrm{MUX}_{\mathcal{F}}) \geq m + n - O(\sqrt{m})$. By Lemma 4.1, there exists a function f such that $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq m + n - O(\sqrt{m}) - \log mn - 2 = m + n - O(\sqrt{m})$. Now set $\mathcal{F} \leftarrow \mathcal{F} \setminus \{f\}$, repeat this process until $|\mathcal{F}| < 2^{-2^{(1-\epsilon)n}} \cdot 2^{2^n}$, then we have found at least $2^{2^n} - 2^{-2^{(1-\epsilon)n}} \cdot 2^{2^n}$ functions $f : \{0,1\}^n \to \{0,1\}$ such that $\mathsf{CC}(\mathrm{U}_m \diamond \mathrm{KW}_f) \geq m + n - O(\sqrt{m})$. □

5 Conclusion and Discussion

Here we make some discussion about our results and point out some future directions. As mentioned before, our method can be used to obtain a similar result for function bundles. That is for most (m, n) function bundles F, $\mathsf{CC}(\mathrm{KW}_F)$ is about $m + n - O(\sqrt{m})$, this can be done with a slightly different way of restriction. We also note that our method can be applied to other related conjectures in [DM18]. Take Conjecture 9.4 in [DM18] as example, Dinur and Meir conjectured given a subset $\mathcal{X} \subseteq \{0,1\}^{m \times n}$ with density at least $2^{-(m-\tilde{O}(\sqrt{m}))}$, then the restriction of $\mathrm{U}_m \diamond \mathrm{KW}_f$ to $\mathcal{X} \times \mathcal{X}$ has communication complexity at least $\mathsf{CC}(\mathrm{KW}_f) - \tilde{O}(\sqrt{m})$. With our method, we can show that when choosing $f, \mathcal{X}$ randomly, it is true with high probability. But it is not clear whether our method is helpful in the case of 1-out-of-k problem of KW relation [DM18]. Furthermore, comparing to the optimal lower bound in the case of $\mathrm{KW}_f \diamond \mathrm{U}_n$, there still is room for improvement, thus the question is can we prove a lower bound for $\mathrm{U}_m \diamond \mathrm{KW}_f$ with poly-logarithmic additive loss. We also suspect that our result can be extended to a slightly weaker lower bound in terms of protocol size like those in [GMWW17, KM18], but we haven't fully verify it.

The next major step is to consider the composition of two multiplexors. Let $\mathcal{F}$ be the set of all functions $f : \{0,1\}^m \to \{0,1\}$, $\mathcal{G}$ be the set of all functions $g : \{0,1\}^n \to \{0,1\}$ and $\Delta = \mathcal{F} \times \mathcal{G}$. In KW relation $\mathrm{MUX}_m \diamond \mathrm{MUX}_n$, Alice gets a pair of functions $(f, g) \in \Delta$ and a Boolean matrix $X \in \{0,1\}^{m \times n}$, Bob gets a pair of functions $(f', g') \in \Delta$ and a Boolean matrix $Y \in \{0,1\}^{m \times n}$, their goal is to find an entry (i, j) such that $X_{i,j} \neq Y_{i,j}$. If $(f, g) \neq (f', g')$ or $f \diamond g(X) = f' \diamond g'(Y)$, they can also output $\perp$. We think it may be easier to prove lower bound for composition of two multiplexors than composition of a function

and a multiplexor. Comparing to the KW relation of a function, the multiplexor looks more like the universal relation. But our current way of restriction won't immediately work in the case of two multiplexors. When constructing the binary tree for the second stage, in each step downward, the number of functions in each child decreases by a factor of (at most) 2 while the total number of functions in all its children decreases by a (average) factor of approximate 2^m. Due to the fact that the composite function $f \diamond g$ takes Boolean values, the protocol can easily divide a set $\mathcal{S} \subseteq \Delta \times \mathcal{X}$ into two parts such that in each part $f \diamond g(X)$ is the same for every $(f \diamond g, X) \in \mathcal{S}$. Thus considering a square $\mathcal{S} \times \mathcal{S}$ is not helpful anymore in this case, we should consider general rectangle like those in [Mei20, Mei23] and new ideas are needed. Maybe we should try our method in the case of strong composition of two multiplexors in the first place. A less ambitious question is to show a composition theorem of a parity function and a multiplexor.

Acknowledgments

The author is grateful to Ivan Mihajlin and Alexander Smal for many detailed comments and valuable suggestions that greatly improved the presentation of this paper.

References

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

[dRMN+20] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, and Robert Robere. KRW composition theorems via lifting. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 43–49. IEEE, 2020.

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

[FMT21] Yuval Filmus, Or Meir, and Avishay Tal. Shrinkage under random projections, and cubic formula lower bounds for AC0 (extended abstract). In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 89:1–89:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[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.

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

[HIMS18] Kenneth Hoover, Russell Impagliazzo, Ivan Mihajlin, and Alexander V. Smal. Half-duplex communication complexity. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation, ISAAC 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan*, volume 123 of *LIPIcs*, pages 10:1–10:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[HW93] Johan Håstad and Avi Wigderson. Composition of the universal relation. In *ADVANCES IN COMPUTATIONAL COMPLEXITY THEORY, AMS-DIMACS*, 1993.

[IMS22] Artur Ignatiev, Ivan Mihajlin, and Alexander Smal. Super-cubic lower bound for generalized karchmer-wigderson games. In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation, ISAAC 2022, December 19-21, 2022, Seoul, Korea*, volume 248 of *LIPIcs*, pages 66:1–66:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[KM18] Sajin Koroth and Or Meir. Improved composition theorems for functions and relations. *Leibniz International Proceedings in Informatics, LIPIcs*, 116(48):1–18, 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. *Comput. Complex.*, 5(3/4):191–204, 1995.

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

[Mei20] Or Meir. Toward better depth lower bounds: Two results on the multiplexor relation. *Comput. Complex.*, 29(1):4, 2020.

[Mei23] Or Meir. Toward better depth lower bounds: A krw-like theorem for strong composition. *Electron. Colloquium Comput. Complex.*, TR23-078, 2023.

[MS21] Ivan Mihajlin and Alexander Smal. Toward better depth lower bounds: The XOR-KRW conjecture. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 38:1–38:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[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. IEEE Computer Society, 2014.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213. ACM, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il