# Lower Bounds for Linear Decision Trees via An Energy Complexity Argument

Kei Uchizawa[1] and Eiji Takimoto[2*]

[1] Graduate School of Information Sciences, Tohoku University, Aramaki Aoba-aza 6-6-05, Aoba-ku, Sendai, 980-8579, Japan.
[2] Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan.
`uchizawa@ecei.tohoku.ac.jp` `eiji@i.kyushu-u.ac.jp`

**Abstract.** A linear decision tree is a binary decision tree in which a classification rule at each internal node is defined by a linear threshold function. In this paper, we consider a linear decision tree $T$ where the weights $w_1, w_2, ..., w_n$ of each linear threshold function satisfy $\sum_i |w_i| \leq w$ for an integer $w$, and prove that if $T$ computes an $n$-variable Boolean function of large unbounded-error communication complexity (such as the Inner-Product function modulo 2), then $T$ must have size (i.e., the number of leaves) $2^{\Omega(\sqrt{n})}/w$. To obtain the lower bound, we utilize a close relationship between the size of linear decision trees and the energy complexity of threshold circuits; the energy of a threshold circuit $C$ is defined to be the maximum number of gates outputting "1," where the maximum is taken over all inputs to $C$. In addition, we consider threshold circuits of depth $\omega(1)$ and bounded energy, and provide two exponential lower bounds on the size (i.e., the number of gates) of such circuits.

## 1 Introduction

A linear decision tree is a binary decision tree in which a classification rule at each internal node is defined by a linear function so that right and left edges correspond to $\geq 0$ and $< 0$, respectively. The complexity of a linear decision tree is usually measured by the depth and the size, where the depth is the length of the longest path from the root to a leaf and the size is the number of leaves. The depth and size are reasonable measures of time and space required for the corresponding algorithm, respectively. In previous research, the depth complexity of linear decision trees is extensively studied, and lower bounds are obtained for many problems [3–5, 8, 16]. In particular, Gröger and Turán obtain a linear lower bound on the depth of the linear decision trees computing the Inner-Product function $IP_n$ [8]. However, the size complexity is less understood especially for Boolean functions. Since the depth is a lower bound on the size, the liner lower bound on the depth given in [8] yields a linear lower bound on the size. To the best of our knowledge, this is the largest known lower bound on the size of linear decision trees computing an explicit Boolean function.

* Supported by MEXT Grant-in-Aid for Scientific Research (C) No.20500001

In this paper, we restrict ourselves to the case where the weights of each linear functions are not too large, and show an exponential lower bound on the size of linear decision trees for a large class of Boolean functions including the Inner-Product function $IP_n$. More precisely, we prove that any linear decision tree computing an $n$-variable Boolean function of unbounded-error communication complexity $\Omega(n)$ has size $2^{\Omega(\sqrt{n})}/w$, provided that the linear threshold function at each internal node has a weight vector $(w_1, w_2, ..., w_n)$ such that $\sum_i |w_i| \leq w$; hence we have an exponential lower bound on the size if $w = 2^{o(\sqrt{n})}$. For the model of ternary decision trees computing a function $f : \mathbb{R}^n \to \{0, 1\}$ where each internal node gives ternary classification into "$> 0$", "$= 0$", or "$< 0$", exponential lower bounds are known even if the classification rule is defined by a high degree polynomial [2, 9]. However, they consider ternary decision trees computing a real function, and hence their results do not immediately imply lower bounds for binary linear decisions computing a Boolean function. In fact, the explicit function used to derive the exponential lower bound on the size of ternary decision trees in [9] can be computed by a binary linear decision tree of polynomial size. In the paper [5], Fleischer obtained an exponential lower bound on the size of binary linear decision trees, but they consider a real function too. For a simpler and more standard model of binary decision trees in which the classification rule is defined by a Boolean variable, Wegener derives an exponential lower bound on the size of the trees computing the Parity function [20].

To obtain our lower bound, we utilize a close relationship between the size of linear decision trees and the energy complexity of threshold circuits; a threshold circuit is a combinatorial circuit of threshold gates, and the energy of a threshold circuit $C$ is defined to be the maximum number of gates outputting "1," where the maximum is taken over all inputs to $C$. More precisely, we use the following fact given by Uchizawa *et al.* in [18]: if a function $f$ cannot be computed by any threshold circuit of size $O(l)$, energy $O(\log l)$ and weight $O(w)$, then $f$ cannot be computed by any linear decision tree of size $O(l)$ and weight $O(w)$ either. Thus, a lower bound on the size of threshold circuits of small energy and weight implies a lower bound on the size of linear decision trees. Using the unbounded-error communication complexity argument, we prove that if a threshold circuit $C$ of energy $e$ and weight $w$ computes $IP_n$, then the size $s$ of $C$ is $s = 2^{\Omega(n/e)}/w$, which suffices to provide the desired lower bound for linear decision trees. Our result appears to be the first application of the notion of energy complexity of a threshold circuit for a computational model other than a threshold circuit.

In addition, we obtain lower bounds on the size (i.e., the number of gates) of threshold circuits of depth $\omega(1)$ and bounded energy. The above lower bound $2^{\Omega(n/e)}/w$ immediately yields an exponential lower bound on the size of threshold circuits of energy $n^{o(1)}$ and weight $2^{o(n)}$. Note that this lower bound is independent of depth. We also consider the case where weights of threshold gates are unrestricted, and provide an exponential lower bound on the size of threshold circuits of depth $n^{o(1)}$ and energy $O(1)$. These two exponential lower bound are of independent interest, since they contrast with the known super-polynomial lower bounds for threshold circuits which require the depth to be a constant with further restrictions on fan-in, weight, or energy [1, 6, 7, 10–12, 15, 19].

The rest of the paper is organized as follows. In Section 2, we define some terms on linear decision trees, threshold circuits, and communication complexity. In Section 3, we give the lower bound for linear decision trees and a technical lemma. In Section 4, we prove the technical lemma. In Section 5, we give lower bounds for threshold circuits of depth $\omega(1)$ and bounded energy. In Section 6, we conclude with some remarks.

## 2 Definitions

### 2.1 Linear Decision Trees

Let $g$ be a *linear threshold function* with $n$ inputs, weights $w_1, w_2, \ldots, w_n$ and a threshold $t$. Then, for every input $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$, $g(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{i=1}^{n} w_i x_i - t\right)$ where $\mathrm{sign}(z) = 1$ if $z \geq 0$ and $\mathrm{sign}(z) = 0$ if $z < 0$. We assume throughout the paper that the weights and threshold of every threshold function are integers. A *linear decision tree* $T$ computing a Boolean function of $n$ variables is a binary decision tree in which each internal node is labeled by a linear threshold function of the $n$ variables and each leaf is labeled by 0 or 1. For a given input $\boldsymbol{x} \in \{0,1\}^n$, the output $T(\boldsymbol{x})$ of $T$ is determined by the following procedure starting from the root until reaching a leaf: if the linear threshold function at the current node outputs 0 for the input $\boldsymbol{x}$, then go to the left child; otherwise go the right. If the leaf reached is labeled by $z \in \{0,1\}$, then $T(\boldsymbol{x}) = z$. The *size* $l$ of $T$ is defined to be the number of leaves in $T$. We say that $T$ has weight $w$ if the linear threshold function $\mathrm{sign}(\sum_i w_i x_i - t)$ at each internal node of $T$ satisfies $\sum_i |w_i| \leq w$.

### 2.2 Threshold Circuits

A *threshold gate* with an arbitrary number $k$ of inputs computes a linear threshold function of $k$ inputs. A *threshold circuit* is a directed acyclic graph where each internal node is a threshold gate or an input variable. The *size* $s$ of a threshold circuit is defined to be the number of threshold gates in the circuit.

Let $C$ be a threshold circuit computing a Boolean function $f$ of $n$ variables $x_1, x_2, \ldots, x_n$, and have size $s$. Let $g_1, g_2, \ldots, g_s$ be the gates in $C$, where $g_1, g_2, \ldots, g_s$ are topologically ordered with respect to the underlying directed acyclic graph of $C$. We regard the output of $g_s$ as the *output* $C(\boldsymbol{x})$ *of* $C$, and call the gate $g_s$ the *top gate* of $C$. A threshold circuit $C$ *computes* a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if $C(\boldsymbol{x}) = f(\boldsymbol{x})$ for every input $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$. We say that $C$ has *weight* $w$ if the sum of the absolute values of the weights for the $n$ input variables $x_1, x_2, \ldots, x_n$ of each gate in $C$ is at most $w$. Note that only the weights of the input corresponding to $x_1, x_2, \ldots, x_n$ are taken into account. The *level* of a gate in $C$ is the length of the longest directed path to the gate from an input node. The *depth* $d$ of $C$ is the level of the top gate $g_s$ of $C$. We define the *energy* $e$ of a threshold circuit $C$ as the maximum number of gates outputting "1", where the maximum is taken over all inputs $\boldsymbol{x} \in \{0,1\}^n$ [18]. Thus, $e = \max_{\boldsymbol{x}\in\{0,1\}^n} \sum_{i=1}^{s} g_i(\boldsymbol{x})$, where $g_i(\boldsymbol{x})$ is the output of $g_i$ for $\boldsymbol{x} \in \{0,1\}^n$. Clearly $0 \le e \le s$. It should be noted that the inequality $d \le e$ does not necessarily holds for threshold circuits of depth $d$ and energy $e$. For example, the Parity function of $n$ variables can be computed by a threshold circuit of size $O(n)$, depth $O(n)$, and energy two [17].

### 2.3 Communication Complexity

Consider a game of two players, say Alice and Bob, with a Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, where Alice and Bob have unlimited computational power. Alice receives an input $\boldsymbol{x} \in \{0,1\}^n$ and Bob does an input $\boldsymbol{y} \in \{0,1\}^n$. Alice and Bob exchange bits according to a protocol, and try to compute the value $f(\boldsymbol{x},\boldsymbol{y})$. The *cost* of a protocol is defined to be the maximum number of exchanged bits in the protocol. There are several variants of communication complexity measures of Boolean functions. In this paper, we consider the following three of them.

**Definition 1.** *The* deterministic communication complexity *of $f(\boldsymbol{x},\boldsymbol{y})$, denoted by $D(f)$, is defined to the minimum cost over all the deterministic protocols that compute $f(\boldsymbol{x},\boldsymbol{y})$ for every input $\boldsymbol{x} \times \boldsymbol{y} \in \{0,1\}^n \times \{0,1\}^n$.*

**Definition 2.** *Alice and Bob can use an unlimited "private" source of random bits. The* unbounded-error communication complexity *of $f(\boldsymbol{x},\boldsymbol{y})$, denoted by $U(f)$, is defined to be the minimum cost over all the randomized protocols that compute $f(\boldsymbol{x},\boldsymbol{y})$ correctly with probability strictly greater than $1/2$ for every input $\boldsymbol{x} \times \boldsymbol{y} \in \{0,1\}^n \times \{0,1\}^n$.*

**Definition 3.** *Alice and Bob share an unlimited "public" source of random bits. For each real number $\epsilon$, $0 \le \epsilon < 1/2$, the* bounded-error communication complexity *of $f(\boldsymbol{x},\boldsymbol{y})$, denoted by $R_\epsilon(f)$, is defined to be the minimum cost over all the randomized protocols that compute $f(\boldsymbol{x},\boldsymbol{y})$ correctly with probability $1-\epsilon$ for every input $\boldsymbol{x} \times \boldsymbol{y} \in \{0,1\}^n \times \{0,1\}^n$.*

The Inner-Product function $IP_n$ of $2n$ variables is defined as follows. For every pair of inputs $\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \{0,1\}^n$ and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n) \in \{0,1\}^n$, $IP_n(\boldsymbol{x},\boldsymbol{y}) = x_1y_1 \oplus x_2y_2 \oplus \cdots \oplus x_ny_n$, where $\oplus$ denotes the XOR function. It is known that $IP_n$ has large unbounded-error and bounded-error communication complexity:

**Proposition 1 ([6, 13]).** *$U(IP_n) = \Omega(n)$, and $R_{\frac{1}{2}-\delta}(IP_n) = \Omega(n + \log \delta)$ for every number $\delta$, $0 < \delta \le 1/2$.*

## 3 Lower Bounds for Linear Decision Trees

Our main result is the following lower bound on size of linear decision trees:

**Theorem 1.** *Let $f$ be a Boolean function of $n$ variables such that $U(f) = \Omega(n)$. If a linear decision tree $T$ of weight $w$ computes $f$, then the size of $T$ is $2^{\Omega(\sqrt{n})}/w$.*

By Proposition 1 and Theorem 1, we immediately obtain the following exponential lower bound on the size of linear decision trees.

**Corollary 1.** *If a linear decision tree $T$ of weight $w = 2^{o(n)}$ computes $IP_n$, then the size of $T$ is $2^{\Omega(\sqrt{n})}$.*

We below give a proof of Theorem 1. In the paper [18], Uchizawa *et al.* show that there is close relationship between linear decision trees and threshold circuits of small energy, as follows.

**Lemma 1 ([18]).** *Assume that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a linear decision tree of $l$ leaves and weight $w$. Then $f$ can also be computed by a threshold circuit $C$ of size $O(l)$, energy $O(\log l)$ and weight $O(w)$.*

In other words, if a function $f$ cannot be computed by any threshold circuit of size $O(l)$, energy $O(\log l)$ and weight $O(w)$, then $f$ cannot be computed by any linear decision tree of size $O(l)$ and weight $O(w)$. Thus, a lower bound for threshold circuits implies a lower bound for linear decision trees. The following theorem gives the desired lower bound.

**Theorem 2.** *Let $f$ be a Boolean function of $n$ variables such that $U(f) = \Omega(n)$. If $f$ can be computed by a threshold circuit $C$ of energy $e$ and weight $w$, then the size $s$ of $C$ is $s = 2^{\Omega(n/e)}/w$.*

Combining Lemma 1 and Theorem 2, we can easily prove Theorem 1 as follows.

*Proof of Theorem 1.* Let $T$ be a linear decision tree that computes $IP_n$ and has size $l$ and weight $w$. If $\log l \geq \sqrt{n}$, we are done. Consider the other case

$$\log l < \sqrt{n}. \tag{1}$$

Then Lemma 1 implies that $IP_n$ can be computed by a threshold circuit $C$ of size $s = O(l)$, energy $e = O(\log l)$ and weight $O(w)$. By Theorem 1, we have $l \geq 2^{\Omega(n/\log l)}/w$, and hence Eq. (1) implies that $l = 2^{\Omega(\sqrt{n})}/w$. $\square$

Thus, it suffices to prove Theorem 2. We prove Theorem 2 by an unbounded-error communication complexity argument used in [7]. The following lemma summarize the argument.

**Lemma 2.** *Assume that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be represented by a threshold function of a number $k$ of Boolean functions $f_1, f_2, \ldots, f_k$ with weights $w_1, w_2, \ldots, w_k$, that is,*

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{k} w_i f_i(\boldsymbol{x})\right) \tag{2}$$

*for every input $\boldsymbol{x} \in \{0,1\}^n$. Then*

$$U(f) \leq \max_i D(f_i) + O(\log k). \tag{3}$$

The proof of Lemma 2 is simple and omitted due to the page limitation, but described in the appendix for completeness. Lemma 2 implies that we can obtain an upper bound on the unbounded-error communication complexity of a Boolean function $f$ by expanding $f$ to a linear combination of Boolean functions $f_1, f_2, \ldots, f_k$. The following lemma plays a key role in our proof, and give such Boolean functions.

**Lemma 3.** *Assume that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a threshold circuit $C$ of size $s$, depth $d$, energy $e$ and weight $w$. Then $f$ can be represented by a threshold function of a number $k$ of depth-2 threshold circuits $C_1, C_2, \ldots, C_k$ with weights $w_1, w_2, \ldots, w_k$, where*

$$k = \sum_{i=0}^{e} \binom{s}{i} \leq s^e. \tag{4}$$

*That is,*

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{k} w_i C_i(\boldsymbol{x})\right) \tag{5}$$

*for every input $\boldsymbol{x} \in \{0,1\}^n$. Besides, for every index $i$, $1 \leq i \leq k$, $C_i$ has size*

$$s_i \leq e + 1, \tag{6}$$

*and weight $w$. Moreover, the weights $w_1, w_2, \ldots, w_k$ satisfy*

$$\sum_{i=1}^{k} |w_i| \leq s^{2(d+1)^{e+1}}. \tag{7}$$

**Remark.** The above lemma has a quiet similar form to Lemma 2 in [19]. The main difference between them is that the right hand side of Eq. (7) is $s^{2(d+1)^{e+1}}$ while that of the corresponding equation of Lemma 2 in [19] is $2s^{3(e+1)^d}$. Note that the depth $d$ and energy $e$ symmetrically appear in the exponents in the formulas. This difference is critical to yield an exponential lower bound on the size of threshold circuits of depth $\omega(1)$ in Section 5. In fact, we require a new idea to obtain Eq. (7).

We prove Lemma 3 in the next section. Lemma 2 and 3 immediately imply Theorem 2, as follows.

*Proof of Lemma 2* Let $f$ be a Boolean function of $2n$ variables such that

$$U(f) = \Omega(n). \tag{8}$$

Assume that $f$ can be computed by a threshold circuit $C$ of size $s$, depth $d$, energy $e$ and weight $w$. Lemma 3 implies that $f$ can be represented by a threshold function of a number $k$ of threshold circuits $C_1, C_2, \ldots, C_k$ with weights

$w_1, w_2, \ldots, w_k$ satisfying Eqs. (4)–(6). For each integer $i$, $1 \le i \le k$, let $f_i$ be the function that $C_i$ computes, then Eq. (3) implies that

$$U(f) \le \max_i D(f_i) + O(\log k). \tag{9}$$

Let $s_i$ be the size of $C_i$. Since $C_i$ has weight $w$, Alice and Bob can compute the output of every gate in $C_i$ by such a protocol that one of the players sends in binary representation the sum of the products between the weights and the inputs of the gate. Hence we have $D(f_i) = O(s_i \log w)$ for every integer $i$, $1 \le i \le k$. Thus, Eqs. (4), (6) and (9) imply that

$$U(f) = O(s_i \log w + \log k) = O(e(\log w + \log s)) \tag{10}$$

By Eqs. (8) and (10), we obtain the desired result. □

# 4 Proof of Lemma 3

In this section, we prove Lemma 3. Assume that a function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a threshold circuit $C$ of size $s$, depth $d$, energy $e$, and weight $w$. Let $C$ consist of the gates $g_1, g_2, \ldots, g_s$. One may assume that $g_1, g_2, \ldots, g_s$ are topologically ordered with respect to the underlying directed acyclic graph of $C$, and that $g_s$ is the output gate of $C$. For each index $i$, $1 \le i \le s$, we denote by $w_{i,1}, w_{i,2}, \ldots, w_{i,n}$ the weights of the gate $g_i$ for the inputs $x_1, x_2, \ldots, x_n$ and by $w_{i,g_1}, w_{i,g_2}, \ldots, w_{i,g_s}$ the weights of $g_i$ for the outputs of the gates $g_1, g_2, \ldots, g_s$, respectively. Since the gates $g_1, g_2, \ldots, g_s$ are topologically ordered, we have $w_{i,g_i} = w_{i,g_{i+1}} = \cdots = w_{i,g_s} = 0$. We denote by $t_i$ the threshold of $g_i$. Thus, the output $g_i(\boldsymbol{x})$ of $g_i$ for $\boldsymbol{x} \in \{0,1\}^n$ is represented as

$$g_i(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{j=1}^{n} w_{i,j} x_j + \sum_{j=1}^{i-1} w_{i,g_j} g_j(\boldsymbol{x}) - t_i\right). \tag{11}$$

For each gate $g_i$, $1 \le i \le s$, we denote by $\mathrm{lev}(g_i)$ the level of the gate $g_i$ in $C$. We shall present threshold circuits and weights satisfying Eqs. (4)–(7).

Define $\mathbb{S}$ as a family of subsets of $\{1, 2, \ldots, s\}$ such that

$$\mathbb{S} = \{S \subseteq \{1, 2, \ldots, s\} \mid 0 \le |S| \le e\}.$$

For every set $S \in \mathbb{S}$, we construct a depth-2 threshold circuit $C_S$ consisting of $|S| + 1$ gates as follows. In the first level, the circuit $C_S$ contains a gate $g_i^S$ for every index $i \in S$ that computes

$$g_i^S(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{j=1}^{n} w_{i,j} x_j + \sum_{j \in S} w_{i,g_j} - t_i\right) \tag{12}$$

for every input $\boldsymbol{x} \in \{0,1\}^n$. In the second level, $C_S$ contains a gate $g_S$ computing AND of the outputs of the gates $g_i^S$ for every $i \in S$, that is,

$$C_S(\boldsymbol{x}) = g_S(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{i\in S} g_i^S(\boldsymbol{x}) - |S|\right). \tag{13}$$

If $S = \emptyset$, then $C_S(\boldsymbol{x}) = \mathrm{sign}(0) = 1$ for every input $\boldsymbol{x} \in \{0,1\}^n$.

For each set $S \in \mathbb{S}$, we decide the weight $w_S$ for the circuit $C_S$ as follows. For $S = \emptyset$, we decide $w_S = -1$. For each set $S \in \mathbb{S}\backslash\{\emptyset\}$, assume that $S$ contain the indices $i_1, i_2, \ldots, i_{|S|}$ such that $1 \le i_1 < i_2 < \cdots < i_{|S|} \le s$. Then we denote by $\boldsymbol{v} = (v_1, v_2, \ldots, v_e)$ the *weight vector* for $S$ defined as follows: For each index $j$, $1 \le j \le e$,

$$v_j = \begin{cases} d + 1 - \mathrm{lev}(g_{i_j}) & \text{if } 1 \le j \le |S|; \\ 0 & \text{if } |S| + 1 \le j \le e. \end{cases}$$

Clearly, we have $0 \le v_j \le d$, for every index $j$, $1 \le j \le e$. Then we define $[S]$ as the integer whose $(d+1)$-nary representation is the weight vector $\boldsymbol{v}$, that is,

$$[S] = \sum_{j=1}^{e} v_j (d+1)^{e-j}.$$

We then decide the weight for $C_S$ as

$$w_S = \begin{cases} k^{[S]} & \text{if } g_s \in S; \\ -k^{[S]} & \text{otherwise.} \end{cases} \tag{14}$$

where

$$k = |\mathbb{S}| = \sum_{i=0}^{e} \binom{s}{i}.$$

Consequently, we obtain the following threshold function:

$$\mathrm{sign}\left(\sum_{S\in\mathbb{S}} w_S C_S(\boldsymbol{x})\right). \tag{15}$$

In the rest of the section, we prove that the threshold function (15) satisfies Eqs. (4)$-$(7).

Since

$$k = |\mathbb{S}| = \sum_{i=0}^{e} \binom{s}{i} \le s^e,$$

Eq. (4) holds. Clearly, for every set $S \in \mathbb{S}$, $\mathrm{size}(C_S) = |S| + 1 \le e + 1$, and hence the threshold function (15) satisfies Eq. (6). Moreover, it holds that

$$[S] \le \sum_{j=1}^{e} d \cdot (d+1)^{e-i} \le (d+1)^e$$

for every $S \in \mathbb{S}$. Hence

$$\begin{aligned}\sum_{S\in\mathbb{S}} |w_S| &= \sum_{S\in\mathbb{S}} k^{[S]} \\ &\leq k \cdot k^{(d+1)^e} \\ &\leq s^{e(1+(d+1)^e)} \\ &\leq s^{2(d+1)^{e+1}}.\end{aligned}$$

We have thus proved Eq. (7). Below, we prove Eq. (5), that is,

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{S\in\mathbb{S}} w_S C_S(\boldsymbol{x})\right). \tag{16}$$

for every $\boldsymbol{x} \in \{0,1\}^n$.

Consider an arbitrary and fixed input $\boldsymbol{x} \in \{0,1\}^n$. Let

$$S^* = \{i \in \{1,2,\dots,s\} \mid \text{the gate } g_i \text{ outputs 1 for } \boldsymbol{x} \text{ in } C\}.$$

Thus, for every index $i \in S^*$,

$$g_i(\boldsymbol{x}) = 1. \tag{17}$$

Let

$$\mathbb{F} = \{S \in \mathbb{S} \mid \forall i \in S, \text{ the gate } g_i^S \text{ outputs 1 for } \boldsymbol{x} \text{ in } C_S\},$$

then Eq. (13) implies that, for every set $S \in \mathbb{F}$,

$$C_S(\boldsymbol{x}) = g_S[\boldsymbol{x}] = 1. \tag{18}$$

Eqs. (11) and (12) imply that, for every index $i \in S^*$,

$$\begin{aligned}g_i(\boldsymbol{x}) &= \operatorname{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{j=1}^{i-1} w_{i,g_j} g_j[\boldsymbol{x}] - t_i\right) \\ &= \operatorname{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{j\in S^*} w_{i,g_j} - t_i\right) = g_i^{S^*}(\boldsymbol{x})\end{aligned}$$

and hence Eq. (17) implies that $g_i^{S^*}(\boldsymbol{x}) = 1$. We thus have

$$S^* \in \mathbb{F}. \tag{19}$$

Eqs. (18) and (19) imply that

$$g_{S^*}[\boldsymbol{x}] = 1. \tag{20}$$

Then the following claim holds.

*Claim.* For every set $S \in \mathbb{F} \backslash \{S^*\}$, $[S] \leq [S^*] - 1$.

The proof of Claim 1 is omitted due to the page limitation, but described in the appendix. We are now ready to prove Eq. (16). There are two cases to consider: (i) $f(\boldsymbol{x}) = C(\boldsymbol{x}) = 1$, and (ii) $f(\boldsymbol{x}) = C(\boldsymbol{x}) = 0$. We prove Eq. (16) only for the case (i), since the proof for the other case is similar. Consider an arbitrary input $\boldsymbol{x}$ such that $f(\boldsymbol{x}) = C(\boldsymbol{x}) = 1$. In this case, it suffices to prove that

$$\sum_{S \in \mathbb{S}} w_S C_S(\boldsymbol{x}) \geq 0. \tag{21}$$

Eqs. (18) and (20) implies that

$$\begin{aligned} \sum_{S \in \mathbb{S}} w_S C_S(\boldsymbol{x}) &= w_{S^*} + \sum_{S \in \mathbb{S} \backslash \{S^*\}} w_S C_S(\boldsymbol{x}) \\ &\geq w_{S^*} - \sum_{S \in \mathbb{F} \backslash \{S^*\}} |w_S|. \end{aligned} \tag{22}$$

Since $C(\boldsymbol{x}) = g_s(\boldsymbol{x}) = 1$, we have $s \in S^*$, and hence Eqs. (14) and (22) imply that

$$\sum_{S \in \mathbb{S}} w_S C_S(\boldsymbol{x}) \geq k^{[S^*]} - \sum_{S \in \mathbb{F} \backslash \{S^*\}} k^{[S]}. \tag{23}$$

If $\mathbb{F} = \{S^*\}$, then we have

$$\sum_{S \in \mathbb{S}} w_S C_S(\boldsymbol{x}) \geq k^{[S^*]} > 0,$$

and hence Eq. (21) holds. If $\mathbb{F} \neq \{S^*\}$, then Claim 1 implies that $[S] \leq [S^*] - 1$ for every $S \in \mathbb{F} \backslash S^*$. Therefore, by Eq. (23) we have

$$\begin{aligned} \sum_{S \in \mathbb{S}} w_S C_S(\boldsymbol{x}) &\geq k^{[S^*]} - (k-1) k^{[S^*]-1} \\ &= k^{[S^*]-1} > 0, \end{aligned}$$

and hence Eq. (21) holds.

## 5 Lower Bounds for Threshold Circuits of Depth $\omega(1)$

In this section, we give two exponential lower bounds on the size of threshold circuits of depth $\omega(1)$. The first lower bound is immediately obtained from Theorem 2, as follows.

**Corollary 2.** *If $IP_n$ can be computed by a threshold circuit $C$ of size $s$, energy $e = n^{o(1)}$ and weight $w = 2^{o(n)}$, then $s = 2^{\Omega(n^{1-o(1)})}$.*

Note that the bound is, in fact, independent of depth.

The other lower bound is obtained from Lemma 2 by which arbitrary threshold circuit $C$ computing a Boolean function $f$ can be converted to a threshold function of a number $k$ of depth 2 threshold circuits $C_1, C_2, \ldots, C_k$ of size at most $e+1$. This lemma enable us to use a standard communication complexity argument that yields a lower bound on the size of threshold circuits of depth three. More precisely, we obtain the following theorem.

**Theorem 3.** *Assume that $f$ is a Boolean function of $2n$ variables such that*

$$R_{\frac{1}{2}-\delta}(f) = \Omega(n + \log \delta) \tag{24}$$

*for every number $\delta$, $0 < \delta \leq 1/2$. If a threshold circuit $C$ of depth $d$ and energy $e$ computes $f$, then the size $s$ of $C$ satisfies*

$$s = exp\left(\Omega\left(\frac{n}{e(d+1)^{(e+1)}}\right)\right) \tag{25}$$

We prove Theorem 3 in the appendix. Since $IP_n$ satisfies Eq. (24), Theorem 3 immediately yields the following corollary.

**Corollary 3.** *If a threshold circuit $C$ of depth $d = n^{o(1)}$ and energy $e = O(1)$ computes $IP_n$, then the size $s$ of $C$ satisfies $s = 2^{\Omega(n^{1-o(1)})}$.*

## 6 Conclusion

In this paper, we consider a binary linear decision tree $T$ computing a Boolean function $f$. We prove that if $T$ has weight $w$ and $f$ has large unbounded-error communication complexity, then $T$ must have size $2^{\Omega(\sqrt{n})}/w$, which implies an exponential lower bound on the size of linear decision trees computing $IP_n$ provided that $w = 2^{o(\sqrt{n})}$. The energy complexity of threshold circuits plays important role in our proof; our result suggests that we can obtain a strong lower bound for some computational model if the model can be simulated by threshold circuits of small energy complexity.

In addition, we consider threshold circuits of depth $\omega(1)$ and bounded energy. We obtain an exponential lower bound on the size of threshold circuits of energy $n^{o(1)}$ and weight $2^{o(n)}$ and an exponential lower bound on the size of threshold circuits of depth $n^{o(1)}$ and energy $O(1)$.

## References


1. K. Amano and A. Maruoka. On the complexity of depth-2 circuits with threshold gates. In *Proceedings of the 30th Int. Symp. on Mathematical Foundations of Computer Science,* Springer Lect. Notes in Computer Science 3618, pp. 107–118, 2005.

2. A. Björner, L. Lovász, and A. C. C. Yao. Linear decision trees: volume estimates and topological bounds. In *Procedings of the 24th ACM Symposyum on Theory of Computing*, pp. 170–177, 1992.
3. D. P. Dobkin and R. J. Lipton. On the complexity of computations under varying sets of primitives. *Journal of Computer and System Sciences*, 3:1–8, March 1982.
4. J. Erickson. Lower bounds for linear satisfiability problems. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 388–395. 1995.
5. R. Fleischer. Decision trees: Old and new results. *Information and Computation*, 152:44–61, March 1999.
6. J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65:612–625, 2002.
7. J. Forster, M. Krause, S. V. Lokam, R. Mubarakzjanov, N. Schmitt, and H. U. Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science* Springer Lect. Notes in Computer Science 5734, pp. 171–182, 2001.
8. H. D. Gröger and G. Turán. On linear decision trees computing boolean functions. In *Proceedings of the 18th Int. Conf. on Automata, Languages and Programming,* Springer Lect. Notes in Computer Science 510, pp. 707–718, 1991.
9. D. Grigoriev, M. Karpinski, and A.C. Yao. An exponential lower bound on the size of algebraic decision trees for max. *Computational Complexity*, 7(3):193–203, December 1998.
10. A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and Sysntem Sciences*, 46:129–154, 1993.
11. K. A. Hansen and V. V. Podolskii. Exact threshold circuits. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pp. 270–279, 2010.
12. J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1993.
13. E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.
14. N. Nisan. The communication complexity of threshold gates. *Proceeding of "Combinatorics, Paul Erdös is Eighty"*, pp. 301–315, 1993.
15. A. Razborov and A. Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth 3 threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45:303–307, 1993.
16. J. Michael Steele and Andrew C. Yao. Lower bounds for algebraic decision trees. *Journal of Algorithms*, 18:86–91, 1979.
17. A. Suzuki, K. Uchizawa, and X. Zhou. Energy-efficient threshold circuits computing mod functions. In *Proceedings of the 17th Computing: the Australasian Theory Symposium,* Perth, Australia. CRPIT, 119. Alex Potanin and Taso Viglas Eds., ACS., pp. 105–110, 2011.
18. K. Uchizawa, R. Douglas, and W. Maass. On the computational power of threshold circuits with sparse activity. *Neural Computation*, 18(12):2994–3008, 2006.
19. K. Uchizawa and E. Takimoto. Exponential lower bounds on the size of constant-depth threshold circuits with small energy complexity. *Theoretical Computer Science*, 407(1-3):474–487, 2008.
20. I. Wegener. Optimal decision trees and one-time-only branching programs for symmetric boolean functions. *Information and Control*, 62(2-3):129–143, August 1984.

# Appendix

In this appendix, we describe proofs of Lemma 2, Claim 1, and Theorem 3.

**Proof of Lemma 2**

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Assume that $f$ can be represented by a threshold function of a number $k$ of Boolean functions $f_1, f_2, \ldots, f_k$ with weights $w_1, w_2, \ldots, w_k$ (See Eq. (49)). Without loss of generality, we assume that none of the weights $w_1, w_2, \ldots, w_k$ is zero. Let

$$W = \sum_{i=1}^{k} |w_i|.$$

Below, we shall show that there is a randomized protocol that compute a value of $f$ by exchanging

$$\max_i D(f_i) + O(\log k) \tag{26}$$

bits with error probability

$$\epsilon < \frac{1}{2} \tag{27}$$

Let $f_{k+1}$ be a trivial Boolean function $\{0,1\}^n \times \{0,1\}^n \to \{1\}$. In the protocol, Alice and Bob consider the following new threshold function:

$$\text{sign}\left(\sum_{i=1}^{k+1} w'_i f_i(\boldsymbol{x}, \boldsymbol{y})\right),$$

where $w'_i = 2w_i$ for each index $i$, $1 \le i \le k$, and $w'_{k+1} = 1$. Clearly,

$$\begin{aligned}\sum_{i=1}^{k+1} w'_i f_i(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=1}^{k} 2w_i f_i(\boldsymbol{x}, \boldsymbol{y}) + w'_{k+1} f_{k+1}(\boldsymbol{x}, \boldsymbol{y}) \\ &= 2\left(\sum_{i=1}^{k} w_i f_i(\boldsymbol{x}, \boldsymbol{y}) + \frac{1}{2}\right).\end{aligned} \tag{28}$$

The weights $w_1, w_2, \ldots, w_k$ are integers, and hence the value

$$\sum_{i=1}^{k} w_i f_i(\boldsymbol{x}, \boldsymbol{y})$$

is integer. Thus, Eq. (28) implies that

$$f(\boldsymbol{x}, \boldsymbol{y}) = \text{sign}\left(\sum_{i=1}^{k+1} w'_i f_i(\boldsymbol{x}, \boldsymbol{y})\right) = \text{sign}\left(\sum_{i=1}^{k} w_i f_i(\boldsymbol{x}, \boldsymbol{y})\right). \tag{29}$$

and

$$\sum_{i=1}^{k+1} w'_i f_i(\boldsymbol{x}, \boldsymbol{y}) \neq 0. \tag{30}$$

Let

$$W' = \sum_{i=1}^{k+1} |w'_i|,$$

then

$$W' \leq 2W + 1 \leq 3W. \tag{31}$$

Our protocol consists of the following three steps, and gives a Boolean value $b \in \{0, 1\}$ as $f(\boldsymbol{x}, \boldsymbol{y})$ for every $\boldsymbol{x} \times \boldsymbol{y} \in \{0, 1\}^{2n}$.

**[Step 1]** Using a random bit string, Alice chooses one of the indices $1, 2, \dots, k+1$, so that for each index $i$, $1 \leq i \leq k+1$,

$$\Pr[\text{Alice choose } i] = \frac{|w'_i|}{W'} \tag{32}$$

where the probability is taken over all random strings. Suppose that Alice chooses an index $j$, $1 \leq j \leq k+1$. Then, Alice send the integer $j$ in binary representation.

**[Step 2]** Using the best deterministic protocol for the function $f_j$, Alice and Bob computes $f_j(\boldsymbol{x}, \boldsymbol{y})$.

**[Step 3]** If $f_j(\boldsymbol{x}, \boldsymbol{y}) = 1$, then the output $b \in \{0, 1\}$ of our protocol is decided as follows:

$$b = \begin{cases} f_j(\boldsymbol{x}, \boldsymbol{y}) & \text{if } w'_j > 0; \\ \overline{f_j(\boldsymbol{x}, \boldsymbol{y})} & \text{if } w'_j < 0, \end{cases} \tag{33}$$

where $\overline{f_j(\boldsymbol{x}, \boldsymbol{y})}$ is the negation of $f_j(\boldsymbol{x}, \boldsymbol{y})$. If $f_j(\boldsymbol{x}, \boldsymbol{y}) = 0$, then Alice sends a Boolean value 0 or 1 with probability 1/2 as the output $b$.

In Step 1, Alice sends $O(\log k)$ bits. In Step 2, Alice and Bob send at most $D(f_j)$ bits. In Step 3, Alice sends at most one bit. We have thus verified Eq. (26).

We then prove Eq. (27). By Eq. (31), it suffices to prove that

$$\epsilon \leq \frac{1}{2} - \frac{1}{2W'}. \tag{34}$$

Consider an arbitrary fixed pair of inputs $\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^n$. We below prove Eq. (34) only for the case where $f(\boldsymbol{x}, \boldsymbol{y}) = 1$, since the proof for the other case is similar. Let $I^+$, $I^-$ and $I^*$ be sets of the indices $1, 2, \dots, k+1$ defined as follows:

$$I^+ = \{i \in \{1, 2, \dots, k+1\} \mid f_i(\boldsymbol{x}, \boldsymbol{y}) = 1 \text{ and } w_i > 0\};$$

$$I^- = \{i \in \{1, 2, \ldots, k+1\} \mid f_i(\boldsymbol{x}, \boldsymbol{y}) = 1 \text{ and } w_i < 0\};$$

and

$$I^* = \{i \in \{1, 2, \ldots, k+1\} \mid f_i(\boldsymbol{x}, \boldsymbol{y}) = 0\};$$

By the protocol, we clearly have

$$\epsilon = \Pr[j \in I^-] + \frac{1}{2} \cdot \Pr[j \in I^*]. \tag{35}$$

Let $W^+, W^-$ and $W^*$ respectively denote

$$W^+ = \sum_{i \in I^+} |w_i|; \quad W^- = \sum_{i \in I^-} |w_i|; \quad W^* = \sum_{i \in I^*} |w_i|. \tag{36}$$

Clearly, we have

$$W' = W^+ + W^- + W^*. \tag{37}$$

By Eqs. (35)−(37), we have

$$\begin{aligned}
\epsilon &= \frac{W^-}{W'} + \frac{W^*}{2W'} \\
&= \frac{W^-}{W'} + \frac{W' - W^+ - W^-}{2W'} \\
&= \frac{1}{2} - \frac{W^+ - W^-}{2W'}
\end{aligned} \tag{38}$$

Since $f(\boldsymbol{x}, \boldsymbol{y}) = 1$, we have by Eq. (30)

$$W^+ - W^- \geq 1. \tag{39}$$

By Eqs. (38) and (39), we thus verified Eq. (34). □

**Proof of Claim 1** Let $S$ be an arbitrary set in $\mathbb{F}$ such that

$$S \neq S^*. \tag{40}$$

Thus, for every index $i \in S$,

$$g_i^S(\boldsymbol{x}) = 1. \tag{41}$$

For each index $j$, $1 \leq j \leq d$, we define $S_j$ as a set of indices $i$ such that the gate $g_i$ is in the level $j$ of $C$, that is,

$$S_j = \{i \in S \mid \text{lev}(g_i) = j\}$$

Similarly, we define

$$S_j^* = \{i \in S^* \mid \text{lev}(g_i) = j\}.$$

By Eq. (40), there exists an index $h$ such that, for every index $j$, $1 \le j \le h-1$,

$$S_j = S_j^* \tag{42}$$

and

$$S_h \ne S_h^*.$$

Let $i$ be an arbitrary index in $S_h$. Then,

$$\begin{aligned} g_i^S(\boldsymbol{x}) &= \operatorname{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{j\in S} w_{i,g_j} - t_i\right) \\ &= \operatorname{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{j\in S_1\cup S_2\cup\cdots\cup S_{h-1}} w_{i,g_j} - t_i\right). \end{aligned} \tag{43}$$

By Eq. (42) and (43), we have

$$\begin{aligned} g_i^S(\boldsymbol{x}) &= \operatorname{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{j\in S^*} w_{i,g_j} - t_i\right) \\ &= \operatorname{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{j=1}^{i-1} w_{i,g_j} g_j(\boldsymbol{x}) - t_i\right) = g_i(\boldsymbol{x}) \end{aligned}$$

Hence, by Eqs. (41) and (44) we have $g_i(\boldsymbol{x}) = 1$, and thus $i \in S_h^*$. Consequently, we have

$$S_h \subset S_h^*. \tag{44}$$

Eq. (42) implies that for every index $j$, $1 \le j \le h-1$,

$$|S_j| = |S_j^*|, \tag{45}$$

and Eq. (44) implies that

$$|S_h| \le |S_h^*| - 1. \tag{46}$$

Let $\boldsymbol{v} = (v_1, v_2, \ldots, v_e)$ be the weight vector for $S$, let $\boldsymbol{v}^* = (v_1^*, v_2^*, \ldots, v_e^*)$ be the weight vector for $S^*$, and let $\alpha = |S_1| + |S_2| + \cdots + |S_h|$. Then Eqs. (45) and (46) imply that, for every index $j$, $1 \le j \le \alpha$,

$$v_j = v_j^*,$$

and

$$v_{\alpha+1} < v_{\alpha+1}^*.$$

Thus, $\boldsymbol{v^*}$ is lexicographically larger than $\boldsymbol{v}$, and hence the claim follows.

**Proof of Theorem 3** Assume that $f$ is a Boolean function of $2n$ variables and there is a number $\gamma > 0$ such that

$$R_{\frac{1}{2}-\delta}(f) = \Omega(n^\gamma + \log \delta) \tag{47}$$

for every number $\delta$, $0 < \delta \leq 1/2$, and that $f$ can be computed by a threshold circuit $C$ of size $s$, depth $d$, and energy $e$. Lemma 3 implies that $f$ can be represented by a threshold function with a number $k$ of threshold circuits $C_1, C_2, \ldots, C_k$ and weights $w_1, w_2, \ldots, w_k$ satisfying Eqs. (4)−(7). Let

$$W = \sum_{i=1}^{k} |w_i|,$$

and let $f_i$ be the Boolean function computed by the circuit $C_i$ for every index $i$, $1 \leq i \leq k$.

For Boolean functions that can be computed by threshold circuits, Nisan obtains the following upper bound on the bounded-error communication complexity [14].

**Lemma 4 ([14]).** *If a Boolean function $f$ of $2n$ variables can be computed by a threshold circuit of size $s$, then*

$$R_\epsilon(f) = O\left(s\left(\log n + \log \frac{s}{\epsilon}\right)\right)$$

*for every number $\epsilon$, $0 \leq \epsilon < 1/2$.*

Thus, by Lemma 4 and Eq. (6) we have

$$\max_i R_{\frac{1}{12W}}(f_i) = O((e+1)(\log n + \log 12W(e+1))) \tag{48}$$

By Eqs. (7) and (48), we have

$$\max_i R_{\frac{1}{12W}}(f_i) = O(e(\log n + (d+1)^{e+1} \log s)).$$

The following lemma is easy to verify:

**Lemma 5.** *Assume that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be represented by a threshold function with a number $k$ of Boolean functions $f_1, f_2, \ldots, f_k$ and weights $w_1, w_2, \ldots, w_k$, that is,*

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{k} w_i f_i(\boldsymbol{x})\right) \tag{49}$$

*for every input $\boldsymbol{x} \in \{0,1\}^n$. Then*

$$R_{\frac{1}{2}-\frac{1}{12W}}(f) \leq \max_i R_{\frac{1}{12W}}(f_i) \tag{50}$$

*where*

$$W = \sum_{i=1}^{k} |w_i|. \tag{51}$$

*Proof.* Similarly to the proof of Lemma 2, we can prove Lemma 5 by providing a protocol that upperbounds $R_{\frac{1}{2}-\frac{1}{12W}}(f)$. In this proof, we use same proof scheme and notations as the ones in the proof of Lemma 2. Note that we here consider the bounded-error communication complexity, and thus Alice and Bob share an public source of random bits.

The desired protocol is almost same as the one presented in the proof of Lemma 2, and consists of the following three steps.

**[Step 1]** Using a public random bit string, Alice and Bob choose one of the indices $1, 2, \ldots, k+1$, so that for each index $i$, $1 \le i \le k+1$,

$$\Pr[\text{Alice choose } i] = \frac{|w'_i|}{W'}$$

where the probability is taken over all random strings. Suppose that Alice and Bob choose an index $j$, $1 \le j \le k+1$.

**[Step 2]** Using the best randomized protocol with error probability $1/4W'$ for the function $f_j$, Alice and Bob computes $f_j(\boldsymbol{x}, \boldsymbol{y})$.

**[Step 3]** If $f_j(\boldsymbol{x}, \boldsymbol{y}) = 1$, then the output $b \in \{0, 1\}$ of our protocol is decided as follows:

$$b = \begin{cases} f_j(\boldsymbol{x}, \boldsymbol{y}) & \text{if } w'_j > 0; \\ \overline{f_j(\boldsymbol{x}, \boldsymbol{y})} & \text{if } w'_j < 0, \end{cases}$$

where $\overline{f_j(\boldsymbol{x}, \boldsymbol{y})}$ is the negation of $f_j(\boldsymbol{x}, \boldsymbol{y})$. If $f_j(\boldsymbol{x}, \boldsymbol{y}) = 0$, then Alice and Bob choose a Boolean value 0 or 1 with probability 1/2 as the output $b$ by a public random bit.

Clearly, Alice and Bob communicate $R_{\frac{1}{4W'}}(f_j)$ bits in the protocol, and the error probability $\epsilon$ of the protocol is

$$\epsilon \le \Pr[j \in I^-] + \frac{1}{2} \cdot \Pr[j \in I^*] + \frac{1}{4W'}.$$

Since we prove that

$$\Pr[j \in I^-] + \frac{1}{2} \cdot \Pr[j \in I^*] \le \frac{1}{2} - \frac{1}{2W'}$$

in the proof of Lemma 2 (See Eqs. (34) and (35)), we have

$$\begin{aligned} \epsilon &\le \frac{1}{2} - \frac{1}{2W'} + \frac{1}{4W'} \\ &\le \frac{1}{2} - \frac{1}{4W'}. \end{aligned} \tag{52}$$

By Eqs (31) and (52), we verified the theorem. □

Lemma 5 implies that

$$R_{\frac{1}{2}-\frac{1}{12W}}(f) = O(e(\log n + (d+1)^{e+1} \log s)). \tag{53}$$

By Eq. (47) with $\delta = 1/12W$ and Eq. (53), we obtain the desired result. □