Electronic Colloquium on Computational Complexity, Report No. 153 (2018)

New Bounds for Energy Complexity of Boolean Functions

Krishnamoorthy Dinesh* Samir Otiv† Jayalal Sarma*

Abstract

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ computed by a circuit C over a finite basis $\mathcal{B}$, the *energy complexity* of C (denoted by $\mathsf{EC}_{\mathcal{B}}(C)$) is the maximum over all inputs $\{0,1\}^n$ the numbers of gates of the circuit C (excluding the inputs) that output a one. Energy Complexity of a Boolean function over a finite basis $\mathcal{B}$ denoted by $\mathsf{EC}_{\mathcal{B}}(f) \stackrel{\text{def}}{=} \min_C \mathsf{EC}_{\mathcal{B}}(C)$ where C is a circuit over $\mathcal{B}$ computing f.

We study the case when $\mathcal{B} = \{\wedge_2, \vee_2, \neg\}$, the standard Boolean basis. It is known that any Boolean function can be computed by a circuit (with potentially large size) with an energy of at most $3n(1+\epsilon(n))$ for a small $\epsilon(n)$(which we observe is improvable to $3n-1$). We show several new results and connections between energy complexity and other well-studied parameters of Boolean functions.

- For all Boolean functions f, $\mathsf{EC}(f) \leq O(\mathsf{DT}(f)^3)$ where $\mathsf{DT}(f)$ is the optimal decision tree depth of f.
- We define a parameter *positive sensitivity* (denoted by psens), a quantity that is smaller than sensitivity [4] and defined in a similar way, and show that for any Boolean circuit C computing a Boolean function f, $\mathsf{EC}(C) \geq \mathsf{psens}(f)/3$.
- For a monotone function f, we show that $\mathsf{EC}(f) = \Omega(\mathsf{KW}^+(f))$ where $\mathsf{KW}^+(f)$ is the cost of monotone Karchmer-Wigderson game of f.
- Restricting the above notion of energy complexity to Boolean formulas, we show $\mathsf{EC}(F) = \Omega\left(\sqrt{\mathsf{L}(F)} - \mathsf{Depth}(F)\right)$ where $\mathsf{L}(F)$ is the size and $\mathsf{Depth}(F)$ is the depth of a formula F.

1 Introduction

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ computed by a circuit C over a basis $\mathcal{B}$, the *energy complexity* of C (denoted by $\mathsf{EC}_{\mathcal{B}}(C)$) is the maximum over all inputs $\{0,1\}^n$ the numbers of gates of the circuit C (excluding the inputs) that outputs a one. The energy complexity of a Boolean function over a basis $\mathcal{B}$ denoted by $\mathsf{EC}_{\mathcal{B}}(f) \stackrel{\text{def}}{=} \min_C \mathsf{EC}_{\mathcal{B}}(C)$ where C is a circuit over $\mathcal{B}$ computing f. A particularly interesting case of this measure of Boolean function, is when the individual gates allowed in the basis $\mathcal{B}$ are threshold gates (with arbitrary weights allowed). In this case, the term energy in the above model captures the number of neurons firing in the cortex of the human brain (see [18] and the references therein). This motivated the study of upper and lower bounds [18] on

*Indian Institute of Technology Madras, Chennai, India. Email : {kdinesh,jayalal}@cse.iitm.ac.in

†Maximl Labs, India. Part of the work was done while this author was a student at the Indian Institute of Technology Madras. Email : samir.otiv@maximl.com

ISSN 1433-8092

various parameters of energy efficient circuits - in particular the question of designing threshold circuits which are efficient in terms of energy as well as size computing various Boolean functions.

Indeed, irrespective of the recently discovered motivation mentioned above, the notion of energy complexity of Boolean functions, has been studied much before. Historically, the measure of energy complexity of Boolean functions[1] was first studied by Vaintsvaig [22] (under the name "power of a circuit"). Initial research was aimed at understanding the maximum energy needed to compute any n bit Boolean function for a *finite* basis $\mathcal{B}$ (denoted by $\mathsf{EC}_{\mathcal{B}}(n)$). Towards this end, Vaintsvaig [22] showed that for any finite basis $\mathcal{B}$, the value of $\mathsf{EC}_{\mathcal{B}}(n)$ is asymptotically between n and $\frac{2^n}{n}$. Refining this result further, Kasim-zade [13] gave a complete characterization by showing the following remarkable trichotomy : for any finite complete basis $\mathcal{B}$, either $\mathsf{EC}_{\mathcal{B}}(n) = \Theta(2^n/n)$ or $\Omega(2^{n/2}) \leq \mathsf{EC}_{\mathcal{B}}(n) \leq O(\sqrt{n}2^{n/2})$ or $\Omega(n) \leq \mathsf{EC}_{\mathcal{B}}(n) \leq O(n^2)$.

An intriguing question about the above trichotomy is where exactly does the standard Boolean basis $\mathcal{B} = \{\wedge_2, \vee_2, \neg\}$ fits in. By an explicit circuit construction, Kasim-zade [13] showed that $\mathsf{EC}_{\mathcal{B}}(n) \leq O(n^2)$. Recently, Lozhkin and Shupletsov [11] states (without proof) that the circuit construction by Kasim-zade [13] over the complete Boolean basis is of energy $4n$, thus deriving that $\mathsf{EC}_{\mathcal{B}}(n) \leq 4n$. Lozhkin and Shupletsov improves it to $3n(1+\epsilon(n))$ by constructing a circuit of size $\frac{2^n}{n}(1+\epsilon(n))$ for an $\epsilon(n)$ tending to 0 for large n. We observe that this bounds can be further improved to be at most $3n-1$ while the size is $2^{O(n)}$ by carefully following the construction in [11] (Proposition 2.1).

As mentioned in the beginning, in a more recent work, for the case when the basis is threshold gates[2], Uchizawa *et al*. [18] initiated the study of energy complexity for threshold circuits. More precisely, they defined the energy complexity of threshold circuits and gave some sufficient conditions for certain functions to be computed by small energy threshold circuits. In a sequence of works, Uchizawa *et al*. [19, 21] related energy complexity of Boolean functions under the threshold basis to the other well-studied parameters like circuit size and depth for interesting classes of Boolean functions. In a culminating result, Uchizawa and Takimoto [20] showed that constant depth thresholds circuits of unbounded weights with the energy restricted to $n^{o(1)}$ needs exponential size to compute the Boolean inner product function[3]. This is also important in the context of circuit lower bounds, where it is an important open question to prove exponential lower bounds against constant depth threshold circuits in general (without the energy constraints) for explicit functions.

Our Results : Returning to the context of standard Boolean basis $\mathcal{B} = \{\wedge_2, \vee_2, \neg\}$, we show several new results and connections between energy complexity and other Boolean function parameters. Since we are interested only in the standard Boolean basis $\mathcal{B}$, we use $\mathsf{EC}(f)$ to denote $\mathsf{EC}_{\mathcal{B}}(f)$.
Relation to parameters of Boolean functions : As our first and main contribution, we relate energy complexity, $\mathsf{EC}(f)$ of Boolean functions to other two parameters of Boolean functions that are not known to be related before, one in terms of a upper bound and the other in terms of a lower bound. In addition, we show lower bounds on the energy complexity of Boolean functions when

[1] A related notion has been studied in [9] where the energy is the number times the gates in a circuit switches its value. Recent studies [2, 3] looks at the energy of a circuit as a function of the voltage applied to the gates thereby allowing some of the gates to fail. We remark that the notion of energy of Boolean circuits studied in this paper is very different from those studied in the works mentioned.

[2] With values of the weights and threshold being arbitrary rational numbers, notice that this basis is no longer finite and hence the bounds and the related trichotomy are not applicable.

[3] $\mathsf{IP}(x,y) = \sum_i x_i y_i \mod 2$

restricted to formulas (instead of circuits), in terms of its formula size and depth.

For a function $f : \{0,1\}^n \to \{0,1\}$, let $\mathsf{DT}(f)$ denote the decision tree complexity of the Boolean function - the smallest depth of any decision tree computing the function f. We state our main result:

Theorem 1.1 (Main)**.** *For any Boolean function f, $\mathsf{EC}(f) \le O(\mathsf{DT}(f)^3)$.*

We remark that the size of the circuit constructed above is exponentially in $\mathsf{DT}(f)$. However, in terms of the energy of the circuit, this improves the bounds of [11] since it now depends only on $\mathsf{DT}(f)$. There are several Boolean functions, for which the decision trees are very shallow - a demonstrative example is the address function[4] where the decision tree is of depth $O(\log n)$. This gives a circuit computing the address function with $O(\log^3 n)$ energy.

On a related note, Uchizawa *et al.* [18], as a part of their main proof, showed a similar result for threshold decision trees which are decision trees where each internal node can query an arbitrary weighted threshold function on input variables. Let $\mathsf{DT}_{th}(f)$ denotes the depth of smallest depth threshold decision tree computing f. For a basis $\mathcal{T}$ consisting of arbitrary threshold functions, their results implies that $\mathsf{EC}_{\mathcal{T}}(f) \le 1 + \mathsf{DT}_{th}(f)$ (see Proposition 2.4 for details). Since their construction produces a weighted threshold circuit, it does not directly give us a low energy Boolean circuit even for Boolean decision trees.

To obtain lower bounds on energy, we define a new parameter called the positive sensitivity (which is at most the sensitivity of the Boolean function [4]). For a function $f : \{0,1\}^n \to \{0,1\}$ and an input $a \in \{0,1\}^n$, we define the *positive sensitivity* (denoted by $\mathsf{psens}(f)$) as the maximum over all inputs $a \in \{0,1\}^n$ - of the number of indices $i \in [n]$ such that $a_i = 1$ and $f(a \oplus e_i) \neq f(a)$. Here, $e_i \in \{0,1\}^n$ has the i^{th} bit alone set to 1. Using this parameter, we show the following.

Theorem 1.2. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ computed by a circuit C, $\mathsf{EC}(C) \ge \mathsf{psens}(f)/3$.*

The main tool in proving the above results is the notion of *continuous positive paths* which are paths in a circuit where all the gates in the path evaluate to 1. Using the same tool, we show that the monotone Karchmer-Wigderson games can be solved by exchanging at most $\mathsf{EC}(C) \log c$ where C is a circuit with fan-in at most c (see Lemma 4.2 for more details). This implies the following energy lower bound for computing monotone functions.

Theorem 1.3. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. Then $\mathsf{EC}(f) = \Omega(\mathsf{KW}^+(f))$.*

It is known that for the perfect matching function of a graph on n edges, denoted as f_{PM}, $\mathsf{KW}^+(f_{PM}) = \Omega(n)$ [14]. Hence, Theorem 1.3 implies that any Boolean circuit with bounded fan-in, computing f_{PM} will require energy at least $\Omega(\sqrt{n})$.

All the models considered so far are of fan-in 2. We now relax this requirement and consider the energy complexity of unbounded fan-in constant depth circuits computing specific functions. In this direction, we show the following.

Theorem 1.4. *Let C be any unbounded fan-in circuit of depth 3 computing the parity function on n variables. Then, $\mathsf{EC}(C)$ is $\Omega(n)$.*

---

[4] $\mathsf{ADDR}_k(x_1, x_2, \ldots, x_k, y_0, y_1, \ldots, y_{2^k-1}) = y_{int(x)}$ where $int(x)$ is the integer representation of the binary string $x_1 x_2 \ldots x_k$.

For a formula F, let $\mathsf{L}(F)$ be the number of leaves in F and $\mathsf{Depth}(F)$ be the length of the longest path from root to any leaf in F. For a Boolean function f, let $\mathsf{L}(f)$ be the minimum $\mathsf{L}(F)$ among all the formulas F computing f. Let $\mathsf{EC}^{\mathsf{F}}(f)$ be the minimum energy for any bounded fan-in formula computing f. Intuitively, Boolean formulas can take more energy than a circuit since we cannot "reuse" computation. Also for any formula F, $\mathsf{EC}^{\mathsf{F}}(F) \leq \mathsf{L}(F) - 1$. Hence, it would not be surprising if $\mathsf{EC}^{\mathsf{F}}(F)$ is also lower bounded by $\Omega(\mathsf{L}(F))$ giving a tight bound of $\mathsf{EC}^{\mathsf{F}}(f) = \Theta(\mathsf{L}(f))$. Towards this direction we show the following result.

Theorem 1.5. *For a Boolean function f, computed by a formula F,*

$$\mathsf{EC}^{\mathsf{F}}(F) = \Omega\left(\sqrt{\mathsf{L}(F)} - \mathsf{Depth}(F)\right)$$

Related work : We discuss some of the recent results on energy complexity of computing Boolean functions in various circuit models.

Observe that since any Boolean circuit is also a threshold circuit, $\mathsf{EC}(f) \geq \mathsf{EC}_{\mathcal{T}}(f)$. Hence, for a function f, known lower bound on $\mathsf{EC}_{\mathcal{T}}(f)$ translates to a lower bound on $\mathsf{EC}(f)$. In this context, Table 1 summarizes known results on bounds on energy complexity of threshold circuits in terms of the parameters size, depth and fan-in for certain classes of Boolean functions. For designing energy efficient circuits, techniques or tools to reduce the energy complexity of circuits is relevant in this context. Table 2 summarizes known results on energy complexity of Boolean functions on ways to transform circuits to energy efficient ones.

`Param`	`Function` f `is ...`	`Gate`	Trade-off	Ref
ℓ	Symmetric	any	$\ell \geq \frac{n-b_f}{e}$	[16]
s	Symmetric	Unate	$s^e \geq \frac{n+1-a_f}{b_f}$	[21]
d, s	any	Threshold	$R_{0.5-\delta}(f) = O(e^d \log s)$, $\delta = \frac{1}{s^{O(e^d)}}$	[20]

Table 1: Known bounds on energy e of circuits computing Boolean functions

Energy vs circuit parameters : Table 1 presents information : *"Energy vs Parameter tradeoff for any circuit C using specific type of gates computing the function f"*. The parameters involved are $s = \mathsf{Size}(f)$, $d = \mathsf{Depth}(f)$, ℓ is fanin of gates in C and e is the optimum energy of a circuit with gates of type `Gate` computing f. We denote by $R_\delta(f)$ the two sided error public coin randomized communication complexity of f with error probability δ.

We now describe some of the non-standard notation used in Table 1. Any symmetric function $f : \{0,1\}^n \to \{0,1\}$ can be described by via an $n+1$ length vector – v_f as $f(x) = v_f(|x|)$ for all $x \in \{0,1\}^n$ where $|x|$ is the number of ones in x. For $b \in \{0,1\}$, let g_b is maximum length of consecutive b's in v_f. Let $a_f \stackrel{\text{def}}{=} \min\{g_0, g_1\}$ and $b_f \stackrel{\text{def}}{=} \max\{g_0, g_1\}$.

Results on transforming circuits to energy efficient ones : Table 2 presents the information : *"Given a circuit C with $\mathsf{Energy}(C) = e$ of with gates of type A then, there exists a circuit C′ with gates of type B computing the same function as C with bounds on $\mathsf{Size}(C')$, $\mathsf{Depth}(C')$, $\mathsf{Energy}(C')$."*

A	B	$\mathsf{Size}(C')$	$\mathsf{Depth}(C')$	$\mathsf{Energy}(C')$	Ref
Any	Threshold	$\leq O((e+n)\mathsf{Size}(C))$	$O(e)$	-	[19]
Threshold/Unate	Threshold/Unate	$\leq 2 \cdot e \cdot \mathsf{Size}(C) + 1$	$\leq 2 \cdot e + 1$	e	[19]

Table 2: Transforming circuit of type A to type B

Organization of the paper. The rest of the paper is organized as follows. We start with preliminaries in Section 2. We show new bounds on energy complexity in terms of decision tree depth in Section 3. Then, we show two methods to obtain lower bounds on energy complexity in Section 4.1 and Section 4.2 using the notion of *continuous positive paths* (introduced in Section 4.1.1). In Section 4.3, we show energy lower bounds for depth 3 circuits computing a specific function. Following this, in Section 5, we show energy lower bounds for Boolean formulas. We conclude in Section 6 outlining some directions for further exploration.

2 Preliminaries

A Boolean circuit C over the basis $\mathcal{B} = \{\wedge_2, \vee_2, \neg\}$ is a directed acyclic graph (DAG) with a root node (of out-degree zero), input gates labeled by variables (of in-degree zero) and the non-input gates (inclusive of root) labeled by functions in $\mathcal{B}$. Define the size to be the number of non-input gates and, depth to be the length of the longest path from root to any input gate of the circuit C denoted, respectively, as $\mathsf{Size}(C)$ and $\mathsf{Depth}(C)$. A Boolean formula is a Boolean circuit where the underlying DAG is a tree. We call a negation gate that takes input from a variable as a *leaf negation*. A circuit is said to be *monotone* if it does not use any negation gates. A function is monotone if it can be computed by a monotone circuit. Equivalently, a function f is monotone if $\forall\, x, y \in \{0,1\}^n$, $x \prec y \implies f(x) \leq f(y)$ where $x \prec y$ iff $x_i \leq y_i$ for all $i \in [n]$. For a circuit C, $\mathsf{negs}(C)$ denotes the number of NOT gates in the circuit C. Fix an arbitrary ordering among the gates of C. A *firing pattern* of a circuit C on a given input is the binary string of evaluation of the gates on the input as per the fixed ordering. The *number of firing patterns of a circuit* C is the number of distinct firing patterns for C over all inputs.

Let $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. For $i \in [n]$, let e_i denote the n length Boolean vector with the i^{th} entry alone as 1. For an $a \in \{0,1\}^n$, $a \oplus e_i$ denotes the input obtained by flipping the i^{th} bit of a. The positive sensitivity of f on a, denoted by $\mathsf{psens}(f, a)$, is the number of $i \in [n]$ such that $a_i = 1$ and $f(a \oplus e_i) \neq f(a)$. We define $\mathsf{psens}(f)$ as $\max_{a \in \{0,1\}^n} \mathsf{psens}(f, a)$.

For a monotone function $f : \{0,1\}^n \to \{0,1\}$, $x \in f^{-1}(1)$ and $y \in f^{-1}(0)$, define $S_f^+(x,y) = \{i \mid x_i = 1, y_i = 0, i \in [n]\}$. The monotone Karchmer-Wigderson cost of f (denoted by $\mathsf{KW}^+(f)$) is the optimal communication cost of the problem where Alice has x, Bob has y and they have to find an $i \in [n]$ such that $i \in S_f^+(x,y)$. It is known that $\mathsf{KW}^+(f)$ equals the minimum depth monotone circuit computing f. For more details about this model, see [10].

Energy Complexity : For a Boolean circuit C and an input a, the energy complexity of C on the input a (denoted by $\mathsf{EC}(C, a)$) is defined as the number of non-input gates that output a 1 in C on the input a. Define the *energy complexity* of C (denoted by $\mathsf{EC}(C)$) as $\max_a \mathsf{EC}(C, a)$. The energy complexity of a function f, (denoted by $\mathsf{EC}(f)$) is the energy of the minimum energy circuit over the Boolean basis $\mathcal{B}$ computing f.

As mentioned in the introduction, Lozhkin and Shupletsov [11] showed that $\mathsf{EC}(f) \leq 3n(1+\epsilon(n))$ by constructing a circuit of size $\frac{2^n}{n}(1+\epsilon(n))$ where $\epsilon(n) \to 0$ as $n \to \infty$. Their idea is to construct a Boolean circuit of low energy that outputs all product terms on n variables where each of them appear exactly once in a negated or unnegated form. We call such terms as *minterms*. We slightly improve the above bound using the same idea by constructing a circuit of size $2^{O(n)}$.

Proposition 2.1. *For any $f : \{0,1\}^n \to \{0,1\}$, $\mathsf{EC}(f) \leq 3n - 1$.*

Proof. We show that all minterms in n variables can be computed by a circuit of energy at most $2n-1$. Assuming this, to compute f, construct an $\vee$ formula on 2^n inputs of depth n and connect the minterms on which f evaluates to 1 to the $\vee$ leaves (and the rest of the inputs as 0). Since on any input, exactly one of the leaves will evaluate to 1, there is only 1 path to the output gate where all $\wedge$ gates evaluate to 1. Hence the overall energy complexity is at most $2n-1+n = 3n-1$. We construct a circuit of energy $2n-1$ to compute all minterms on n variables.

Proof is by induction on n. Let $x_1, \ldots, x_n$ be the variables. For $n = 1$, the circuit is x_1, $\neg x_1$ which has energy 1. Hence the base case holds.

By induction, we have constructed a circuit C (on n inputs and having 2^n outputs) computing all 2^n minterms on $x_1, \ldots, x_n$. We modify the circuit as follows : branch out each output gate into two (left and right branch). Connect the left (resp. right) branch output to x_{n+1} (resp. $\neg x_{n+1}$) by an $\wedge$ gate. Note that out of all 2^{n+1} outputs created this way, exactly one of them will output 1 on any input. Also we have computed all 2^{n+1} minterms on $x_1, \ldots, x_{n+1}$. The resulting circuit has an energy of $2n-1$ for circuit C by induction plus 2 due to the output and the negation gate of x_{n+1}. Hence overall energy is $2n+1 = 2(n+1)-1$. This completes the induction. $\square$

Observe that in a circuit C, for the leaf negation gates, there is always an input where all of them output a 1. For the non-leaf negation gates, irrespective of the input, either the negation gate or its input gate will output a one. Due to this reason, we have,

Proposition 2.2. *For any circuit C, $\mathsf{EC}(C) \geq \mathsf{negs}(C)$.*

Model specific variants of energy complexity : We now consider the notion of energy complexity for three other circuit models, namely monotone circuits, Boolean formulas and threshold circuits.

Energy Complexity and Monotone circuits : For a monotone Boolean function f, computed by a monotone circuit C, define $\mathsf{EC}^{\mathsf{M}}(C)$ as the maximum over all the inputs the number of non-input gates that output a 1. We define $\mathsf{EC}^{\mathsf{M}}(f)$ as $\min_C \mathsf{EC}^{\mathsf{M}}(C)$ where C is a monotone circuit computing f. The following proposition gives an exact characterization for $\mathsf{EC}^{\mathsf{M}}(f)$.

Proposition 2.3. *For a monotone Boolean function f, let $\mathsf{mSize}(f)$ denotes the size of the smallest monotone circuit computing f. Then, $\mathsf{EC}^{\mathsf{M}}(f) = \mathsf{mSize}(f)$.*

Proof. Let C be a monotone circuit of minimum size computing f. Clearly, $\mathsf{EC}^{\mathsf{M}}(f) \leq \mathsf{EC}^{\mathsf{M}}(C) \leq \mathsf{mSize}(f)$. Also, for any monotone circuit C' computing f, on the input 1^n, all the gates in C' output a 1 implying $\mathsf{EC}^{\mathsf{M}}(C') \geq \mathsf{EC}^{\mathsf{M}}(C', 1^n) = \mathsf{mSize}(C')$. In particular, for the monotone circuit C'' of minimum energy computing f, $\mathsf{EC}^{\mathsf{M}}(f) = \mathsf{EC}^{\mathsf{M}}(C'') \geq \mathsf{Size}(C'') \geq \mathsf{mSize}(f)$. Hence, $\mathsf{EC}^{\mathsf{M}}(f) = \mathsf{mSize}(f)$. $\square$

Energy Complexity and Threshold circuits : Let $\mathcal{T}$ be a basis consisting of all weighted threshold functions. A threshold circuit is a Boolean circuit where the gates are from the basis $\mathcal{T}$. Uchizawa *et al.* [18] introduced the notion of energy complexity of threshold circuits denoted by $\mathsf{EC}_{\mathcal{T}}(C)$, again defined as the worst energy of the threshold circuit C among all the inputs. Define $\mathsf{EC}_{\mathcal{T}}(f)$ as $\min_C \mathsf{EC}_{\mathcal{T}}(C)$ where C is a threshold circuit over the basis $\mathcal{T}$ computing f.

A *decision tree* is a rooted tree with all the non-leaf nodes labeled by variables and leaves labeled by a 0 or 1. Note that every assignment to the variable in the tree defines a unique path from root to leaf in the natural way. A Boolean function f is said to be computed by a decision tree if for every input a, the path from root to a leaf guided by the input is labeled by $f(a)$. Depth of a decision tree is the length of the longest path from root to any leaf. Define decision tree depth of f (denoted by $\mathsf{DT}(f)$) as the depth of the minimum depth decision tree computing f. A threshold decision tree is similar to the decision tree except that queries at each non-leaf node can be an arbitrary threshold function on the input variables. We denote the depth of the minimum depth threshold decision tree computing f by $\mathsf{DT}_{th}(f)$.

For an $f : \{0,1\}^n \to \{0,1\}$, Uchizawa *et al.* [18] introduced a measure of energy for threshold decision tree T computing f, denoted by $cost(T)$ defined as the maximum over all paths from root to leaf, the number of right turns taken in a path. As a part of their main result they showed, given any threshold decision tree T, (1) how to construct another threshold decision tree T' with a bound on $cost(T')$ (Lemma 2, 3 of [18]) and (2) how to obtain a low energy threshold circuit C' computing f from T (Lemma 5 of [18]).

This implies the following relation between $\mathsf{EC}_{\mathcal{T}}(f)$ and $\mathsf{DT}_{th}(f)$.

Proposition 2.4. *For any Boolean function f, $\mathsf{EC}_{\mathcal{T}}(f) \leq \mathsf{DT}_{th}(f) + 1$*

Proof. Let T be an optimum threshold decision tree computing f with depth $\mathsf{DT}_{th}(f)$. We first state the results (1) and (2) formally. Result (1) says that f can be computed by another threshold decision tree T' with same depth as T, same number of leaves as T and have $cost(T') \leq \log(\#\text{ leaves of } T)$ (Lemma 2, 3 of [18]). Result (2) says that there exists a threshold circuit C computing f with $\mathsf{EC}_{\mathcal{T}}(C) \leq cost(T) + 1$ (Lemma 5 of [18]).

Since T has at most $2^{\mathsf{DT}_{th}(f)}$ many leaves, applying (1), we get a threshold decision tree T' with $cost(T') \leq \log(\#\text{ leaves of } T) \leq \mathsf{DT}_{th}(f)$. The result now follows by applying (2) to T'. □

Energy Complexity and Formulas: For a Boolean formula F, define $\mathsf{EC}^{\mathsf{F}}(F)$ is the worst case energy complexity of the formula F over the Boolean basis $\mathcal{B}$. We define, $\mathsf{EC}^{\mathsf{F}}(f)$ as $\min_F \mathsf{EC}^{\mathsf{F}}(F)$ where F is formula (over the Boolean basis $\mathcal{B}$) computing f. See Section 5 for more details.

3 Energy Complexity and Decision Trees

In this section, we show new techniques to obtain upper bounds and (weak) lower bounds on $\mathsf{EC}(f)$. In the upper bound front, we show that a Boolean function f with a low $\mathsf{DT}(f)$ can be computed by a low energy circuit (Section 3.1). We then show how to construct decision trees for a Boolean function given a circuit computing the function and its firing patterns. This implies the following weak converse to the above statement : if $s = \mathsf{Size}(f)$ and $e = \mathsf{EC}(f)$, then $s^e = \Omega(\mathsf{DT}(f))$ (Section 3.2).

3.1 Energy Upper Bounds from Decision Trees

We know that any n bit function f can be computed by a circuit of energy at most $3n - 1$ (Proposition 2.1). In this section, we identify the property of having low depth decision trees as a sufficient condition to guarantee energy efficient circuits. More precisely, we show that for any Boolean function f, $\mathsf{EC}(f) \leq O(\mathsf{DT}(f)^3)$.

One of the challenges in constructing a Boolean circuit is to use as few negation gates as possible. The reason is that non-leaf negation gates always contribute to the energy since either the gate or its input will always output a 1 on any input to the circuit. We achieve this in our construction via an idea inspired by the *connector circuit* introduced by Markov [12]. Before describing the construction, we need the following result (Lemma 3.1) which helps in controlling the number of negation gates in our construction.

Lemma 3.1. *Let f_0 and f_1 be any two Boolean functions on n variables computed by Boolean circuits C_0 and C_1 respectively. Fix an $i \in [n]$. Define $f(x) = (\neg x_i \wedge f_0(x)) \vee (x_i \wedge f_1(x))$. Then, a circuit C computing f can be obtained using C_1 and C_2 such that $\mathsf{negs}(C) = 1 + \max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\}$.*

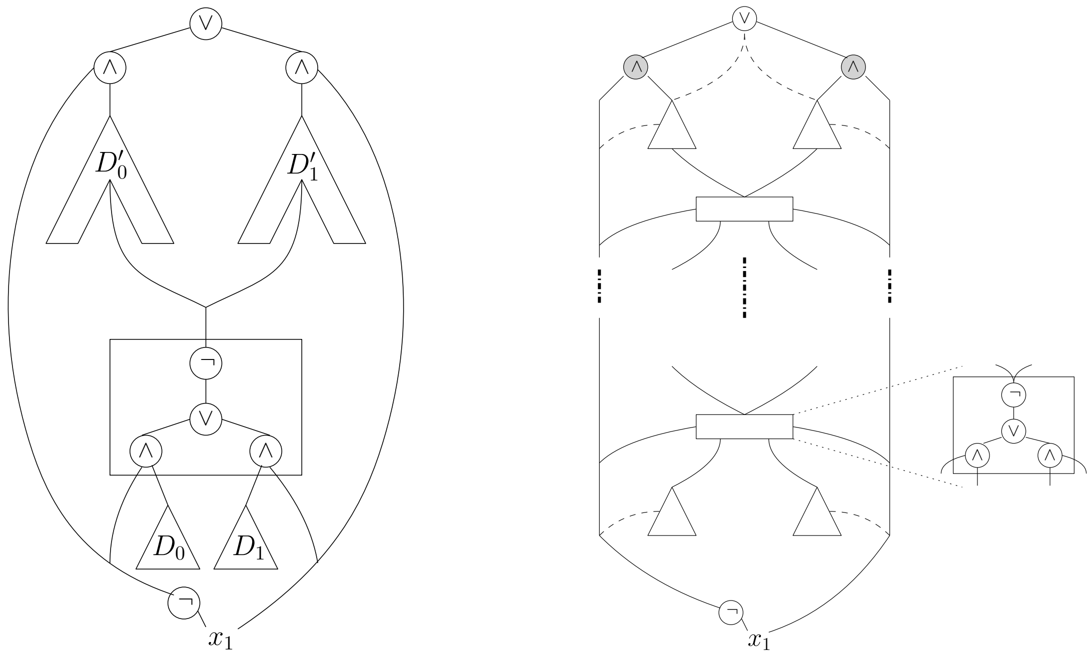
Note that the existence of the circuit in Lemma 3.1 can be shown by the result of Markov [12] (see Section 10.2 of Jukna [8]). However, the construction using the result of Markov is inherently inductive and can potentially have high energy. Since, the Boolean function f which we intent to compute is structured, we take advantage of this observation to give a construction of low energy circuit which is then used to prove the main result of this section (Theorem 1.1).

Proof of Lemma 3.1. We start with the circuit $A = (\neg x_i \wedge C_0(x)) \vee (x_i \wedge C_1(x))$ which uses $1 + \mathsf{negs}(C_0) + \mathsf{negs}(C_1)$ negations to compute f. If one of $\mathsf{negs}(C_0)$ or $\mathsf{negs}(C_1)$ is zero, then A is the required circuit. Otherwise, we modify this circuit in $\min\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\}$ steps where, in each step, we reduce the number of negations by 1 such that the resulting circuit computes f correctly. Hence the resulting circuit C has $1 + \mathsf{negs}(C_0) + \mathsf{negs}(C_1) - \min\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\} = 1 + \max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\}$ negations.

We describe the modifications starting with A. Let g_0 be a gate in C_0 that feeds into a negation gate such that the function computed at g_0 does not depend on output of any negation gate. Let D_0 be the sub-circuit rooted at g_0. Similarly, let g_1 be a gate in C_1 with the similar property and let D_1 be the sub-circuit rooted at g_1. We remove the negation gates that g_0 and g_1 feeds into from C_0 and C_1 respectively and construct the *connector circuit* (as shown in the box in Fig. 1a). We feed the output from the selector as output of the negation gates in C_1 and C_2. Let D'_0 (resp. D'_1) be the circuit C_0 (resp. C_1) with the output of selector circuit acting as output of the negation gate with the negation gate alone removed (Note that we do not completely disconnect the sub-circuits from the circuit. The wires connecting D_0 (resp. D_1) to D'_0 (resp. D'_1) are not shown in Fig. 1a to avoid clutter).

When $x_i = 0$, we claim that this circuit outputs $C_0(x)$. This is because when $x_i = 0$, D'_0 gets $\neg D_0$ as output of g_0 correctly and hence computes $C_0(x)$ while the output of D'_1 is inhibited by the $\wedge$ gate which it feeds into. By a similar argument, this circuit computes $C_1(x)$ when $x_i = 1$. Hence the resulting circuit indeed computes f correctly.

Observe that the number of negation goes down by one in each step since we replace two negations by one. We repeat the previous steps restricted to gates in D'_0 and D'_1 as long as the negations in at least one of the circuits is exhausted. By the earlier argument, the final circuit C correctly computes f. $\square$



(a) First step in the construction of C in the proof of Lemma 3.1.

(b) Circuit C' in the proof of Lemma 3.2

Figure 1: Energy efficient circuit construction

When $x_1 = 0$ (resp. $x_1 = 1$) the part of the circuit computing f_1 (resp. f_0) is not necessary in computing f. Having obtained a circuit construction which minimizes the usage of negations, we need a way to "turn off" such gates that are not needed in computing f. In Lemma 3.2, we demonstrate how this is achieved, thereby saving energy, at the cost of increasing the fan-in of $\wedge$ gates.

Lemma 3.2. *For any non-constant Boolean function f, there exists a circuit C computing f with, (1) all $\vee$ gates are of fan-in 2 and all $\wedge$ gates are of fan-in at most $\mathsf{DT}(f) + 2$, (2) no $\vee$ gate have a negation gate or a variable directly as its input, (3) $\mathsf{negs}(C) \leq \mathsf{DT}(f)$ and, (4) $\mathsf{EC}(C) \leq 2\mathsf{DT}(f)^2$.*

Proof. We describe the construction of the circuit by an induction on $\mathsf{DT}(f)$.
Base Case : For f with $\mathsf{DT}(f) = 1$, f is either a variable or its negation and hence the trivial circuit satisfies (1) to (4). For $\mathsf{DT}(f) = 2$, let T be an optimal decision tree with x_1 as its root. Then, f can be computed by the circuit $C = (\neg x_1 \wedge \ell_1) \vee (x_1 \wedge \ell_2)$ where ℓ_1, ℓ_2 could be a variable, negation of a variable or a constant. Also if C has 3 negations, we use Lemma 3.1 to get a circuit with two negations. Hence condition (3) is satisfied. In either cases, the conditions (1) and (2) are also satisfied and it can be verified that the energy of the circuit is at most $5 \leq 2\mathsf{DT}(f)^2$. Hence condition (4) is also satisfied and the base case holds.
Inductive Step : Let f be a Boolean function computed by a decision tree T of depth $\mathsf{DT}(f) \geq 3$. By induction, assume that for any Boolean function g with $\mathsf{DT}(g) \leq \mathsf{DT}(f) - 1$ there exists a circuit C computing g satisfying (1) to (4). Let the root variable of T be x_1 and T_0 (resp. T_1) be the left (resp. right) subtree computing the function $f_0 = f|_{x_1=0}$ (resp. $f_1 = f|_{x_1=1}$). Since f_0 and f_1 are computed by decision trees of depth $\mathsf{DT}(f) - 1$, by induction, there exists circuits C_0 and C_1 computing f_0

and f_1, respectively, satisfying (1) to (4).

Observe that $f(x) = (\neg x_1 \wedge f_0) \vee (x_1 \wedge f_1)$. Hence by Lemma 3.1, there exists a circuit C computing f (Fig. 1b omitting the thinly dashed lines) with $\mathsf{negs}(C) = \max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\} + 1$. We modify the circuit C as follows : for each $\wedge$ gate which was originally in C_0 (resp. C_1), we add $\neg x_1$ (resp. x_1) as input thereby increasing its fan-in by 1. We also remove the $\wedge$ gate (shaded in Fig. 1b) feeding into the top $\vee$ gate and feed the output of the circuits directly to the top $\vee$ gate (shown as dashed in Fig. 1b). Call the resulting circuit C' and the gates from C_0 as C'_0 (the left part in Fig. 1b) and the gates from C_1 as C'_1 (the right part in Fig. 1b).

We first argue that the conditions (1) and (2) holds true for C'. We then argue that C' correctly computes f using which we argue (3) and (4) thereby completing the induction.

We observe that the condition (1) holds since $\vee$ gate has fan-in 2 by construction and $\wedge$ gate has fan-in at most $\max\{\mathsf{DT}(f_0)+3, \mathsf{DT}(f_1)+3\}$ which is at most $\mathsf{DT}(f)+2$. The removal of the shaded $\wedge$ gates never causes a variable or a negation to be fed to the top $\vee$ gate since f_0 and f_1 have a decision tree depth of at least 2 and hence the circuits of the respective functions have top gate as $\vee$ which is guaranteed by base case for depth 2 and by induction otherwise. Hence condition (2) holds. We now argue that C' correctly computes f. When $x_1 = 1$, all the $\wedge$ gates in C'_0 evaluates to 0. Since no input variable or negation gate feeds into any $\vee$ gate in C'_0 (condition (2)), all the $\vee$ gates and $\wedge$ gates output 0 irrespective of the remaining input bits. Hence the C'_0 outputs 0. Since $x_1 = 1$, C'_1 behaves exactly same as C_1. By Lemma 3.1, the circuit C_1 correctly computes f when $x_1 = 1$. Hence the circuit C' correctly computes f for $x_1 = 1$. The same argument with C_0 and C_1 interchanged can be used to show that C' correctly computes f with $x_1 = 0$.

Since no new negations were added in C', $\mathsf{negs}(C') = \mathsf{negs}(C)$ which, by Lemma 3.1, equals $\max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\} + 1 \le \max\{\mathsf{DT}(f_1), \mathsf{DT}(f_2)\} + 1 \le \mathsf{DT}(f)$. Hence condition (3) holds. We show that condition (4) holds for C'. Let x be an input with $x_1 = 1$. We have already argued that when $x_1 = 1$, none of the $\wedge$ or $\vee$ gates of C'_0 output a 1. Hence the gates that can output a 1 in C' are the negations in C'_0, the gates that output 1 in C'_1, the selector gates (in the construction of Lemma 3.1), the root gate and the negation gate for x_1 (recall that the shaded $\wedge$ gates are removed). Observe that the negations in C'_0 is at most $\mathsf{negs}(C_0)$ and C'_1 behaves exactly as C_1 for $x_1 = 1$. Also the number of selector circuits used in Lemma 3.1 is at most $\max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\}$.[5] Also each such circuit can have at most 2 gates that output 1 on any input (see Fig. 1b). Hence, $\mathsf{EC}(C', x) \le \alpha_0 = \mathsf{negs}(C_0) + \mathsf{EC}(C_1) + 2\max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\} + 2$. For x with $x_1 = 0$, by a similar argument, $\mathsf{EC}(C', x) \le \alpha_1 = \mathsf{negs}(C_1) + \mathsf{EC}(C_0) + 2\max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\} + 2$. Hence, $\mathsf{EC}(C') \le \max\{\alpha_0, \alpha_1\}$ which is at most $\max\{\mathsf{EC}(C_0), \mathsf{EC}(C_1)\} + 3\max\{\mathsf{negs}(C_0), \mathsf{negs}(C_1)\} + 2$. By induction, we have $\mathsf{EC}(f) \le 2(\mathsf{DT}(f)-1)^2 + 3(\mathsf{DT}(f)-1) + 2$ which implies $\mathsf{EC}(f) \le 2\mathsf{DT}(f)^2$ as f is non-constant. This completes the induction. ∎

We prove the main result of this section.

Theorem 1.1. For any Boolean function f, $\mathsf{EC}(f) \le O(\mathsf{DT}(f)^3)$.

Proof. If f is constant, the result holds. Otherwise, applying Lemma 3.2, we have a circuit C' computing f with fan-in of $\vee$ gate being 2 and fan-in of $\wedge$ gate being at most $\mathsf{DT}(f)+2$ of energy at most $2\mathsf{DT}(f)^2$. Without loss of generality, let x_1 be the variable at the root of decision tree. By construction, all the unbounded fan-in $\wedge$ gates of the circuit C' have x_1 or $\neg x_1$ as an input.

[5] While this quantity should be the minimum of the negations of C_1 and C_2, as seen in the proof of Lemma 3.1, we upper bound this by the maximum of negations.

To obtain a bounded fan-in circuit from C', we replace each of the $\wedge$ gates by a fan-in 2 circuit as follows. Let g be a $\wedge$ gate of the circuit C' of fan-in $c \leq \mathsf{DT}(f) + 2$ which takes in $\ell \in \{x_1, \neg x_1\}$ as one of its input. We replace g by a tree of fan-in 2 $\wedge$ gates of c leaves and of depth $c - 1$ with ℓ as a leaf at depth $c - 1$ as shown in Fig. 2.

(a) $\wedge$ gate of fan-in c (for $c = 4$)

(b) Tree of depth $c - 1$

Figure 2: Handling $\wedge$ gates of large fan-in

We now argue that this replacement can only increase the overall energy by a factor of at most $c - 1$. Consider an input for which $\ell = 0$. Then, irrespective of the values of other $c - 1$ inputs, none of the fan-in 2 $\wedge$ gates output a 1 as ℓ forces all $\wedge$ gates to output 0. On the other hand if $\ell = 1$, then the added gates can contribute an energy of at most $c - 1$. Hence for any input, the $\wedge$ gates in C' that output a 0 does not contribute any energy and those that output a 1 can contribute of an energy of at most $c - 1 \leq \mathsf{DT}(F) + 1$. Since, in the worst case, all the gates that output a 1 can be an $\wedge$,

$$\mathsf{EC}(f) \leq \mathsf{EC}(C) \cdot (\mathsf{DT}(f) + 1) \leq 2\mathsf{DT}(f)^2 \cdot (\mathsf{DT}(f) + 1) = O(\mathsf{DT}(f)^3)$$

□

3.2 Decision Trees from Bounded Firing Patterns

Given Theorem 1.1, a natural question to ask is whether for all Boolean functions f, $\mathsf{DT}(f) \leq \mathsf{poly}(\mathsf{EC}(f))$. We are not aware of an explicit counter example to this. We prove a weaker version of this converse. More precisely, we show that for any Boolean function f, and a circuit C over an arbitrary finite basis (of finite arity), $\mathsf{DT}(f)$ is at most the number of firing patterns of C. This is weak since a circuit with s internal gates and energy e can potentially have $\sum_{i=0}^{e} \binom{s}{i} \approx s^e$ firing patterns. Albeit being a weak converse, this immediately implies a size energy trade-off for Boolean function with high decision tree depth (Corollary 3.4).

Lemma 3.3. *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, let C be any Boolean circuit computing f having gates computing an arbitrary function of a finite arity. Then, $\mathsf{DT}(f)$ is, asymptotically, at most the number of firing patterns of C.*

Proof. Let the number of firing patterns of C be t and ℓ be the maximum arity of gates in C. We show that there exists a decision tree computing f of depth $\ell \cdot t$. Since ℓ is a finite constant, $t \geq \mathsf{DT}(f)/\ell = \Omega(\mathsf{DT}(f))$.

Proof is by strong induction on n. For $n = 1$, $\mathsf{DT}(f) \leq 1$ and there must be at least one firing pattern for C. Hence $\mathsf{DT}(f) \leq \ell \cdot t$. Suppose the claim holds for all Boolean functions on $< n$ variables. Let f be an n bit Boolean function computed by a circuit C of size s with gates of fan-in at most ℓ. For the circuit C, let there be t distinct firing patterns $p_1, p_2, \ldots, p_t$ where each $p_i \in \{0,1\}^s$. Let C' be the circuit obtained from C by removing all the gates that have the same value in all the firing patterns. Observe that this transformation does not alter the number of firing patterns and let $p'_1, p'_2, \ldots, p'_t$ be the firing patterns of C'. Let g be a gate in C' whose evaluation depends only on input variables. The decision tree queries all the inputs to this gate. Let f' be the function f after setting the queried variables to the values read. Also set the queried values in C' and evaluate the circuit (as far as possible) to get C'' which computes f'. Since f' is on $\leq n - \ell$ variables, by induction, $\mathsf{DT}(f') \leq \ell \times$ Number of firing patterns of C''.

Since the value of gate g is fixed, C'' can have at most $t-1$ firing patterns (for otherwise, all the firing patterns have the same value for gate g due to which g would have been removed in C', a contradiction). Hence $\mathsf{DT}(f) \leq \mathsf{DT}(f') + \ell \leq \ell \times$ Number of firing patterns of $C'' + \ell \leq \ell \cdot (t-1) + \ell = \ell \cdot t$. □

We describe an application of this converse in obtaining size-energy trade-offs in Boolean circuits. Recalling from Table 1, Uchizawa *et al*. [21] showed that for any symmetric function f computed by a unate circuit C, the number of firing patterns is lower bounded by $(n+1-a_f)/b_f$. On the other hand, for any circuit C (not necessarily unate) of size s and energy e, the number of firing patterns of C is at most $s^e + 1$ [21]. This immediately implies a size-energy trade-off for unate circuits computing symmetric functions. Since, any Boolean circuit can be converted to a unate circuit of same size, the same trade-off holds for Boolean circuits also.

The above converse (Lemma 3.3) implies a similar trade-off result for a more general class of Boolean functions, namely functions with large decision tree depth.

Corollary 3.4. *For any Boolean function f and a circuit C of size s and energy e computing f over an arbitrary finite basis $\mathcal{B}$, $s^e \geq \Omega(\mathsf{DT}(f))$.*

4 Lower Bounds on Energy Complexity

In this section, we introduce new methods to show lower bounds on energy complexity of Boolean functions. We introduce the notion of continuous positive paths (Section 4.1.1) using which we prove two energy lower bounds. Firstly, we show that the positive sensitivity of a function is a lower bound on its energy complexity (Section 4.1.2). Secondly, we show that for monotone Boolean functions, the cost of the monotone Karchmer- Wigderson game for the function is a lower bound on its energy complexity (Section 4.2). We conclude the section by proving an energy lower bound of $\Omega(n)$ for any depth 3 unbounded fan-in circuit computing parity function on n bits (Section 4.3).

4.1 Energy Lower Bounds from Positive Sensitivity

In this section, we prove Theorem 1.2 from the introduction. We first describe an outline here. As a starting case, consider a monotone circuit C computing f evaluates to 1 on an input $a \in \{0,1\}^n$. Let $i \in [n]$ be such that $a_i = 1$ and flipping a_i to 0 causes the circuit to evaluate to 0. We show that for such an index i on input a, there is a path from x_i to the root such that all the gates in the path outputs a 1. The latter already implies a weak energy lower bound. We then generalize this idea to non-monotone circuits as well and use it to prove energy lower bounds. This generalization also helps us to prove upper bounds for KW^+ games in Section 4.2.

To keep track of all input indices that are sensitive in the above sense, we introduce the measure of positive sensitivity denoted by $\mathsf{psens}(f)$ (as defined in Section 2). For example, the functions $f \in \{\oplus_n, \wedge_n\}$ have $\mathsf{psens}(f) = n$ while $\mathsf{psens}(\vee_n) = 1$. Let $\widetilde{\mathsf{psens}}(f, a)$ denote the set of positive sensitive indices on a.

4.1.1 Continuous Positive Paths

Let C be a Boolean circuit computing $f : \{0,1\}^n \to \{0,1\}$. For an input $a \in \{0,1\}^n$, we call a path of gates such that every gate in the path output 1 on a as a *continuous positive path* in C.

Fix an $a \in \{0,1\}^n$. We argue that for every positive sensitive index i on a, either there is a continuous positive path from x_i to the root or it must be broken by a negation gate of the circuit. Using this we show, in the next section, that energy complexity of a function is lower bounded by its positive sensitivity.

Lemma 4.1. *Let $f : \{0,1\}^n \to \{0,1\}$ and $a \in \{0,1\}^n$ be an input such that $\mathsf{psens}(f, a) \neq 0$ and $i \in \widetilde{\mathsf{psens}}(f, a)$. Let C be any circuit computing f. Then, either (1) there is a continuous positive path from x_i to root or (2) there is a continuous positive path from x_i to a gate which feeds into a negation gate of C.*

Proof. It suffices to prove the following stronger statement : for a Boolean function f and an $a \in \{0,1\}^n$ with $\mathsf{psens}(f, a) \neq 0$ and $i \in \widetilde{\mathsf{psens}}(f, a)$, let C be any circuit such that $C(a) = f(a)$ and $C(a \oplus e_i) = f(a \oplus e_i)$. Then, either (1) there is a continuous positive path from x_i to root or (2) there is a continuous positive path from x_i to a gate which feeds into a negation gate of C.

Proof is by induction on $\mathsf{negs}(C)$. Let C be any circuit such that $f(a) = C(a)$ and $f(a \oplus e_i) = C(a \oplus e_i)$.

Base Case : For the base case, $\mathsf{negs}(C) = 0$ and C is a monotone circuit. By definition, $i \in \mathsf{psens}(f, a)$ implies that $a > a \oplus e_i$. We claim that if $i \in \mathsf{psens}(f, a)$, then $C(a) = 1$. For a contradiction, suppose that $f(a) = C(a) = 0$. Then $C(a \oplus e_i) = 0$ because C is monotone. But then $f(a \oplus e_i) = 0$ which contradicts the fact that $i \in \mathsf{psens}(f, a)$.

Since $C(a)$ outputs 1 and since C is a monotone circuit, the root being an $\vee$ or $\wedge$ gate must have a child gate evaluating to 1. Since this gate is again $\vee$ or $\wedge$ the same argument applies and we get a series of gates all evaluating to 1 reaching some inputs. For any $i \in \mathsf{psens}(f, a)$, we show that there must a path from x_i to the root with all the gates in the path evaluating to 1 in C on input a. For a contradiction, suppose that every path from x_i to the root gate passes via some gate that evaluates to 0. Among all the paths from x_i to the root, collect all the gates that evaluate to 0 for the first time in the path and call this set as T. We fix all the variables except x_i to the values in a and view each of the gates in $g \in T$ as a function of x_i. Now, flipping x_i from $a_i = 1$ to 0 does not change the output of any $g \in T$ as they compute monotone functions and already evaluate to 0. Since all other values are fixed, the output of the root gate does not change by this flip which contradicts the fact that $i \in \mathsf{psens}(f, a)$.

Induction Step : Let C be a circuit with $f(a) = C(a)$ and $f(a \oplus e_i) = C(a \oplus e_i)$ and $\mathsf{negs}(C) \geq 1$. Let g be the first gate that feeds into a negation in the topologically sorted order of the gates of C.

We have the following two possibilities. In both the cases, we argue existence of continuous positive path in C from the variable x_i, thereby completing the induction.

- *On input a, flipping a_i change the output of g.* Denote the function computed at g as f_g. Then f_g is monotone and $i \in \widetilde{\mathsf{psens}}(f_g, a)$ and is non-empty. Hence applying the argument in the base case to f_g and the monotone circuit rooted at g, we are guaranteed to get a continuous positive path from x_i to g. Since the circuit at g is a sub-circuit of C (that is, it appear as an induced subgraph), this gives a continuous positive path in C also.

- *On input a, flipping a_i does not change the output of g.* In this case, we remove the negation gate that g feeds into and hard wire the output of this negation gate (on input a) in C to get a circuit C'. Note that all other gates in C are left intact. Observe that $C'(a) = f(a)$. Since flipping a_i did not change the output of g and as all other gates are left intact, $C'(a \oplus e_i) = f(a \oplus e_i)$. As $\mathsf{negs}(C') = \mathsf{negs}(C) - 1$, by induction, either (1) there is a continuous positive path from x_i to root or (2) there is a continuous positive path from x_i to a gate which feeds into a negation gate of C'. By construction, C' is same as C except for the negation gate. Hence a continuous positive path in C' is also a continuous positive path in C.

□

4.1.2 From Positive Sensitivity to Energy Lower Bounds

We call the negation gates and the root gate of a circuit as *target gates*. In Lemma 4.1, we have already shown the existence of continuous positive paths from a positive sensitive index up to a target gate. Using this, we show an energy lower bound for any circuit of bounded fan-in computing a Boolean function f in terms of $\mathsf{psens}(f)$. Since the fan-in of the circuit is limited, we exploit the idea that in a connected DAG, the number of internal nodes (in-degree at least 1) is lower bounded by the number of source nodes (in-degree 0).

Since every such positive sensitive index is reachable via a continuous positive path from a target gate, we obtain a lower bound on energy by applying this idea on an appropriate subgraph constructed from our circuit.

Theorem 1.2. For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ computed by a circuit C over the Boolean basis, $\mathsf{EC}(C) \geq \mathsf{psens}(f)/3$.

Proof. Without loss of generality assume that f is non-constant. Let C be any circuit computing f of fan-in 2 such that $\mathsf{EC}(C) = \mathsf{EC}(f)$. We prove that $\forall a \in \{0,1\}^n, \mathsf{psens}(f,a) \leq 3\mathsf{EC}(C)$.

Let $a \in \{0,1\}^n$ by any input. If $\mathsf{psens}(f,a) = 0$, the claim holds. Hence we can assume, $\mathsf{psens}(f,a) \neq 0$. Let T be the set of all target nodes in C. For every $i \in \widetilde{\mathsf{psens}}(f,a)$, by Lemma 4.1, there exists continuous positive paths starting from x_i to a gate $g \in T$.

For every $g \in T$, let X_g be the set of all gates that lie in a continuous positive path from an x_i to g for some $i \in \widetilde{\mathsf{psens}}(f,a)$. Note that the subgraph induced by vertices in X_g is connected and does not include g. We now obtain a connected DAG with $\mathsf{psens}(f,a)$ leaves as follows. Let D be a full binary tree (with edges directed from child to parent) with $|T|$ many leaves and hence $|T| - 1$ internal nodes. For each $g \in T$ if it is a negation, we attach the gate feeding into g as a leaf of the D and if it is a root, we attach the root as a leaf of the D. Call the resulting DAG as H.

Since graph induced on X_g is connected for each g, this gives us a connected DAG on $\mathsf{psens}(f,a)$ many source nodes. Let $X = \cup_{g \in T} X_g$. Observe that the number of internal nodes is $|X| + (|T| - 1) + 1$ where the first term is the gates in X, the second term is the number of internal nodes of the tree and third term is due to the root. Since the target gates include negations and the root, $|T| = \mathsf{negs}(C) + 1$. Since the total number of negation gates in any circuit computing f is at most $\mathsf{EC}(f)$ (Proposition 2.2), we get that number of internal nodes of H is at most

$|X| + |T| - 1 \leq |X| + \mathsf{EC}(f) + 1 - 1 \leq 2\mathsf{EC}(f)$. Since the resulting DAG is connected, the number of leaves, which is $\mathsf{psens}(f,a)$, is at most the number of internal nodes $+1$ which is at most $2\mathsf{EC}(f) + 1 \leq 3\mathsf{EC}(f)$ as f is non-constant. □

This implies that since $\mathsf{psens}(\wedge_n) = n$, for $\mathsf{EC}(\wedge_n) \geq n/3$ which is asymptotically tight by Proposition 2.1. We observe that even thought $\wedge_n$ is symmetric, the result of Suzuki *et al.* [16] on the energy lower bound on threshold circuits computing symmetric functions (which applies to Boolean circuits too), only yields a trivial lower bound (see Table 1). We remark that both these bounds does not give any non-trivial energy lower bound for $f = \vee_n$. Note that Theorem 1.2 uses the fact that the circuits used have fan-in 2. If fan-in of the circuit C is c, then replacing each gate by a tree of $c-1$ gates of fan-in 2, by a similar argument as before, $\mathsf{EC}(C) \geq \mathsf{psens}(f)/(c+1)$.

4.2 Energy Lower Bounds from Karchmer-Wigderson Games

Proposition 2.3 says that the monotone circuits are not energy efficient for computing monotone functions. In this section, we explore the limits on how energy efficient can non-monotone circuits be in computing monotone functions by showing the following.

Theorem 1.3. Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. Then $\mathsf{EC}(f) = \Omega(\mathsf{KW}^+(f))$.

Our approach is to use Lemma 4.1 and utilize the existence of continuous positive paths to design a KW^+ protocol of cost $\mathsf{EC}(C)\log(\text{fan-in}(C))$ in Lemma 4.2 which immediately proves the above theorem. For the perfect matching function f_{PM} on a graph of n edges since $\mathsf{KW}^+(f_{PM}) = \Omega(\sqrt{n})$ [14], this implies that any circuit C with constant fan-in computing f_{PM} require an energy of $\Omega(\sqrt{n})$.

Recall that for $x \in f^{-1}(1)$ and $y \in f^{-1}(0)$, $S_f^+(x,y) \stackrel{\text{def}}{=} \{i \mid x_i = 1, y_i = 0, i \in [n]\}$. Also, we call the set of all negation gates, along with the root gate of C as the *target gates* of C.

Lemma 4.2. *For a non-constant monotone Boolean function f, let Alice and Bob hold inputs $a \in f^{-1}(1)$ and $b \in f^{-1}(0)$ respectively. Let C be any circuit computing f, and every gate in the circuit is either a $\wedge$, $\vee$ with fan-in of at most c or a negation gate. Then,* $\mathsf{KW}^+(f) \leq \mathsf{EC}(C)\log c$.

Proof. We argue that, without loss of generality it can be assumed that $\widetilde{\mathsf{psens}}(f,a) = \{i \mid a_i = 1\}$. Alice finds an $a' \prec a$ with $f(a') = f(a) = 1$ such that for any $a'' \prec a'$, $f(a'') = 0$. Observe that $a' \neq 0^n$ for otherwise, $f(0^n) = 1$ and since f is monotone, f must be a constant which is a contradiction. By construction, every bit in a' which is 1 is sensitive. Since $a' \prec a$, $S_f^+(a',b) \subseteq S_f^+(a,b)$, thereby it suffices to find an index in $S_f^+(a',b)$.

We now describe the protocol. Let $a \in f^{-1}(1)$ such that $\widetilde{\mathsf{psens}}(f,a) = \{i \mid a_i = 1\}$. Before the protocol begins, Alice does the following pre-computation. Let $\mathcal{P}$ be the collection of positive paths one each for every $i \in \widetilde{\mathsf{psens}}(f,a)$, which exists as per Lemma 4.1. Alice computes $\mathcal{P} = \bigcup_{g \in T} \mathcal{P}_g$ where $\mathcal{P}_g$ is the collection of all continuous positive paths ending at the target gate g. This ends the pre-processing.

Now Alice and Bob fixes an ordering of the target gates. For each target gate $g \in T$ in the order, the following procedure is repeated. For each continuous positive path $p \in \mathcal{P}$, ending at g, Alice sends the address of the previous gate in the path p (using $\log c$ bits) until they trace back to an input index i. Now, Bob checks if $b_i = 0$, and if so, we have found $i \in S_f^+(a,b)$, else, they attempt on the next $p \in \mathcal{P}_g$.

We argue about the correctness of the protocol. Notice that the above protocol searches through all $i \in \widetilde{\mathsf{psens}}(f,a)$ by traversing through all $\mathcal{P}_g$, for $g \in T$. Since $\widetilde{\mathsf{psens}}(f,a) = \{i \mid a_i = 1\}$ and $S_f^+(a,b) \subseteq \widetilde{\mathsf{psens}}(f,a)$ the protocol correctly computes i such that $a_i = 1$ and $b_i = 0$. Since the protocol visits only those gates that output 1 on the input a, we have a protocol with communication cost $\leq \mathsf{EC}(C,a) \times \log(c) \leq \mathsf{EC}(C)\log c$. □

4.3 Energy Lower Bounds for Depth Three Circuits

We now consider lower bounds on the energy complexity of constant depth circuits computing the parity function on n bits. For any Boolean function f, the trivial depth 2 circuit of unbounded fan-in computing f has an energy $n+2$ and it can be shown that any depth two circuit computing the parity on n bits require an energy of $n+1$.

Proposition 4.3. *Let C be any depth 2 circuit computing $\oplus_n$. Then $\mathsf{EC}(C) \geq n+1$.*

Proof. Since C computes $\oplus_n$, no variable or its negation can feed into the root gate, and every variable or its negation must feed into all gates at depth 2.

We now argue that at least $n-1$ variables must be negated in C. Suppose not, then there must be two variables, say x_i and x_j, that feeds into all the gates in depth 2 unnegated. Setting $x_i = 0$, all the $\wedge$ gates at depth 2 must evaluate to 0. Similarly, setting $x_j = 1$, causes all $\vee$ gates in depth 2 to evaluate to 1. Hence the circuit C evaluates to a fixed value irrespective of the remaining $n-2$ inputs unset which is a contradiction. Thus we conclude that at least $n-1$ variables must be negated. Consider an input x that is 0 on these $n-1$ negated variables and 1 on the remaining variable. On this input, all the negation gates, the $\wedge$ gate which they feed into and the root gate evaluates to 1. Hence $\mathsf{EC}(C) \geq \mathsf{EC}(C,x) \geq n-1+1+1 = n+1$. □

Santha and Wilson [15] showed that for any unbounded fan-in circuit C of depth d computing $\oplus_n$, $\mathsf{negs}(C) \geq d(\lceil n/2 \rceil)^{1/d} - d$. Since energy complexity of a circuit C is at least the number of negation gates in C (Proposition 2.2), this implies that $\mathsf{EC}(C) \geq d(\lceil n/2 \rceil)^{1/d} - d$ for any such circuit C computing $\oplus_n$.

While we are unable to prove strong lower bounds for circuits of depth d for an arbitrary constant d, we show that any depth $d = 3$ unbounded fan-in circuit computing the parity function requires large energy. We achieve this by appealing to the known lower bounds on size of any constant depth circuit computing $\oplus_n$. Razbarov (also independently by Håstad [7]) showed that any circuit C of depth d of unbounded fan-in computing parity on n bits must be of size at least $2^{\Omega(n^{1/4d})}$ [1]. Using this result we show an energy lower bound of $\Omega(n)$ for any depth 3 circuit computing $\oplus_n$.

Theorem 1.4. Let C be any unbounded fan-in circuit of depth 3 computing the parity function on n variables. Then, $\mathsf{EC}(C) = \Omega(n)$.

Proof. We call the root gate of the circuit as the "top" level and the two level immediately below as the "middle" and "bottom" levels respectively. Note that negations can appear anywhere in the circuit and does not count towards the level. Assume without loss of generality that the circuit C does not have any redundant gates.

Let there be i negated input variables and without loss of generality assume $i < n$. We set these variables to 0 and let C' be the resulting circuit obtained. Let $g_1, g_2, \ldots, g_k$ be the k gates in the bottom layer that feeds to the layers above via negation gates. We set input variables to these k gates such that the output of the negations are fixed in the following way : for the gate g_i, consider any input variable, say x_j, that feeds into g_i and set it to 0 if g_i is $\wedge$ gate and 1 if g_i is $\vee$. We also remove the gates that have become a constant and hardwire their output to get the result circuit C'. Hence, all the gates at the bottom level are not fed negated to the level above.

In this process, we have eliminated the k negations leaving us with the circuit C'' where all the gates at bottom and middle layer computes some monotone function on the remaining $m = n - (i + j)$ for some $j \leq k$ variables. Since the resulting circuit must compute parity on m variables, by [1], $size(C'') \geq 2^{\Omega(m^{1/12})}$. Since C'' is of depth 3, the number of bottom and middle gates in C' must also be at least $2^{\Omega(m^{1/12})}$. As the gates in the bottom and middle level computes monotone function, setting all the variables to 1 in C'' forces all of them must output 1 (Here we use the fact that the redundant gates are eliminated in C). Hence in C, there is a setting of input such that at least $i + k \geq i + j = n - m$ gates contributes an energy of 1 (since either the input to the negation or the negation gate itself will be 1) and $2^{\Omega(m^{1/12})}$ gates in C that evaluate to 1. Hence $\mathsf{EC}(C) \geq n - m + 2^{\Omega(m^{1/12})}$. Let c be the smallest integer such that for $m \geq c$, $2^{\Omega(m^{1/12})}$ is larger than m. Then, for $m \geq c$, $n - m + 2^{\Omega(m^{1/12})} \geq n$ and for $m < c$, $\mathsf{EC}(C) \geq n - c + 1 = \Omega(n)$. Hence $\mathsf{EC}(C) = \Omega(n)$. □

5 Energy Complexity of Boolean Formulas

For a formula F, let $\mathsf{L}(F)$ denote the number of leaves in the formula F. For any formula F, clearly $\mathsf{EC}^{\mathsf{F}}(F) \leq \mathsf{L}(F) - 1$. Unlike circuits, any subfunction computed in a formula cannot be reused which can potentially lead to many gates that output a 1 on some input. For this reason, one would expect that it is unlikely for Boolean formulas to be energy efficient. As a warm up, we first implement the above argument for structured Boolean formulas where we prove strong lower bounds of $\Omega(\mathsf{L}(F))$ (Section 5.1) and discuss its limitations. Then, using a different approach, we show a weaker lower bound of $\Omega(\sqrt{\mathsf{L}(F)} - \mathsf{Depth}(F))$ (Section 5.2) for arbitrary Boolean formulas.

5.1 A Warm up

We consider the following approach to prove a lower bound on energy complexity of a formula F by exhibiting an input on which many gates are guaranteed to output a 1. Suppose t be the number of gates in a formula which have both its inputs as variables. We call such gates as *non-skew gates*. Now, set the n variables to 0 or 1 uniformly at random. Then, each of the t gates evaluate to a 1 with probability at least $1/4$. Hence, on expectation, there are at least $t/4$ such gates evaluating to 1. This implies the existence of an input on which $\Omega(t)$ gates fire which gives the following proposition.

Proposition 5.1. *For a formula F, let t be the number of non-skew gates in F. If $t = \Omega(\mathsf{L}(F))$, then $\mathsf{EC}^{\mathsf{F}}(F) = \Theta(\mathsf{L}(F))$.*

However this argument fails[6] for formulas where the gates are *skew* (i.e. exactly one of the input to the gate is a variable) since randomly setting the input does not necessarily guarantee

[6]In the conference version of this paper [5], it was erroneously claimed that Proposition 5.1 holds for *all* Boolean formulas (that is, irrespective of t).

a constant probability for the skew part to output a 1 (for example, consider formulas whose underlying graph is the fully right skewed tree). Hence this approach does not give a lower bound for $\mathsf{EC}^{\mathsf{F}}(F)$ for an arbitrary formula F.

Nevertheless, there can be special formulas for which we can prove the lower bound of $\Omega(\mathsf{L}(F))$. For instance, consider the read-once formulas with negations at leaf. Similar to the argument of energy of monotone circuits (Proposition 2.3), the following can be concluded about them (irrespective of the formula structure).

Proposition 5.2. *For any read-once formula F with negations at the leaf,* $\mathsf{EC}^{\mathsf{F}}(F) = \mathsf{L}(F) - 1$.

5.2 Bounds on Energy Complexity for Boolean Formulas

In this section, we take a different approach and show the following lower bound on the energy complexity of any Boolean formula.

Theorem 1.5. For a Boolean function f, computed by a formula F,

$$\mathsf{EC}^{\mathsf{F}}(F) = \Omega\left(\sqrt{\mathsf{L}(F)} - \mathsf{Depth}(F)\right)$$

Though the above result applies for any Boolean formula, it does not give any non-trivial lower bound for formulas that have large depths due to presence of long path of skew gates.

We now describe our approach. The main idea is to use a structural decomposition result for Boolean formulas due to Guo and Komargodski [6] (see also Tal [17]). They showed that any formula F can be transformed to another "structured formula" F' without blowing up the size. More precisely,

Theorem 5.3 (Theorem 3.1 of [6])**.** *Let a Boolean function f be computed by a Boolean formula F with* $\mathsf{negs}(F) \geq 1$. *Then, there exists $T \leq 5\mathsf{negs}(F) - 2$ monotone functions $\{g_1 \ldots, g_T\}$ where each g_i is computed by a monotone formula G_i and a function $h : \{0,1\}^T \to \{0,1\}$ computed by a read-once formula H such that $f(x) = h(g_1, \ldots, g_T)$ computed by the formula $F' \stackrel{def}{=} H(G_1, \ldots, G_T)$ satisfy* $\mathsf{L}(F') \leq 2\mathsf{L}(F)$.

We now describe our proof strategy : firstly, we analyze the energy of the formula F' obtained in Theorem 5.3 and show (in Lemma 5.5) that $\mathsf{EC}^{\mathsf{F}}(F')$ is upper bounded asymptotically by $O(\mathsf{negs}(F) \times (\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F))$. This implies that the decomposition in Theorem 5.3 is not only size efficient but also energy efficient. The specific structure of the formula from Theorem 5.3 implies that $\mathsf{EC}^{\mathsf{F}}(F')$ is lower bounded by $\Omega(\mathsf{L}(F) - \mathsf{negs}(F))$ (Lemma 5.6). Finally, comparing the upper and lower bound for $\mathsf{EC}^{\mathsf{F}}(F')$ gives a lower bound on $\mathsf{EC}^{\mathsf{F}}(F)$ in terms of $\mathsf{L}(F), \mathsf{Depth}(F)$ and $\mathsf{negs}(F)$ using which we prove Theorem 1.5. Before proceeding, we need the following observation.

Proposition 5.4. *Let F be any formula and g be any gate of F other than the root. Let D be a formula obtained by replacing the subtree at gate g by a variable z. Then for any $b \in \{0,1\}$,* $\mathsf{EC}(D|_{z=b}) \leq \mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F)$.

Proof. Fix a $b \in \{0,1\}$ and let a be an input on which $D|_{z=b}$ achieves the maximum energy. Consider the evaluation of gates in F on this input a. If we ignore the gates in the subtree rooted at g

in F, as F is a formula, the evaluation of gates on the input a for F and $D|_{z=b}$ can differ only on those gates that lie in the path from g to the root. Hence,

$$\mathsf{EC}^{\mathsf{F}}(F) \geq \mathsf{EC}^{\mathsf{F}}(F,a) \geq \mathsf{EC}(D|_{z=b},a) - \mathsf{Depth}(F) = \mathsf{EC}(D|_{z=b}) - \mathsf{Depth}(F)$$

which completes the proof. □

Lemma 5.5 (Upper Bounding $\mathsf{EC}^{\mathsf{F}}(F')$)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be computed by a Boolean formula F with $\mathsf{negs}(F) \geq 1$. Then, the formula F' computing f obtained by applying the decomposition of Theorem 5.3 to F satisfies,*

$$\mathsf{EC}^{\mathsf{F}}(F') \leq (5\mathsf{negs}(F) - 2)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1) \tag{1}$$

Proof. We proceed by tracing the construction in Theorem 3.1 of [6] (Theorem 5.3) where we bound the energy of the resulting formula thereby proving the result.
[By strong induction on $\mathsf{negs}(F)$] For the base case with $\mathsf{negs}(F) = 1$, let F_1 be the minimal formula containing all negations of F. If $F_1 = F$, then the root gate of F must be a NOT gate and $F' = F$ satisfies Eq. (1). Otherwise, let F_2 be the formula obtained by replacing F_1 in F by a new variable z. As F_1 has the only negation gate of F, F_2 is monotone implying $F_2 = F_2|_{z=0} \vee (F_2|_{z=1} \wedge z)$. Also there exists a formula $\widetilde{F_1}$ such that $F_1 = \neg\widetilde{F_1}$. Now the formula $F' = F_2|_{z=0} \vee (F_2|_{z=1} \wedge \neg\widetilde{F_1})$ computes the same function as F. Since $\mathsf{EC}^{\mathsf{F}}(F')$ is upper bounded by the energy of the individual formulas and the connecting gates,

$$\begin{aligned}
\mathsf{EC}^{\mathsf{F}}(F') &\leq \mathsf{EC}^{\mathsf{F}}(F_2|_{z=0}) + \mathsf{EC}^{\mathsf{F}}(F_2|_{z=1}) + \mathsf{EC}(\widetilde{F_1}) + 3 \\
&\leq 2\mathsf{EC}^{\mathsf{F}}(F) + 2\mathsf{Depth}(F) + \mathsf{EC}(\widetilde{F_1}) + 3 && [\text{Proposition 5.4}] \\
&\leq 3\mathsf{EC}^{\mathsf{F}}(F) + 2\mathsf{Depth}(F) + 3 && [\widetilde{F_1} \text{ is a subformula of } F] \\
&\leq 3(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1)
\end{aligned}$$

For the inductive case, let F be any Boolean formula with $t = \mathsf{negs}(F) > 1$ and the result holds for all formulas with negations less than t. Let F_1 be the smallest subformula of F that contains all the negations of F. There can be two cases.

Case 1. F_1 **is same as** F **:** In this case, we show that there is an F' computing the same function as F with $\mathsf{EC}^{\mathsf{F}}(F') \leq (5\mathsf{negs}(F) - 4)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1)$ satisfying Eq. (1). Based on the root gate of F, there can be two subcases.

Suppose the root of F is a NOT gate. Then, there exists a formula E such that $F = \neg E$. Since $\mathsf{negs}(E) = \mathsf{negs}(F) - 1$, by induction, there exists an E' computing the same function as E with $\mathsf{EC}^{\mathsf{F}}(E') \leq (5\mathsf{negs}(E) - 2)(\mathsf{EC}^{\mathsf{F}}(E) + \mathsf{Depth}(E) + 1)$. Now the formula $F' = \neg E'$ computes the same function as F. Estimating $\mathsf{EC}^{\mathsf{F}}(F')$, we have

$$\begin{aligned}
\mathsf{EC}^{\mathsf{F}}(F') &\leq \mathsf{EC}^{\mathsf{F}}(E') + 1 \\
&\leq (5\mathsf{negs}(E) - 2)(\mathsf{EC}^{\mathsf{F}}(E) + \mathsf{Depth}(E) + 1) + 1 && [\text{Induction}] \\
&\leq (5(\mathsf{negs}(F) - 1) - 2)(\mathsf{EC}^{\mathsf{F}}(E) + \mathsf{Depth}(F)) + 1 && [\mathsf{Depth}(E) = \mathsf{Depth}(F) - 1] \\
&\leq (5\mathsf{negs}(F) - 4)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1) && [\mathsf{EC}^{\mathsf{F}}(E) \leq \mathsf{EC}^{\mathsf{F}}(F) + 1]
\end{aligned}$$

Suppose the root of F is AND/OR. Without loss of generality, let the root be OR gate. A similar argument holds for the case of AND gate. Then, let $F = E_\ell \vee E_r$. where E_ℓ, E_r are the left and right subtrees of the root, respectively. Since E_ℓ and E_r are subformulas of F, observe that $\mathsf{EC}^{\mathsf{F}}(E_\ell) \leq \mathsf{EC}^{\mathsf{F}}(F)$ and $\mathsf{EC}^{\mathsf{F}}(E_r) \leq \mathsf{EC}^{\mathsf{F}}(F)$. Since $F_1 = F$, it must be that $\mathsf{negs}(E_\ell) \geq 1$ and $\mathsf{negs}(E_r) \geq 1$. Hence, by induction, there exists formulas E'_ℓ and E'_r computing the same function as E_ℓ and E_r, respectively. Consider the formula $F' = E'_\ell \vee E'_r$. We now show that F' satisfies the required energy bound.

$$\begin{aligned}
\mathsf{EC}^{\mathsf{F}}(F') &\leq \mathsf{EC}^{\mathsf{F}}(E'_\ell) + \mathsf{EC}^{\mathsf{F}}(E'_r) + 1 \\
&\leq (5\mathsf{negs}(E_\ell) - 2)(\mathsf{EC}^{\mathsf{F}}(E_\ell) + \mathsf{Depth}(E_\ell) + 1) \\
&\quad + (5\mathsf{negs}(E_r) - 2)(\mathsf{EC}^{\mathsf{F}}(E_r) + \mathsf{Depth}(E_r) + 1) + 1 \quad \text{[Induction]} \\
&\leq (5\mathsf{negs}(E_\ell) - 2)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F)) \\
&\quad + (5\mathsf{negs}(E_r) - 2)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F)) + 1 \quad [\mathsf{Depth}(E_\ell), \mathsf{Depth}(E_r) \leq \mathsf{Depth}(F) - 1] \\
&\leq (5\mathsf{negs}(F) - 4)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1)
\end{aligned}$$

Case 2. F_1 **is not same as** F **:** Let F_2 be the formula obtained by replacing F_1 in F by a new variable z. Similar to the argument in the base case, $F' = F_2|_{z=0} \vee (F_2|_{z=1} \wedge F_1)$ computes the same function as F. Since F_1 does not have a smaller subformula containing all its negations, we can apply Case 1 to F_1 to get a formula F'_1 computing same function as F_1 with $\mathsf{EC}^{\mathsf{F}}(F'_1) \leq (5\mathsf{negs}(F_1) - 4)(\mathsf{EC}^{\mathsf{F}}(F_1) + \mathsf{Depth}(F_1) + 1)$. Hence,

$$\begin{aligned}
\mathsf{EC}^{\mathsf{F}}(F') &\leq \mathsf{EC}^{\mathsf{F}}(F_2|_{z=0}) + \mathsf{EC}^{\mathsf{F}}(F_2|_{z=1}) + \mathsf{EC}^{\mathsf{F}}(F'_1) + 2 \\
&\leq 2\mathsf{EC}^{\mathsf{F}}(F) + 2\mathsf{Depth}(F) + \mathsf{EC}^{\mathsf{F}}(F'_1) + 2 \quad \text{[Proposition 5.4]} \\
&\leq (5\mathsf{negs}(F_1) - 4)(\mathsf{EC}^{\mathsf{F}}(F_1) + \mathsf{Depth}(F_1) + 1) \\
&\quad + 2(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1) \\
&\leq (5\mathsf{negs}(F_1) - 4 + 2)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1) \quad [F_1 \text{ is a subformula of } F] \\
&= (5\mathsf{negs}(F) - 2)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1)
\end{aligned}$$

□

Lemma 5.6 (Lower Bounding $\mathsf{EC}^{\mathsf{F}}(F')$)**.** *Let F be a formula and F' be the formula obtained by applying Theorem 5.3 to F. Then,* $\mathsf{EC}^{\mathsf{F}}(F') \geq \mathsf{L}(F) - (5\mathsf{negs}(F) - 2)$.

Proof. By Theorem 5.3 the F' obtained is a composition of a read-once formula H over monotone formulas $G_1, \ldots, G_T$ for $T \leq 5\mathsf{negs}(F) - 2$. In addition, by tracing the construction of F' in the proof of Theorem 5.3, it can be inferred that (1) all leaves of F' forms a part of some monotone formula G_i and (2) every leaf in F must appear at least once as a leaf of F'. Now,

$$
\begin{aligned}
\mathsf{EC}^{\mathsf{F}}(F') &\geq \mathsf{EC}^{\mathsf{F}}(F', 1^n) \\
&\geq \sum_{i=1}^{T} \mathsf{EC}^{\mathsf{F}}(G_i, 1^n) \\
&\geq \sum_{i=1}^{T} (\mathsf{L}(G_i) - 1) && [G_i\text{s are monotone}] \\
&\geq \mathsf{L}(F) - T && [\text{By Property (1) and (2)}] \\
&\geq \mathsf{L}(F) - (5\mathsf{negs}(F) - 2)
\end{aligned}
$$

□

Theorem 1.5 holds directly from the following cumbersome but slightly stronger claim.

Claim 5.7. *For any formula F,* $\mathsf{EC}^{\mathsf{F}}(F) = \Omega\left(\sqrt{\mathsf{L}(F) + \mathsf{Depth}(F)^2 + \mathsf{Depth}(F)} - \mathsf{Depth}(F)\right)$

Proof. If $\mathsf{negs}(F) = 0$, then F is monotone and $\mathsf{EC}^{\mathsf{F}}(F) = \mathsf{EC}^{\mathsf{F}}(F, 1^n) = \mathsf{L}(F) - 1$. Otherwise, $\mathsf{negs}(F) \geq 1$ and applying Lemma 5.5 we have $\mathsf{EC}^{\mathsf{F}}(F') \leq (5\mathsf{negs}(F) - 2)(\mathsf{EC}^{\mathsf{F}}(F) + \mathsf{Depth}(F) + 1)$ and by Lemma 5.6 the formula F' obtained satisfy, $\mathsf{EC}^{\mathsf{F}}(F') \geq \mathsf{L}(F) - (5\mathsf{negs}(F) - 2)$.

Combining the two bounds on $\mathsf{EC}^{\mathsf{F}}(F')$, we have $\mathsf{EC}^{\mathsf{F}}(F) \geq \frac{\mathsf{L}(F)}{5\mathsf{negs}(F)-2} - \mathsf{Depth}(F) - 2$. Along with Proposition 2.2, we have

$$
\mathsf{EC}^{\mathsf{F}}(F) \geq \max\left\{\frac{\mathsf{L}(F)}{5\mathsf{negs}(F) - 2} - \mathsf{Depth}(F) - 2, \mathsf{negs}(F)\right\}
$$

Let α be the largest possible value such that $\frac{\mathsf{L}(F)}{5\alpha-2} - \mathsf{Depth}(F) - 2 \geq \alpha$. This gives a quadratic equation in α and it can be verified that the maximizing α is $\frac{\sqrt{(5\mathsf{Depth}(F)+12)^2+20\mathsf{L}(F)}-(5\mathsf{Depth}(F)+8)}{10}$.

If $\mathsf{negs}(F)$ is at least α, then $\mathsf{EC}^{\mathsf{F}}(F) \geq \alpha$. Otherwise, $\mathsf{EC}^{\mathsf{F}}(F)$ is lower bounded by $\frac{\mathsf{L}(F)}{5\alpha-2} - \mathsf{Depth}(F) - 2$ which, by our choice, is at least α. Hence in both cases,

$$
\mathsf{EC}^{\mathsf{F}}(F) \geq \alpha = \Omega\left(\sqrt{\mathsf{L}(F) + \mathsf{Depth}(F)^2 + \mathsf{Depth}(F)} - \mathsf{Depth}(F)\right)
$$

□

6 Discussion and Questions

Having studied $\mathsf{EC}(f)$ as a Boolean function parameter for different circuit models over the Boolean basis $\mathcal{B}$, following are some natural questions that are left unanswered.

- While we showed, as our main result, that for any f, $\mathsf{EC}(f) \leq O(\mathsf{DT}(f)^3)$ (Theorem 1.1), we do not know if for all f, $\mathsf{DT}(f) \leq \mathsf{poly}(\mathsf{EC}(f))$ or the tightness of our main result. For instance, even exhibiting a function f with $\mathsf{EC}(f) = \omega(\mathsf{DT}(f))$ would be interesting.

- While we could show that $\mathsf{EC}(\wedge_n) \geq n/3$ based on the measure positive sensitivity (Theorem 1.2), it completely fails to give any non-trivial lower bound for $\mathsf{EC}(\vee_n)$. So, a natural question is to show non-trivial lower bounds for $\mathsf{EC}(\vee_n)$.

- For unbounded fan-in circuits of depth 3, we showed an energy lower bound of $\Omega(n)$ for parity on n bits (Theorem 1.4). The question here is to extend the same to arbitrary depth unbounded fan-in circuits.

- For any Boolean formula F, we showed a lower bound for $\mathsf{EC}^{\mathsf{F}}(F)$ in terms of its size and depth (Theorem 1.5). Can we remove the dependence on depth thereby showing that for all Boolean functions f, $\mathsf{EC}^{\mathsf{F}}(f) = \Omega(\sqrt{\mathsf{L}(f)})$?

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] A. A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987. English translation in *Mathematical Notes of the Academy of Sci. of the USSR*, 41(4):333-338, 1987.

[2] Antonios Antoniadis, Neal Barcelo, Michael Nugent, Kirk Pruhs, and Michele Scquizzato. Energy-efficient circuit design. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 303–312, 2014.

[3] Neal Barcelo, Michael Nugent, Kirk Pruhs, and Michele Scquizzato. Almost all functions require exponential energy. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part II*, pages 90–101, 2015.

[4] Stephen Cook, Cynthia Dwork, and Rdiger Reischuk. Upper and lower time bounds for parallel random access machines without simultaneous writes. *SIAM J. Comput.*, 15(1):87–97, 1986.

[5] Krishnamoorthy Dinesh, Samir Otiv, and Jayalal Sarma. New bounds for energy complexity of Boolean functions. In *Computing and Combinatorics - 24th International Conference, COCOON 2018, Qing Dao, China, July 2-4, 2018, Proceedings*, pages 738–750, 2018.

[6] Siyao Guo and Ilan Komargodski. Negation-limited formulas. *Theoretical Computer Science*, 660:75–85, 2017. A preliminary version appeared in RANDOM 2015.

[7] Johan Håstad. *Computational limitations of small depth circuits*. PhD thesis, Massachusetts Institute of Technology, 1987.

[8] Stasys Jukna. *Boolean function complexity : Advances and Frontiers*. Algorithms and combinatorics. Springer, Berlin, Heidelberg, 2012.

[9] Gloria Kissin. Measuring energy consumption in VLSI circuits: a foundation. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 99–104, 1982.

[10] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 2nd edition, 2006.

[11] S. A. Lozhkin and M. S. Shupletsov. Switching activity of Boolean circuits and synthesis of Boolean circuits with asymptotically optimal complexity and linear switching activity. *Lobachevskii Journal of Mathematics*, 36(4):450–460, 2015.

[12] A. A. Markov. On the inversion complexity of a system of functions. *J. ACM*, 5(4):331–334, October 1958.

[13] O. M. Kasim-zade. On a measure of active circuits of functional elements. In *Mathematical problems in cybernetics "Nauka"*, volume No. 4 (Russian), pages 218–228, 1992.

[14] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992. A preliminary version appeared in STOC 1990.

[15] Miklos Santha and Christopher Wilson. Limiting negations in constant depth circuits. *SIAM J. Comput.*, 22(2):294–302, April 1993.

[16] Akira Suzuki, Kei Uchizawa, and Xiao Zhou. Energy and fan-in of logic circuits computing symmetric Boolean functions. *Theor. Comput. Sci.*, 505:74–80, 2013.

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

[18] Kei Uchizawa, Rodney J. Douglas, and Wolfgang Maass. On the computational power of threshold circuits with sparse activity. *Neural Computation*, 18(12):2994–3008, 2006.

[19] Kei Uchizawa, Takao Nishizeki, and Eiji Takimoto. Energy and depth of threshold circuits. *Theor. Comput. Sci.*, 411(44-46):3938–3946, 2010. A preliminary version appeared in FCT 2009.

[20] Kei Uchizawa and Eiji Takimoto. Exponential lower bounds on the size of constant-depth threshold circuits with small energy complexity. *Theor. Comput. Sci.*, 407(1-3):474–487, 2008. A preliminary version appeared in CCC 2007.

[21] Kei Uchizawa, Eiji Takimoto, and Takao Nishizeki. Size-energy tradeoffs for unate circuits computing symmetric Boolean functions. *Theor. Comput. Sci.*, 412(8-10):773–782, 2011.

[22] M. N. Vaintsvaig. On the power of networks of functional elements. In *Soviet Physics Doklady*, volume 6, page 545, 1962.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il