Computing Cliques is Intractable

Junichiro Fukuyama, Applied Research Laboratory, The Pennsylvania State University

The class $P$ is in fact a proper sub-class of $NP$. We explore topological properties of the Hamming space $2^{[n]}$ where $[n] = \{1, 2, \ldots, n\}$. With the developed theory, we show:

(i) a theorem that is closely related to Erdős and Rado’s sunflower lemma, and claims a stronger statement in most cases,
(ii) a new approach to prove the exponential monotone circuit complexity of the clique problem,
(iii) $NC \neq NP$ through the impossibility of a Boolean circuit with poly-log depth to compute cliques, based on the construction of (ii), and
(iv) $P \neq NP$ through the exponential circuit complexity of the clique problem, based on the construction of (iii).

Item (i) leads to the existence of a sunflower with a small core in certain families of sets, which is not an obvious consequence of the sunflower lemma. In (iv), we show that any Boolean circuit computing the clique function $CLIQUE_{n, \sqrt[4]{n}}$ has a size exponential in $n$. Thus, we will separate $P/poly$ from $NP$ also.

Razborov and Rudich showed strong evidence that no natural proof can prove exponential circuit complexity of a Boolean function. We confirm that the proofs for (iii) and (iv) are not natural.

Categories and Subject Descriptors: [Theory]: Computational Complexity

General Terms: Theory

Additional Key Words and Phrases: $NP$, $NC$, $P$, the $P$-$NP$ question, circuit complexity, clique, sunflower, extremal set theory

1. INTRODUCTION

In this paper, we show that the circuit complexity of the clique problem is exponential. Thus we will have a proof of $P \neq NP$.

We investigate a mathematical structure in the Hamming space that can handle the nature of the circuit complexity of various problems. Some techniques are explored in extremal set theory. Especially, we will show the existence of a sunflower with a small core in certain families of sets, which is not an obvious consequence of Erdős and Rado’s sunflower lemma.

It should be noted here that the monotone circuit complexity of computing cliques, i.e., the minimum size of a Boolean circuit with no logical negation for the clique problem, was first proven super-polynomial by Razborov in 1985 [1]. Later, Alon and Boppana [2] improved the result to demonstrate that it is actually exponential in the number of vertices in a given graph. Their proofs are based on the existence of a sunflower in a family of sets.

Our investigation creates general tools in the Hamming space. We define the $l$-extension of a family of $m$-sets as in [4], and its generator. We will show the existence of a small generator that produces a majority of $l$-sets in a subset of the Hamming space.

With the developed theory, we show:

(i) a theorem that is closely related to Erdős and Rado’s sunflower lemma, and claims a stronger statement in most cases,
(ii) a new approach to prove the exponential monotone circuit complexity of the clique problem,
(iii) $NC \neq NP$ through the impossibility of a Boolean circuit with poly-log depth to compute cliques, based on the construction of (ii), and
(iv) $P \neq NP$ through the exponential circuit complexity of the clique problem, based on the construction of (iii).
In (iv), we show that any Boolean circuit computing the clique function CLIQUE \( n, \sqrt{n} \) has a size exponential in \( n \). Thus, we will separate \( P/poly \) from \( \text{NP} \) also.

Strong evidence is presented that a natural proof cannot prove exponential circuit complexity of a Boolean function \([14]\). If a proof is natural, it uses inductive logic on the depth of a circuit in a constructive manner, and statistical significance of the primary property (largeness) to separate a target complexity class. We will confirm that the proofs for (iii) and (iv) are not natural due to its non-constructive nature; they decide a proof object by counting in the framework of the Hamming space.

The rest of the paper consists as follows: In Section 2, we develop basic techniques in the Hamming space. Section 3 shows the existence of a small generator, which is the aforementioned generalization of the sunflower lemma. Section 4 proves the exponential monotone circuit complexity of the clique problem with the theorem in Section 3. Its construction can be modified for a general circuit to compute cliques. In Section 5, we will show \( \text{NC} \neq \text{NP} \) over such a circuit with poly-log depth. We will also demonstrate the non-naturalness of the proof in the section. We further generalize the construction to prove \( P \neq \text{NP} \) in Section 6, followed by conclusions in Section 7.

2. THE HAMMING SPACE

2.1. Family of \( m \)-Sets And Its Extension

Let \( [n] = \{1, 2, \ldots, n\} \) if \( n \in \mathbb{Z}_{>0} \), otherwise \( [n] = \emptyset \). The family \( 2^{[n]} \) of all subsets of \( [n] \) is called the Hamming Space, or universal space, whose metric is the Hamming distance. We also say that a subset \( X \subseteq [n] \) is a (sub)space of size \( |X| \), treating it as \( 2^X \). The size \( n \) of the universal space is a special positive integer, and we use it throughout the paper without defining it.

For \( X \subseteq [n] \) and \( m \in [n] \), we denote by \( \binom{X}{m} \) the family of all subsets of \( X \) whose cardinality is \( m \). Such a subset is called an \( m \)-set. The letter \( U \) denotes a given family of \( m \)-sets in the universal space, i.e., \( U \subseteq \binom{[n]}{m} \), whose size is its cardinality \( |U| \). The sparsity of \( U \) is

\[
\kappa (U) = \ln \binom{n}{m} - \ln |U|,
\]

i.e., \( |U| = \binom{n}{m} e^{-\kappa (U)} \). We occasionally emphasize that it is defined in a space \( X \). For example, if \( X = \left( \lfloor n^{1/3} \rfloor \right) \subseteq [n] \) and \( U \subseteq \binom{X}{m} \) such that \( |U| = \binom{|X|}{m}/2 \), the sparsity \( \kappa (U) \) is \( \ln 2 \) in the space \( X \), but is much larger in the universal space in general.

We may also call some special sub-family of \( 2^{[n]} \) a space. For example, the family \( \binom{[n]}{2} \) of 2-sets is said to be the edge space over \( [n] \): the set of all possible edges for a graph with vertex set \( [n] \). The family \( \binom{[n]}{2} \) consists of all the possible edge sets of size \( m \).

The complement \( \overline{U} \) of \( U \) is \( \binom{[n]}{m} \setminus U \) in the universal space. We say

\[
\kappa (\overline{U}) = -\ln \left( 1 - e^{-\kappa (U)} \right)
\]

is the complement sparsity of \( U \). The \( l \)-extension of \( U \) is defined by

\[
\text{Ext} (U, l) = \left\{ t \in \binom{[n]}{l} : \exists s \in U, s \subseteq t \right\},
\]

Footnote: In this paper, a parameter before an object denotes the number of elements unless defined otherwise. Also \( A \subset B \) means that \( A \) is a subset/sub-family of \( B \) such that \( A \neq B \).
where the integer \( l \in [n] \) is its \textit{length}. That is, it is the family of \( l \)-sets each of which contains an \( m \)-set in \( U \). If \( l < m \), \( Ext(U, l) \) is empty. The letter \( V \) is used to express a sub-family of the \( l \)-extension, \( i.e. \), \( V \subseteq Ext(U, l) \).

\textbf{Example 2.1.} If \( n = 7 \), \( m = 3 \), \( l = 5 \) and \( U = \{1, 2, 3, 4, 6\} \),
\[
\begin{align*}
[n] &= \{1, 2, 3, 4, 5, 6, 7\} , \\
Ext(U, l) &= \{\{1, 2, 3, 4, 5\}, \{1, 2, 3, 4, 6\}, \{1, 2, 3, 5, 6\}, \{1, 2, 3, 5, 7\}, \\
&\quad \{1, 2, 3, 6, 7\}, \{1, 2, 4, 5, 6\}, \{1, 2, 4, 5, 7\}, \{1, 2, 4, 6, 7\}, \{1, 3, 4, 5, 6\}, \{1, 3, 4, 5, 7\}, \{1, 3, 4, 6, 7\}, \{1, 4, 5, 6, 7\}\}
\end{align*}
\]
\[
\kappa(U) = \ln \left(\frac{n}{m}\right) - \ln |U| = \ln \left(\frac{7}{3}\right) - \ln 2 = \ln \frac{35}{2} , \quad \text{and}
\]
\[
\kappa(Ext(U, l)) = \ln \left(\frac{n}{l}\right) - \ln |Ext(U, l)| = \ln \left(\frac{7}{5}\right) - \ln 11 = \ln \frac{11}{5}.
\]

\( Ext(U, l) \) is the family of \( 5 \)-sets each of which contains \( \{1, 2, 3\} \in U \) or \( \{1, 4, 6\} \in U \).

The \( l \)-extension together with the \( l \)-shadow \( \{ t \in \binom{n}{l} : \exists s \in U, t \subseteq s \} \) and \( l \)-cover \( \{ s \in \binom{n}{m} : \exists t \in U, |s \setminus t| \leq l \} \) of \( U \) forms a significant mathematical structure in the Hamming space.\(^2\) It is related to the Isometric Problem for \( m \)-Sets that has been known in extremal set theory; the problem finds the minimum size of the \( l \)-cover of \( U \) for all possible \( U \) of a given size. The extension, shadow and cover are among the key objects in the topological properties of the Hamming space. Some facts are found previously \cite{6} \cite{7}. Especially, the following claim is shown with the \( l \)-extension \cite{4}: An \( n \)-vertex graph \( G \) contains at most \( \binom{n}{l} \cdot 2 \exp\left(-\frac{(l-1)k}{2n(n-1)}\right) \) cliques of size \( l \), if the number of edges is \( n(n-1)/2 - k \) in \( G \). Thus if \( k \) edges are removed from the complete graph of \( n \) vertices and \( l \in [n] \) is an integer such that \( lk \gg n^2 \), the number of \( l \)-cliques in the remaining graph is much smaller than \( \binom{n}{l} \).

A set \( s \subseteq [n] \) is said to \textit{generate} \( t \subseteq [n] \) if \( s \subseteq t \). Every \( l \)-set in \( Ext(U, l) \) is generated by an \( m \)-set in \( U \). We also say that \( U \) \textit{generates} \( V \) if \( V \subseteq Ext(U, l) \).

\textbf{2.2. Useful Formulas}

Assume that the size \( n \) of the universal space grows to infinity. Any objects such as numbers, sets and families are actually functions of \( n \). For example, if we say \( m \in [n] \), it is a function \( m : \mathbb{Z}_{>0} \to \mathbb{Z}_{>0} \) of \( n \) such that \( m(n) \in [n] \). A \textit{constant} is a positive real number whose value is the same for all \( n \). The letter \( \gamma \) denotes a constant, and \( \epsilon \) a sufficiently small constant that may depend on \( \gamma \).

For two non-negative real numbers \( f \) and \( g \), write \( f = O(g) \) if there exists a constant \( \gamma \) such that \( f \leq \gamma g \) for every sufficiently large \( n \), and \( f = \Omega(g) \) if \( g = O(f) \). Write \( f = o(g) \) or \( f \ll g \) if \( \lim_{n \to \infty} f/g = 0 \). In addition, \( f = \Theta(g) \) means \( f = O(g) \) and \( f = \Omega(g) \). Such an order notation may express a function of the specified growth rate. For example, if we say that \( m = \sqrt{n} - o(\sqrt{n}) \), it means \( m \) equals \( \sqrt{n} - q \) for a non-negative real number \( q \ll \sqrt{n} \). An obvious floor or ceiling function is omitted.

A family \( U \) of \( m \)-sets is called a \textit{minority} if \( \kappa(U) \gg 1 \) in the considered space, and \textit{majority} if \( \kappa(U) \ll 1 \).

\(^2\) The \( l \)-shadow contains all the \( l \)-sets that are subsets of some \( m \)-sets in \( U \). So it is empty if \( l > m \). The \( l \)-cover is the family of \( m \)-sets, each of which has Hamming distance \( 2l \) or less from an element in \( U \).
We have the Taylor series
\[ \ln(1 + x) = \sum_{j \geq 1} \frac{(-x)^j}{j}, \quad x \in (-1, 1), \]
of natural logarithm. If \( \kappa(U) \geq \lambda \) for a real number \( \lambda \gg 1 \), it means
\[ \kappa(U) \leq -\ln(1 - e^{-\lambda}) = e^{-\lambda} + O(e^{-2\lambda}) = e^{-\lambda + o(1)} \ll 1, \]
by the equation, i.e., if \( U \) is a minority, its complement is a majority in \( |n| \).

The following two well-known identities on Binomial coefficients are especially useful in this paper.

\[ \binom{p}{r} \binom{p-r}{q-r} = \binom{p}{q} \binom{q}{r} \quad (1) \]
\[ \sum_j \binom{p-r}{j} \binom{r}{q-j} = \binom{p}{q} \quad (2) \]

For example, suppose \( U \subseteq \binom{|n|}{m} \) generates \( V \subseteq \text{Ext}(U, l) \). Since each \( m \)-set in \( U \) generates at most \( \binom{n-m}{l-m} \) \( l \)-sets in \( V \), we have with (1)
\[ |U| \geq |V| \frac{\binom{n}{l}}{\binom{n-m}{l-m}} = \frac{\binom{n}{l} e^{-\kappa(V)}}{\binom{n-m}{l-m}} \]
\[ \Rightarrow \quad \kappa(U) \leq \ln \left( \frac{l}{m} \right) \kappa(V). \quad (3) \]

We express summations the same way as [8]. In (2), for instance, regard \( \binom{p-r}{j} = 0 \) if \( j \not\in [p-r] \cup \{0\} \), where the unconstrained index \( j \) denotes every integer \( j \in \mathbb{Z} \).

2.3. Asymptotics on Binomial Coefficients

Define the function \( S : (0, 1) \to \mathbb{R} \) by
\[ S(x) = \sum_{j \geq 1} \frac{x^j}{j(j + 1)}. \quad (4) \]

We have the following theorem to approximate the natural logarithm of binomial coefficient \( \binom{p}{q} \), whose proof is found in Appendix A.

**THEOREM 2.2**

\[ \left| \ln \binom{p}{q} - q \left( \ln \frac{p}{q} + 1 - S \left( \frac{q}{p} \right) \right) - \frac{1}{2} \ln \frac{p}{2q(p-q)} \right| = O \left( \frac{1}{\min(q, p-q)} \right), \]

for \( p, q \in \mathbb{Z} \) such that \( 0 < q < p \). \( \Box \)

It means \( \left| \ln \binom{p}{q} - q \left( \ln \frac{p}{q} + 1 - S \left( \frac{q}{p} \right) \right) - \frac{1}{2} \ln \frac{p}{q(p-q)} \right| = O(1) \). By our definition of order notations, we may write it as
\[ \ln \binom{p}{q} = q \left( \ln \frac{p}{q} + 1 - S \left( \frac{q}{p} \right) \right) + \frac{1}{2} \ln \frac{p}{q(p-q)} \pm O(1). \]
Computing Cliques Is Intractable

Here $O(1)$ is $r$ or $-r$ for a non-negative real number $r$ such that $r = O(1)$. It expresses a possibly negative error whose absolute value is bounded by a constant.

Also, if $p$ and $q$ are positive real numbers, omitted floor/ceiling functions add an extra $\pm O(\ln(p+q))$ error. For example, it is straightforward to see

$$\ln \left( \frac{n}{\sqrt{n}} \right) = \sqrt{n} \left( \ln \frac{n}{\sqrt{n}} + 1 - S \left( \frac{\sqrt{n}}{n} \right) \right) + O(\ln n)$$

from the theorem since $0 \leq \sqrt{n} - \lfloor \sqrt{n} \rfloor < 1$.

Noting the above, we have:

**Corollary 2.3.**

$$\ln \left( \frac{p + q}{r} \right) = \ln \left( \frac{p}{rp + q} \right) \cdot \frac{q}{p + q} \pm O(\ln(p + q)). \quad (5)$$

**Proof.**

$$\ln \left( \frac{p + q}{r} \right) = \ln \left( \frac{p}{rp + q} \right) \cdot \frac{q}{p + q} \pm O(\ln(p + q))$$

$$= \ln \left( \frac{p + q}{r} \right) \pm O(\ln(p + q)).$$

We will also find the following lemmas useful.

**Lemma 2.4.**

$$\binom{n - m}{l} = \binom{n}{l} e^{-\ln n - o\left(\frac{\ln n}{n}\right)},$$

for $m, l \in [n]$ such that $l + m \ll n$. \qed

**Lemma 2.5.**

$$\ln \binom{l - m}{m - j} \binom{m}{j} \leq \ln \binom{l}{m} - j \ln \frac{j}{m^2} + j + \ln \frac{\ln n}{n} O(1),$$

for $l, m \in [n]$ such that $m^2 \leq l$ and $j \in [m]$. \qed

Their proofs are in Appendix A also.

2.4. Some Properties of the $l$-Extension

2.4.1. Marks and Double Marks. Let $U \subseteq \binom{n}{m}$ and its $l$-extension $V = Ext(U, l)$ be fixed so that $l > m$. A mark is a pair $(t, d)$ such that $t \in U$, $d \in V$ and $t \subseteq d$, and a double mark a triple $(t, t', d)$ such that $t, t' \in U$, $d \in V$ and $t \cup t' \subseteq d$. The families of marks and double marks are denoted by $\mathcal{M}$ and $\mathcal{D}$, respectively. Their sparsities are defined as

$$\kappa(\mathcal{M}) = -\ln \left( \frac{|\mathcal{M}|}{\binom{n}{l} \binom{l}{m}} \right) \quad \text{and} \quad \kappa(\mathcal{D}) = -\ln \left( \frac{|\mathcal{D}|}{\binom{n}{l} \binom{l}{m}^2} \right),$$

resp., i.e., $|\mathcal{M}| = \binom{n}{l} \binom{l}{m} e^{-\kappa(\mathcal{M})}$ and $|\mathcal{D}| = \binom{n}{l} \binom{l}{m}^2 e^{-\kappa(\mathcal{D})}$.

**Note:** Both $\mathcal{M}$ and $\mathcal{D}$ are families of tuples of fixed cardinality subsets of $[n]$. Assume that the maximum size $N$ of such a family $\mathcal{F}$ is clearly defined; for example,
the maximum number of double marks \((t, t', d)\) is \(\binom{n}{l} \binom{l}{m}^2\). The sparsity of \(F\) is defined by \(\kappa(F) = \ln N - \ln |F|\), which is consistent with \(\kappa(U) = \ln \binom{n}{m} - \ln |U|\). This leads to the above definition of \(\kappa(M)\) and \(\kappa(D)\). The complement sparsity of \(F\) is \(\kappa(F) = \ln N - \ln (N - |F|)\).

\[ \kappa(M) = \kappa(U). \]

An \(l\)-set \(d \in V\) is incident to at most \(\binom{n}{l} \binom{l}{m} e^{-\kappa(M)}\) marks. There are \(\binom{n}{l} e^{-\kappa(M)} = \binom{n}{l} e^{-\kappa(U)}\) or more \(l\)-sets in \(V\). Thus \(\kappa(V) \leq \kappa(U)\), i.e., the \(l\)-extension \(V\) is at least as dense as \(U\). We can improve the bound with \(\kappa(D)\). Let us observe the following lemma.

**Lemma 2.6.** \(\kappa(U) \leq \kappa(D) \leq 2\kappa(U) - \kappa(Ext(U, l))\).

**Proof.** Let \(V = Ext(U, l) = \{d_1, d_2, \ldots, d_k\}\), and \(x_i\) for \(i \in [k]\) be the number of marks incident to \(d_i\). Regarding \(x_i\) as variables, solve the following optimization problem with a Lagrange multiplier: minimize \(\sum_{i=1}^{k} x_i^2\) subject to \(\sum_{i=1}^{k} x_i = |M|\) (Fig. 1).

Conclude \(\sum_{i=1}^{k} x_i^2 \geq \left(\sum_{i=1}^{k} x_i\right)^2 / k\). Thus,

\[ |V| \cdot |D| \geq |M|^2, \]

\[ \Rightarrow |V| \binom{n}{l} \binom{l}{m}^2 e^{-\kappa(D)} \geq \left( \binom{n}{l} \binom{l}{m} e^{-\kappa(U)} \right)^2, \]

\footnote{Given a family of tuples \(s = (s_1, s_2, \ldots, s_q)\), let \(s'\) be any \(s_i\) or tuple of some \(s_i\) (a projection of \(s\)), and \(s''\) be the tuple of the remaining components. For these \(s, s'\) and \(s''\), we say that \(s'\) is incident to \(s\) and \(s''\).}
\[ |V| \geq \left( \frac{n}{l} \right) e^{-2\kappa(U) + \kappa_D}, \]
\[ \Rightarrow \quad \kappa(V) \leq 2\kappa(U) - \kappa(D). \]

The upper bound \( \kappa(D) \leq 2\kappa(U) - \kappa(V) = 2\kappa(U) - \kappa(Ext(U, l)) \) follows.

To show the lower bound \( \kappa(D) \geq \kappa(U) \), we maximize \( \sum_{\ell=1}^{b} x_{\ell}^2 \) subject to \( \sum_{\ell=1}^{b} x_{\ell} = |M| \) to conclude \( |D| \leq \binom{n}{l} \left( \frac{l}{m} \right)^2 e^{-\kappa(U)}. \]

Put
\[ \kappa_D \overset{\text{def}}{=} \kappa(D) - \kappa(U), \]
to call it the proper sparsity of \( D \). It satisfies
\[
|D| = \left( \frac{n}{l} \right) \left( \frac{l}{m} \right)^2 e^{-\kappa(U) - \kappa_D}. \tag{6}
\]

By the lemma we have
\[ 0 \leq \kappa_D \leq \kappa(U) - \kappa(Ext(U, l)). \tag{7} \]

The implied inequality \( \kappa(V) = \kappa(Ext(U, l)) \leq \kappa(U) - \kappa_D \) is the improvement over the aforementioned simple bound \( \kappa(V) \leq \kappa(U) \).

2.4.2. Sub-Family of \( U \) in a Sphere. The sphere \( S_{t,j} \) of radius \( j \in [m] \) about an \( m \)-set \( t \) is the family of \( t' \in \binom{[n]}{m} \) such that \( |t' \setminus t| = j \). The sub-family of \( U \) in the sphere \( S_{t,j} \) is \( S(t, j) = U \cap S_{t,j} \).

Its sparsity \( \kappa(S(t, j)) \) satisfies \( |S(t, j)| = \binom{n-m}{l-m} \left( \frac{m}{l} \right) e^{-\kappa(S(t, j))} \).

The relationship between \( \kappa(S(t, j)) \) and \( \kappa(D) \) has an interesting property. For each \( t \in U \) and \( t' \in S(t, m-j) \), their union \( t \cup t' \) has size \( 2m - j \). They create exactly \( \binom{n-2m+j}{l-2m+j} \) double marks. With \( (1) \), we have
\[
|D| = \sum_{t \in U} |S(t, m-j)| \left( \frac{n-2m+j}{l-2m+j} \right)
= \sum_{t \in U} \binom{n-m}{l-m-j} \left( \frac{m}{j} \right) e^{-\kappa(S(t, m-j))} \left( \frac{n-m-(m-j)}{l-m-(m-j)} \right)
= \binom{n-m}{l-m} \sum_{t \in U} \binom{l-m}{m-j} \left( \frac{m}{j} \right) e^{-\kappa(S(t, m-j))}.
\]

Notice that \( \sum_{j} \binom{l-m}{m-j} \binom{n}{j} = \frac{1}{m} \) by \( (2) \). Let \( \kappa_S \) be the average of \( \kappa(S(t, j)) \) over all \( t \in U \) and \( j \in [m] \) with respect to this summation, i.e., \( \sum_{t \in U,j} \binom{l-m}{m-j} \binom{n}{n-m} e^{-\kappa_D(t, m-j)} = \binom{n}{m} e^{-\kappa(U)} \binom{1}{m} e^{-\kappa_S}. \) Then
\[
|D| = \binom{n-m}{l-m} \binom{n}{l} e^{-\kappa(U)} \left( \frac{l}{m} \right) e^{-\kappa_S} = \binom{n}{l} \left( \frac{l}{m} \right)^2 e^{-\kappa(U) - \kappa_S}.
\]

Therefore, \( \kappa_D = \kappa_S \) by \( (6) \), i.e., the proper sparsity of the double mark family \( D \) equals the average sparsity of the sub-families of \( U \) in the spheres.

With \( (7) \), we have \( 0 \leq \kappa_S \leq \kappa(U) - \kappa(Ext(U, l)) \). The average sparsity \( \kappa_S \) of the sub-families in the spheres is upper-bounded by \( \kappa(U) \). As it approaches \( \kappa(U) \), the sparsity
of the $l$-extension gets closer to zero. In other words, we have a denser extension with sparser sub-families in the spheres. This is the key observation to prove the main claim of Section 3, the extension generator theorem.

2.4.3. Space-Augmenting Extension. Later, we will encounter a situation where we want to expand the considered space containing a target $l$-extension. The following technique will be useful: Let $U \subseteq \binom{[n]}{m}$ and $X$ be a set disjoint with $[n]$. The space-augmenting extension $V$ of $U$ from the space $[n]$ into $[n] \cup X$ is the $(m + |X|)$-extension of $U$ in $[n] \cup X$, i.e., $V = \{ b \in \binom{[n] \cup X}{m+|X|} : \exists s \in U, s \subseteq b \}$. See the following lemma.

**Lemma 2.7.** (Space-Augmenting Extension) Let $U \subseteq \binom{[n]}{m}$, and $X$ be a set disjoint with $[n]$. The sparsity of the space-augmenting extension $V$ of $U$ from the space $[n]$ into $[n] \cup X$ is at least $\kappa(U)$.

**Proof.** Any element in $V$ is constructed by concatenating $t \in Ext(U, m + j)$ in the space $[n]$ with an $(|X| - j)$-set in $X$, for some $j \in [|X|] \cup \{0\}$. Thus $|U| \geq \sum_{j=0}^{n} \binom{n}{m+j} e^{-\kappa(U)} \left( \frac{|X|}{|X|-j} \right) = \left( \frac{n+|X|}{m+|X|} \right)^e e^{-\kappa(U)}$ by (2). The lemma follows. $\square$

2.5. Other Tuple Enumeration Techniques

Tuple enumeration can be a powerful tool to find topological properties of the Hamming space. In this section, we introduce other techniques related to our proof of $P \neq NP$.

2.5.1. Counting Tuples of Fixed-Cardinality Disjoint Sets. Let $q \in \mathbb{N}$ be given. Suppose we have a family $\mathcal{F}$ of $q$-tuples $(s_1, s_2, \ldots, s_q)$ such that $s_1, s_2, \ldots, s_q$ are pairwise disjoint subsets of $[n]$, and $|s_j|$ are fixed for each $j \in [q]$. The largest possible cardinality of $\mathcal{F}$ is

$$N = \binom{n}{|s_1|} (n - |s_1| - |s_2|) \cdots (n - |s_1| - |s_2| - \cdots - |s_{q-1}|).$$

(8)

This finds $N$ in the order $s_1, s_2, \ldots, s_q$. By symmetry, we have the same number with any order of $s_j$. Or we can put any of $s_j$ together and count the tuples of sub-components within the group, to have $N$ also.

For example, suppose $q = 3$ for which $N = \binom{n}{|s_1|} \binom{n-|s_1|}{|s_2|} \binom{n-|s_1|-|s_2|}{|s_3|}$. If $s_1$ and $s_2$ are put together in the enumeration of $(s_1, s_2, s_3)$, then $N = \binom{n}{|s_1|+|s_2|} \binom{n-|s_1|-|s_2|}{|s_3|}$; count $s_1 \cup s_2$ first, then $s_1$ contained in each. These two $N$ are equal by (1). See that

$$\binom{n}{|s_1|+|s_2|} \binom{n-|s_1|-|s_2|}{|s_3|} = \binom{n}{|s_1|+|s_2|} \binom{n-|s_1|-|s_2|}{|s_3|}.$$  

As a result,

$$N = \binom{n}{|s_1|} \binom{n-|s_1|}{|s_2|} \binom{n-|s_1|-|s_2|}{|s_3|}$$

in which $s_1$ is switched with $s_2$. Using (1) repeatedly, we can change the order of $s_1, s_2, s_3$ in $N$ arbitrarily.

As noted in Section 2.4.1, we define the sparsity and complement sparsity of $\mathcal{F}$ as $\ln N - \ln |\mathcal{F}|$ and $\ln N - \ln (N - |\mathcal{F}|)$, respectively. We say that $\mathcal{F}$ is a majority if its complement sparsity $\kappa(\mathcal{F})$ is asymptotically greater than 1.

Let $U_j$ be a given family of $s_j$. If each $U_j \subseteq \binom{[n]}{|s_j|}$ is a majority with sufficiently large complement sparsity and there is no particular relation between $s_1, s_2, \ldots, s_q$, then the family $\mathcal{F}$ is also a majority. We confirm it by the following lemma.

**Lemma 2.8.** Let

(i) $q \in \mathbb{N}$,

(ii) $\lambda \geq 2 \ln n$ be a real number,

(iii) $m_j \in \mathbb{N}$ for $j \in [q]$ be integers such that $\sum_{j \in [q]} m_j \leq n$, and
(iv) \( U_j \subseteq \binom{[n]}{m_j} \) for \( j \in [q] \) be families of \( m_j \)-sets such that \( \kappa(U_j) \geq \lambda \).

Then the family \( \mathcal{F} \) of \( q \)-tuples \((s_1, s_2, \ldots, s_q)\) such that \( s_j \in U_j \) are pairwise disjoint forms a majority with complement sparsity \( \Omega(\lambda) \).

**Proof.** First, enumerate all the \( q \)-tuples \((s_1, s_2, \ldots, s_q)\) such that \( |s_j| = m_j \) and \( s_j \) are pairwise disjoint, not requiring \( s \in U_j \). Their number is the above \( N \), the maximum possible cardinality of \( \mathcal{F} \).

Second, count \((s_1, s_2, \ldots, s_q)\) such that \( s_j \notin U_j \) for each \( j \in [q] \). Use an order in which the \( j^{th} \) component \( s_j \) is counted first. The maximum number \( N \) is expressed as \( \binom{n}{m_j} \) where \( m_j = |s_j| \), and \( N_j \) is the maximum number of tuples of the remaining components in the space \([n] \setminus s_j\). The number of \((s_1, s_2, \ldots, s_q)\) such that \( s_j \notin U_j \) is exactly \( (\binom{n}{m_j}) \cdot e^{-n/(\ln q)} \cdot N_j = Ne^{-\Omega(\ln q)} \).

Due to Condition (iii),

\[
\text{(#tuples } (s_1, s_2, \ldots, s_q) \text{ such that } s_j \notin U_j \text{)} \leq Ne^{-\lambda}
\]

Adding the above for all \( j \), we find that the complement of \( \mathcal{F} \) has cardinality no more than

\[
q \cdot Ne^{-\lambda} = Ne^{-\lambda + \ln q} = Ne^{-\Omega(\lambda)}
\]
as \( \lambda \geq 2 \ln n \). Hence the complement sparsity of \( \mathcal{F} \) is \( \Omega(\lambda) \). \( \square \)

A particular case of the above lemma occurs when \( U_1 = U_2 = \cdots = U_q \). Suppose every \( U_j \) is a given family \( U \) of \( m \)-sets such that \( \kappa(U) = \Omega(n^\epsilon) \) for a constant \( \epsilon > 0 \). Then the complement sparsity of \( \mathcal{F} \) is \( \Omega(n^\epsilon) \) also. In other words, there are \( N \left(1 - e^{-\Omega(n^\epsilon)}\right) \) tuples \((s_1, s_2, \ldots, s_q)\) of pairwise disjoint \( s_j \in U \). Notice that this holds even if \( q \) divides \( n \) and \( m = n/q \).

2.5.2. Splitting \( m \)-Sets into Tuples. For a family \( U \subseteq \binom{[n]}{m} \) and integer \( q \in [n] \setminus [1] \), a split of \( U \) is a family \( \mathcal{F} \) of \( q \)-tuples \((s_1, s_2, \ldots, s_q)\) such that:

(i) \( \bigcup_{j \in [q]} s_j \subseteq U \) for each \((s_1, s_2, \ldots, s_q)\) \( \in \mathcal{F} \),

(ii) \( s_1, s_2, \ldots, s_q \) are pairwise disjoint, and

(iii) \( s_j \) have fixed cardinality, i.e., \( |s_j| \) have the same positive value in any two tuples in \( \mathcal{F} \).

Write \( s = \bigcup_{j \in [q]} s_j \subseteq U \). We also say that \((s_1, s_2, \ldots, s_q) \in \mathcal{F} \) is a split of \( s \in U \), and \( s \) is split into \((s_1, s_2, \ldots, s_q)\).

Let

\[
X = (X_1, X_2, \ldots, X_q)
\]

be a given fixed split of \([n] \).

A split \( \mathcal{F} \) of \( U \) is said to be space-proportional to \( X \) if

\[
|s_j| = \left\lfloor \frac{|X_j|}{|X|} \right\rfloor \quad \text{for each } j \in [q],
\]

and \( \mathcal{F} \) includes all the \( q \)-tuples \((s_1, s_2, \ldots, s_q)\) such that (i)-(iii) with the above values of \(|s_j|\). The maximum possible cardinality of such \( \mathcal{F} \) is

\[
N_X \overset{def}{=} \prod_{j=1}^{q} \left( \frac{|X_j|}{|X|} \right).
\]

\footnote{Partition of a set is a well-known object in combinatorics. Splits differs from partitions in that they are tuples of fix-cardinality sets. In this theory, it is important to be aware of cardinality and order of the considered sets.}
The sparsity and complement sparsity of $\mathcal{F}$ are defined with it, i.e., $\kappa(\mathcal{F}) = \ln N_X - \ln |\mathcal{F}|$ and $\kappa(\mathcal{F}) = \ln N_X - \ln (N_X - |\mathcal{F}|)$.

Notes:

— The maximum number of general splits is different from $N_X$. For example, the number of general splits $(s_1, s_2)$ of $s \in U$ is $\binom{n}{|s_1| + |s_2|} (|s_1| + |s_2|)^{e^{-\kappa(U)}}$. Its maximum number $N = \binom{n}{|s_1| + |s_2|} (|s_1| + |s_2|)^{e^{-\kappa(U)}}$ is as defined in the previous subsection, and different from $N_X$.

— It changes $N_X$ whether we take the ceiling or floor function of $|X_j|m/n$. Thus there can be more than one splits of $U$ space-proportional to $X$.

Here we show that a space-proportional split $\mathcal{F}$ of $U$ forms a majority in a case when $\kappa(\bar{U}) \gg 1$ is sufficiently larger than $q$. The intuition behind the claim is (5) in Section 2.3. Assume $q = 2$, $\kappa(\bar{U}) \geq n^\varepsilon$ for a constant $\varepsilon > 0$, and a given split $(X_1, X_2)$ of $[n]$. There are at most $\binom{n}{m} e^{-n^\varepsilon}$-sets that do not belong to $U$. There are

$$\binom{n}{m} e^{-n^\varepsilon} = \left(\frac{|X_1|}{|X_1|m}\right) \left(\frac{|X_2|}{|X_2|m}\right) e^{-n^\varepsilon + O(\ln n)} = N_X e^{-n^\varepsilon + O(\ln n)}$$

or less $m$-sets $s \not\in U$ such that $|s \cap X_1| = |X_1|m/n$, due to (5). Each of the remaining $s \in U$ such that $|s \cap X_1| = |X_1|m/n$ is split into $(s_1, s_2)$ space-proportionally to $X$. Their complement sparsity is at least $n^\varepsilon - O(\ln n) = \Omega(n^\varepsilon)$, leading to $\mathcal{F}$ that forms a majority.

As long as $q$ is not too large, we can repeat the above process $q$ times to split $U$ into a majority of $(s_1, s_2, \ldots, s_q)$. We have the following lemma. It will play an important role in Section 5. Find its formal proof in Appendix B.

**Lemma 2.9.** Let

(i) $U \subset \binom{[n]}{m}$,

(ii) $q \in [n] \setminus [1]$ such that $q \ln n \ll \kappa(\bar{U})$, and

(iii) a split $X = (X_1, X_2, \ldots, X_q)$ of $[n]$ be given. There exists a split of $U$ space-proportional to $X$ whose complement sparsity is $\Omega(\kappa(\bar{U}))$. □

3. THE EXTENSION GENERATOR THEOREM

Let

$$U_g \overset{\text{def}}{=} \{ s \in U : s \supseteq g \},$$

for $U \subset \binom{[n]}{m}$ and $g \subset [n]$, i.e., $U_g$ is the family of $m$-sets in $U$ generated by the set $g$. An $(l, \lambda)$-extension generator of $U$ is a set $g \subset [n]$ such that

$$|\text{Ext}(U_g, l)| \geq \frac{(n - |g|)}{l - |g|} (1 - e^{-\lambda}).$$

Here $l \in [n] \setminus [m]$ is the extension length and $\lambda \in \mathbb{R}_{\geq 0}$ the complement sparsity of $g$.

Denote by $y$ an $(l - |g|)$-set in the space $[n] \setminus g$. We may write it as $y \in \binom{[n] \setminus g}{l - |g|}$ according to the definition given in Section 2.1. We say that $y$ is a valid set of $g$ if $g \cup y \in \text{Ext}(U_g, l)$, and error set of $g$ otherwise.

Define

$$U^*_g \overset{\text{def}}{=} \{ s \cap g : s \in U_g \}.$$
It is a family of \((m - |g|)\)-sets in the space \([n]\) \(\setminus g\). If \(g\) is an \((l, \lambda)\)-extension generator of \(U\),

\[
|\text{Ext} \left( \hat{U}_g, l - |g| \right)| \geq \frac{(n - |g|)}{l - |g|} \left( 1 - e^{-\lambda} \right).
\]

Its complement sparsity \(\kappa \left( \text{Ext} \left( \hat{U}_g, l - |g| \right) \right)\) is at least \(\lambda\) in the space \([n]\) \(\setminus g\). When \(\lambda \gg 1\), the extension generator \(g\) is a set such that \(\text{Ext} \left( \hat{U}_g, l - |g| \right)\) is a majority. It is illustrated in Fig. 2.

In this section we develop a theorem that guarantees the existence of a small extension generator \(g\) for certain \(U\), \(l\) and \(\lambda \gg 1\). It is the aforementioned structural theorem that generalizes the sunflower lemma in most cases. Our process consists of two parts to find such \(g\): Phases I and II. We show them in the following subsections.

### 3.1. Phase I

For given \(U \subseteq \binom{[n]}{m}\) such that \(m^2 \ll n\), let an integer \(l_0 \in [n]\) satisfy

\[
m^2 < l_0.
\]

In Phase I, we prove the existence of an \((l_0, \lambda_0)\)-extension generator \(g\) of size at most \(\kappa(U) \ln \frac{l_0}{m^2}\) for a real number \(\lambda_0 = \Omega(1)\)

The family \(\hat{U}_g\) in \([10]\) and its sparsity \(\kappa(\hat{U}_g)\) are defined in the space \([n]\) \(\setminus g\). Our \(g\) is a maximal subset of \([n]\) such that

\[
\kappa(\hat{U}_g) \leq \kappa(U) - r|g|, \quad \text{where} \quad r \overset{\text{def}}{=} \ln \frac{l_0}{m^2}.
\]
Such a set \( g \subset [n] \) has a size bounded by
\[
|g| \leq \frac{\kappa(U)}{r},
\]
(13)
since \( \kappa(\hat{U}_g) \) is non-negative. There exists a maximal set \( g \subset [n] \) such that (12) and (13).

In the rest of the subsection, we prove that \( g \) is an \((l_0, \lambda_0)\)-extension generator of \( U \) for some \( \lambda_0 = \Omega(1) \). The proof considers the double mark family \( \mathcal{D} \) of \( \hat{U}_g \). The arguments are not affected by the set \( g \). Assume
\[
g = \emptyset \quad \text{and} \quad \hat{U}_g = U,
\]
(14)
for simpler expressions. The proof will use
\[
\ln \left( \binom{l_0 - m}{m-j} \right) \leq \ln \left( \binom{l_0}{m} \right) - j \ln \frac{jl_0}{m^2} + j + \ln j + O(1)
\]
\[
\leq \ln \left( \binom{l_0}{m} \right) - j (\ln j + r) + j + \ln j + O(1),
\]
(15)
for \( j \in [m] \), seen with Lemma 2.5. The inequality holds true, whether it is in Case (14) or not.

Observe the following lemma.

**Lemma 3.1.** Assume (14). For each \( s \in U \) and \( j \in [m] \), let \( S(s, m - j) \) be the subfamily of \( U \) in the sphere of radius \( m - j \) about \( s \) as defined in Section 2.4.2. The sparsity of \( S(s, m - j) \) is more than \( \kappa(U) - jr - o(1) \).

**Proof.** We are given an \( m \)-set \( s \in U \subseteq \binom{|n|}{m} \) and \( j \in [m] \). Let \( s' \) be a subset of \( s \) of size \( j \), written as \( s' \in \binom{s}{j} \). Observe that
\[
|U_{s'}| < \binom{n-j}{m-j} \exp \left( -\kappa(U) + jr \right),
\]
i.e., the number of \( t \in U \) such that \( t \supseteq s' \) is upper-bounded as above. Otherwise, \( \kappa(\hat{U}_{s'}) \leq \kappa(U) - jr \) so that \( g = s' \) satisfies (12). The existence of \( s' \in \binom{s}{j} \) such that \( |U_{s'}| \geq \binom{n-j}{m-j} \exp \left( -\kappa(U) + jr \right) \) would contradict the maximality of \( g \).

The observation leads to
\[
|S(s, m - j)| < \binom{n-m}{m-j} \binom{m}{j} \exp \left( -\kappa(U) + jr + o(1) \right). \tag{16}
\]
For each \( s' \in \binom{s}{j} \), the number of \( m \)-sets \( t' \in S(s, m - j) \) such that \( t' \supseteq s' \) is no more than the number of \( t \in U \) such that \( t \supseteq s' \). With Lemma 2.4 and the given condition \( m^2 \ll n \), we have \( \binom{n-m}{m-j} = \binom{n-j}{m-j} \binom{m}{m-j} \exp \left( -O \left( \frac{(m-j)^2}{n-m} \right) \right) = \binom{n-j}{m-j} \exp \left( -O \left( \frac{(m-j)^2}{n-m} \right) \right) = \]

\footnote{If \( g \neq \emptyset \), we have \( \hat{U}_g \) in place of \( U \). This changes \( n, l_0 \) and \( m \) into \( n - |g|, l_0 - |g| \), and \( m - |g| \), respectively. The real number \( l_0/m^2 \) is replaced by \( (l_0 - |g|)/(m - |g|)^2 \) in (15), while \( r \) is independent of \( g \) defined by (12). It is straightforward to show \( (l_0 - |g|)/(m - |g|)^2 \geq l_0/m^2 = r \) for any non-negative integer \( |g| < m \). The inequality (15) is still true with the change.}
\[ \binom{n-j}{m-j} \exp(-O(m^2/n)) = \binom{n-j}{n-m} \exp(-o(1)). \] There are less than
\[ \binom{n-j}{m-j} \exp(-\kappa(U) + jr) = \binom{n-m}{m-j} \exp(-\kappa(U) + jr + o(1)) \]
elements \( t' \in S(s, m - j) \) such that \( t' \supseteq s' \). Since there are exactly \( \binom{m}{j} \) \( j \)-sets \( s' \in \binom{s}{j} \), the size of \( S(s, m - j) \) is bounded by \( 16 \).

Therefore, the sparsity of \( S(s, m - j) \) exceeds \( \kappa(U) - jr - o(1) \). \( \square \)

Each \( s \in U \) and \( t \in S(s, m - j) \) create exactly \( \binom{n-(2m-j)}{l_0-(2m-j)} \) double marks since \( |s \cup t| = 2m - j \). Due to the lemma, \( |S(s, m - j)| < (\binom{n-m}{m-j})^{(m)} e^{-\kappa(U) + jr + o(1)} \). We have the following upper bound of the size of \( \mathcal{D} \):
\[
|\mathcal{D}| = \sum_{s \in U, 0 \leq j \leq m} \binom{n-(2m-j)}{l_0-(2m-j)}
\]
\[
= \sum_{s \in U, j} \binom{n-m}{l_0-m} \binom{m}{m-j} \exp(-\kappa(U) + jr + o(1)) \binom{n-2m+j}{l_0-2m+j}
\]
\[
= e^{-\kappa(U) + o(1)} \sum_{s \in U, j} \binom{n-m}{l_0-m} \binom{l_0-m}{m-j} \binom{m}{m-j} e^{jr}
\]
\[
= e^{-\kappa(U) + o(1)} \sum_{j} \binom{l_0-m}{m-j} \binom{m}{m-j} e^{jr}
\]
\[
= \binom{n}{l_0} \binom{l_0}{m} e^{-2\kappa(U) + o(1)} \sum_{j} \binom{l_0-m}{m-j} \binom{m}{m-j} e^{jr}
\]
\[
= \binom{n}{l_0} \binom{l_0}{m} e^{-2\kappa(U) + O(1)}.
\]

Here we have used:

i) \( \binom{n-m}{l_0-m} \binom{n-2m+j}{l_0-2m+j} = \binom{n-m}{l_0-m} \binom{l_0-m}{m-j} \) by \( 1 \), and

ii)
\[
\sum_{j} \binom{l_0-m}{m-j} \binom{m}{m-j} e^{jr} < \binom{l_0}{m} + \sum_{j \geq 1} \binom{l_0-m}{m-j} \binom{m}{m-j} e^{jr}
\]
\[
\leq \binom{l_0}{m} + \sum_{j \geq 1} \binom{l_0}{m} e^{-j(\ln j + r) + j + \ln j + O(1) + jr} \quad \text{(by } \ref{eq:15})\]
\[
= \binom{l_0}{m} + \sum_{j \geq 1} \binom{l_0}{m} e^{-(j-1) \ln j + O(1)}
\]
\[
\leq \binom{l_0}{m} e^{O(1)} (1 + e^{-0 \ln 1 + 1} + e^{-1 \ln 2 + 2} + e^{-2 \ln 3 + 3} + e^{-3 \ln 4 + 4} + \ldots)
\]
\[
= \binom{l_0}{m} e^{O(1)}.
\]

We have seen \( |\mathcal{D}| < \binom{n}{l_0} \binom{l_0}{m} e^{-2\kappa(U) + O(1)} \). The proper sparsity \( \kappa_\mathcal{D} \) of the double mark family is more than \( \kappa(U) - O(1) \) by \( \ref{eq:6} \).
By \( [7] \), \( \kappa \left( \text{Ext} \left( U, l_0 \right) \right) \leq \kappa \left( U \right) - \kappa_D \). Since \( \kappa_D > \kappa \left( U \right) - O \left( 1 \right) \), it implies \( \kappa \left( \text{Ext} \left( U, l_0 \right) \right) = O \left( 1 \right) \Rightarrow \kappa \left( \text{Ext} \left( U, l_0 \right) \right) = \Omega \left( 1 \right) \). In other words, the complement sparsity of the obtained generator \( g \) is \( \Omega \left( 1 \right) = \lambda_0 \), satisfying the desired property of Phase I.

In summary, we have proven the following lemma.

**Lemma 3.2.** Let \( U \subseteq \left( \binom{n}{m} \right) \) and \( l_0 \in \left[ n \right] \) such that \( m^2 \ll n \) and \( m^2 < l_0 \). There exists an \((l_0, \Omega \left( 1 \right))\)-extension generator of \( U \) whose size is at most \( \kappa \left( U \right) / \ln \frac{\lambda_0}{m^2} \). \( \square \)

If \( l_0 \gg m^2 \) and \( \kappa \left( U \right) \) is not large, the constructed generator \( g \) is small satisfying \( (13) \). We also have:

\[
\left| \text{Ext} \left( U, l_0 \right) \right| \geq \left( n - \left| g \right| \right) \left( 1 - e^{-\lambda_0} \right) \text{ in the space } \left[ n \right], \text{ and,}
\]

\[
\left| \text{Ext} \left( \hat{U}, l_0 - \left| g \right| \right) \right| \geq \left( n - \left| g \right| \right) \left( 1 - e^{-\lambda_0} \right) \text{ in the space } \left[ n \right] \setminus g.
\]

The proper sparsity of \( D \) has played the central role to see it.

**3.2. Phase II**

We now show that the obtained \((l_0, \lambda_0)\)-generator \( g \) is indeed an \((l, \lambda)\)-generator for \( l \in \left[ n \right] \) and \( \lambda \in \mathbb{R}_{>0} \) such that

\[
1 \ll \lambda \ll \frac{n}{m^2} \quad \text{and} \quad m^2 \lambda \leq l \leq n.
\]

In this Phase II of the generator construction, we will improve the complement sparsity from \( \lambda_0 = \Omega \left( 1 \right) \) into \( \lambda \gg 1 \) by increasing the extension length from \( l_0 \) to \( l \).

We will prove that \( \text{Ext} \left( \hat{U}, i \left( l_0 - \left| g \right| \right) \right) \) for each \( i > 0 \) has complement sparsity at least \( i \lambda_0 \) in the space \( \left[ n \right] \setminus g \). Let us assume \( (11) \) such that \( l_0 = O \left( m^2 \right) \). The maximum value of the index \( i \) is \( \frac{l - \left| g \right|}{l_0 - \left| g \right|} = \Omega \left( \frac{l}{m^2} \right) \). The claim means that the complement sparsity is at least

\[
\frac{l - \left| g \right|}{l_0 - \left| g \right|} \lambda_0 = \Omega \left( \frac{l}{m^2} \right) \cdot \lambda_0 = \Omega \left( \frac{m^2 \lambda}{m^2} \right) \cdot \Omega \left( 1 \right) = \Omega \left( \lambda \right).
\]

Then our goal of finding an \((l, \lambda)\)-generator is almost achieved. The \((l_0, \lambda_0)\)-generator \( g \) we found in Phase I will be shown as a desired \((l, \lambda)\)-generator.

Let

\[
U_i = \text{Ext} \left( \hat{U}, i \left( l_0 - \left| g \right| \right) \right),
\]

in the space \( \left[ n \right] \setminus g \), and prove \( \kappa \left( U_i \right) \geq i \lambda_0 \) by induction on \( i \). The basis \( i = 1 \) is true by Phase I. Assume true for \( i \) and prove true for \( i + 1 \).

In addition, assume \( g = \emptyset \) once again; regard for simplicity that \( \left[ n \right] \) instead of \( \left[ n \right] \setminus g \) is the space to include all the considered sets. This means \( U_i = \text{Ext} \left( U, il_0 \right) \) and that the induction hypothesis \( \kappa \left( U_i \right) \geq i \lambda_0 \) is given in the universal space \( \left[ n \right] \).

To show the induction step \( \kappa \left( U_{i+1} \right) > (i + 1) \lambda_0 \), consider the pairs \((s, b)\) such that \( b \in \binom{n}{l_0} \) and \( s \in \binom{n}{l_0} \), i.e., \( b \) is any \( il_0 \)-set and \( s \) is an \( l_0 \)-set disjoint with \( b \). The pairs are the splits of all \((i + 1)l_0\)-sets. (Splits are defined in Section 2.5.2) An \( il_0 \)-set \( b \) is an element in either \( U_i \) or \( U_i^c \). Observe the following lemma.
LEMMA 3.3. Suppose \( g = \emptyset \) and fix any \( b \in U_i \). The sparsity of the family of \( s \in \binom{[n]}{l_0} \) such that \( s \cup b \in U_{i+1} \) is at most \( \kappa(U_1) \) in the space \([n] \setminus b\).

PROOF. Let 
\[
T_j = \{ t \setminus b : t \in U_1 \text{ and } |t \setminus b| = j \}.
\]
We claim there exists \( j \in [l_0] \cup \{0\} \) such that \( T_j \) has sparsity at most \( \kappa(U_1) \) in the space \([n] \setminus b\). Suppose not. Then \( \kappa(T_j) > \kappa(U_1) \) for every \( j \in [l_0] \cup \{0\} \). Any \( l_0 \)-set \( t \in U_1 \) is a \( j \)-set in \( T_j \) joined with an \((l_0 - j)\)-set in the space \( b \). By (2),
\[
|U_1| \leq \sum_j |T_j| \left( \binom{|b|}{l_0 - j} \right) < \sum_j \left( \frac{n - |b|}{j} \right) e^{-\kappa(U_1)} \left( \binom{|b|}{l_0 - j} \right) = \left( \frac{n}{l_0} \right) e^{-\kappa(U_1)} = |U_1|.
\]
A contradiction that \( |U_1| < |U_1| \). There exists \( j \in [l_0] \cup \{0\} \) such that \( \kappa(T_j) \leq \kappa(U_1) \).

Extend this \( T_j \) from length \( j \) to \( l_0 \) in the space \([n] \setminus b\), i.e., consider \( Ext(T_j, l_0) \).

Since \( \kappa(T_j) \leq \kappa(U_1) \), its \( l_0 \)-extension satisfies the same sparsity upper bound. Thus \( \kappa(Ext(T_j, l_0)) \leq \kappa(U_1) \). Since every element \( s \in Ext(T_j, l_0) \) meets the property \( s \cup b \in U_{i+1} \), the lemma follows.

If \( b \in U_1 \), every \( s \in \binom{[n] \setminus b}{l_0} \) creates a split \((s, b)\) of \( s \cup b \in U_{i+1} \). If \( b \in U_i \), the sparsity of \( s \) such that \( s \cup b \in U_{i+1} \) in the space \([n] \setminus b\) is no more than \( \kappa(U_1) \) by the lemma. The total number of splits \((s, b)\) of \( s \cup b \in U_{i+1} \) is at least \( \beta(n, |b|, |s|) \) where
\[
\beta \geq \left(1 - e^{-\kappa(U_1)}\right) + e^{-\kappa(U_1)} \left(1 - e^{-\kappa(U_1)}\right) = 1 - e^{-\kappa(U_1)},
\]
by induction hypothesis. It is equal to
\[
\beta \left( \frac{n}{|b|} \right) \left( \frac{n - |b|}{|s|} \right) = \beta \left( \frac{n}{|b|} \right) \left( \frac{n - |b|}{(|b| + |s|) - |b|} \right) = \beta \left( \frac{n}{|b| + |s|} \right) \left( \frac{|b| + |s|}{|s|} \right) = \beta \left( \frac{n}{(i + 1)l_0} \right) \left( \frac{i + 1}{l_0} \right) \left( \frac{l_0}{l_0} \right),
\]
by (1). An \((i + 1)l_0\)-set in \( U_{i+1} \) produces at most \( \binom{n + i}{l_0} \) such splits \((s, b)\). Therefore,
\[
|U_{i+1}| \geq \beta \left( \frac{n}{(i + 1)l_0} \right) \geq \left( \frac{n}{(i + 1)l_0} \right) \left( 1 - e^{-(i+1)\lambda_0} \right),
\]
meaning that \( \kappa(U_{i+1}) \geq (i + 1)\lambda_0 \). This proves the induction step.

We now have the main claim of this section:

THEOREM 3.4. (Extension Generator Theorem) Let
i) \( U \subseteq \binom{[n]}{m} \) for \( m \in [n] \),
ii) \( \lambda \in \mathbb{R} \) such that \( 1 \ll \lambda \ll \frac{n}{m^2} \),
iii) \( \epsilon \in (0, 1) \) be a sufficiently small constant, and
iv) \( l \in [n] \) such that \( l \geq m^2 \lambda / \epsilon \).

There exists an \((l, \lambda)\)-extension generator of \( U \), whose size is at most \( \kappa(U) \sqrt{\ln \frac{m^2}{m^2 \lambda / \epsilon}} \).

PROOF. Condition (iv) means \( \frac{m^2}{m^2 \lambda / \epsilon} \geq m^2 / \epsilon' \) where \( \epsilon' = \sqrt{\epsilon} \) is a sufficiently small constant. Set \( l_0 = \frac{m^2}{\epsilon'} \geq \frac{m^2}{\epsilon'} \). This satisfies (11) in addition to \( m^2 \ll n \) by (ii). The obtained set \( g \) is an \((l_0, \Omega(1))\)-extension generator of \( U \), as well as \((\epsilon', \Omega(\lambda))\)-generator by the above Phase II. See that it is also an \((l, \lambda)\)-generator by performing Phase II steps for extra \( 1 / \epsilon' \) times. \( \square \)
3.3. Application to Circuit Complexity of the Clique Problem
We will apply Theorem 3.4 to a Boolean circuit $C$ to compute the clique function $\text{CLIQUE}_{n,k}$. It satisfies

$$\text{CLIQUE}_{n,k} \iff \bigvee_{c \in \binom{n}{l}} \bigwedge_{e \in \binom{c}{2}} X_e.$$  

Here we consider a graph with the vertex set $[n] = \{1, 2, \ldots, n\}$, every $k$-clique $c \in \binom{n}{k}$ that is a $k$-set in the space $[n]$, and every edge $e \in \binom{c}{2}$ regarded as a 2-set in a clique $c$. The Boolean variable $X_e$ is true iff the edge $e$ exists in the graph. The arrow denotes the logical equivalence.

Let us assume that $C$ is a monotone circuit where $\alpha$ is its any node. We say that a $k$-clique $c$ is generated at $\alpha$ if the existence of all the edges in $\binom{c}{2}$ implies the truth of $\alpha$. Suppose that there are $\binom{n}{k} \cdot \lambda^k$-k-cliques generated at $\alpha$. This real number $\lambda$ is the sparsity of $c$, or of the set of $c$ generated at $\alpha$.

In our approach to show a size lower bound of $C$, we find an extension generator $g$ of $c$ at each $\alpha$. Related variables and their properties are:

$$k = \sqrt[3]{n} : \text{the clique size},$$
$$\epsilon \in (0, 1) : \text{a sufficiently small constant},$$
$$|C| \leq \exp(n^\epsilon) \text{ where } |C| \text{ is the number of nodes in } C, \text{ the circuit size},$$
$$q = n^{5r} : \text{an integer parameter assumed to divide } n,$$
$$g : (n/q, k)-extension generator of } c \text{ generated at a node } \alpha, \text{ and},$$
$$\lambda_c = n^{\epsilon} : \text{sparsity upper bound of } c \text{ generated at } \alpha.$$

By Theorem 3.4 there exists an $(n/q, k)$-extension generator $g$ of $c$ at every node $\alpha$ where $\binom{n}{k} \cdot \lambda_c$, or more $k$-cliques $c$ are generated. We will detail this construction in Section 2.4.3.

The valid sets $y$ of such a generator $g$ are $(l - |g|)$-sets in the space $[n] \setminus g$ where $l = n/q$. They form a majority with complement sparsity more than $k = \sqrt[3]{n}$, which is seen by the theorem. Our construction on $C$ will require a common scoped space for all the nodes in $C$ rather than $[n] \setminus g$ that depends on $g$ and $\alpha$. We extend the space $[n] \setminus g$ into $[n]$ with a space-augmenting extension defined in Section 2.4.3.

Apply Lemma 2.7 to the family of $y \in \binom{n}{l-|g|}$ in such a way that $[n] \leftarrow [n] \setminus g$ and $X \leftarrow g$. After this, $y$ have size exactly $l$ forming a majority in the space $[n]$. By the lemma, the family of $y$ satisfies the same complement sparsity lower bound $k$. The modified valid sets $y$ still satisfy the same property that:

At a considered node $\alpha$ of $C$, for every valid $y \in \binom{n}{l}$, there exists a $k$-clique $c$ generated at $\alpha$ such that $g \subset c \subset g \cup y$.

After the application of space-augmenting extension, the generator $g$ may intersect with $y$. Similar arguments lead to a more general statement:

**Corollary 3.5.** Let $U$, $m$, $l$, $\lambda$, and $\epsilon$ be as given in Theorem 3.4. There exists $\gamma \subset [n]$ of size at most $k \cdot \ln \frac{d}{m^2 \lambda}$, and a family $\mathcal{Y}_U \subset \binom{n}{l}$ with complement sparsity at least $\lambda$, such that for every $y \in \mathcal{Y}_U$, there exists $s \in U$ satisfying $g \subset s \subset g \cup y$.

The above modification is performed at one node $\alpha$. Let $\mathcal{Y}$ be the family of $l$-sets $y$ that are valid for all $g$ at any nodes in $C$. Its complement sparsity is $\Omega(k)$. For there
are no more than $\binom{n}{l} e^{-k}$ error sets $y$ of $g$ at one particular $\alpha$. The total number of error sets is bounded by
\[
\left(\binom{n}{l} e^{-k} \cdot |C| \right) = \left(\binom{n}{l} e^{-n/4 + n^*} \right) = \left(\binom{n}{l} e^{-\Omega(k)} \right),
\]
since $\epsilon \in (0, 1)$ is a sufficiently small constant. The remaining $\binom{n}{l} \left(1 - e^{-\Omega(k)} \right) l$-sets $y$ are valid for all $g$. Thus
\[
\kappa (Y) = \Omega (k),
\]
i.e., the complement sparsity of $Y$ is $\Omega (k)$.

We now consider
\[
splits y = (y_1, y_2, \ldots, y_q) \text{ of } [n] \text{ such that } y_i \in Y.
\]
By the definition a split in Section 2.5.2 the $n/q$-sets $y_1, y_2, \ldots, y_q$ in such a split $y$ are mutually disjoint so that their union is $[n]$. The fixed cardinality $|y_j| = n/q$ divides $n$ as assumed above.

Let $\mathcal{F}$ be the family of such splits $y$. Its maximum possible cardinality is $N$ given in (8). With $|y_j| = n/q$, it is expressed as
\[
N = \prod_{j=1}^{q} \left( n - (j-1)n/q \right)/n/q.
\]
The sparsity and complement sparsity of $\mathcal{F}$ are defined by $N$. By Lemma 2.8 with (20), the family $\mathcal{F}$ forms a majority with complement sparsity $\Omega (k)$.

In summary, we have constructed a family $\mathcal{F}$ of splits $y$ from the given circuit $C$ computing CLIQUE ${_{n,k}}$ such that:

**Property of $y$:** Each $y \in \mathcal{F}$ is a split $(y_1, y_2, \ldots, y_q)$ of $[n]$ such that $y_j \in Y$,

i.e.,
(i) $y_1 \cup y_2 \cup \cdots \cup y_q = [n],$
(ii) $y_j$ are pairwise disjoint, and
(iii) every $y_j \in \binom{n}{n/q}$ of $y$ is a valid set of a generator $g$ constructed at any node $\alpha$; thus, there exists a $k$-clique $c$ generated at $\alpha$ such that $g \subset c \subset g \cup y_j$.

In addition, the complement sparsity of $\mathcal{F}$ is $\Omega (k)$.

The construction of $\mathcal{F}$ is based on Theorem 3.4.

### 3.4. Extension Generator and Sunflower

Let $\Delta \in [n]$. A $\Delta$-sunflower in $U \subseteq \binom{n}{m}$ consists of $m$-sets $s_1, s_2, \ldots, s_\Delta \in U$ such that $s_j = c \cup p_j$ for some pairwise disjoint $c$, $p_1, \ldots, p_\Delta \subseteq [n]$ with $p_1 \neq \emptyset$ [9]. They are called the core $c$ and petals $p_j$ of the sunflower, respectively. The well-known sunflower lemma provides a size lower bound for $U$ that contains a $\Delta$-sunflower.

**Lemma 3.6.** (Erdős, Rado) Any $U \subseteq \binom{n}{m}$ whose size exceeds $\Delta^m m!$ contains a $\Delta$-sunflower.

---

*We assumed in the above that $n/q$ divides $n$. If it does not, we consider $y = (y_1, y_2, \ldots, y_{q-1})$ such that $|y_1| = |y_2| = \cdots = |y_{q-2}| = n/q$ and $|y_{q-1}| = n - (q-2)n/q$. Lemma 2.8 still shows that $\mathcal{F}$ forms a majority with complement sparsity $\Omega (k)$ for ununiform $|y_j|$.*/
Suppose $m = \sqrt{n}$ and $\kappa(U) \leq m\sqrt{\ln \ln n}$. In what follows, we construct with the extension generator theorem a $\Delta$-sunflower such that

$$\Delta = \frac{n}{m^2 \ln n} = \frac{n^{1/3}}{\ln^2 n},$$

and its core has a small size $|c| \ll m$.

This is not easy to show by Lemma 3.6 alone. Notice that it is possible that $|c| = m - 1$ and $|p_i| = 1$ in the sunflower lemma. If we look for a $\Delta$-sunflower $F$ in $U$ with a small core, say by applying the lemma repeatedly, we could need to guarantee that a petal is disjoint with other already constructed petals, whose sizes amount to $\Theta(m) \cdot \Theta(\Delta) = \Omega(n^{2/3}/\ln^2 n)$ in the middle of the process.

For example, suppose that we have constructed a $\Delta/2$-sunflower $F$ of petal size $|p_j| = \Delta/2$ so far. This situation is possible because the sunflowers directly produced by Lemma 3.6, denoted by $F'$, could have had small petal size $|p_j| = 1$. To have a new $F'$ meaningful for the update of $F$, some petals of $F'$ must be disjoint with those of $F$. We could need a $\Omega(m\Delta)$-sunflower $F'$ to guarantee this property. With $m\Delta = \Omega(n^{2/3}/\ln^2 n)$, it is difficult to have such $F'$ with a straightforward application of the sunflower lemma. Hence an extra observation is essential.

Find a $\Delta$-sunflower $F$ in $U$ with Theorem 3.4 as follows. Put

$$l = \frac{n}{\Delta} = n^{2/3} \ln^2 n \quad \text{and} \quad \lambda = 2 \ln n.$$

By the theorem, there exists an $(l, \lambda)$-extension generator $g$ of $U$ such that

$$|g| \leq \frac{\kappa(U)}{\ln \frac{m}{\lambda} - O(1)} \leq \frac{m\sqrt{\ln n}}{\ln \frac{2^{2/3} \ln^2 n}{m \ln n} - O(1)} = \frac{m}{\Omega(\sqrt{\ln \ln n})} \ll m.$$

Regard this $g$ as the core $c$ so its size $|c| \ll m$ is small. With Lemma 2.8 construct a split $(y_1, y_2, \ldots, y_\Delta)$ of $[n]$ similar to $y$ in Section 3.3. Here $y_j$ are i) valid $(l - |g|)$-sets of $g$ in the space $[n] \setminus g$, ii) pairwise disjoint, and iii) such that each $g \cup y_j$ contains an $m$-set $s_j \subseteq U$ since $g$ is an extension generator.

The split $y$ creates a $\Delta$-sunflower $F = \{s_1, s_2, \ldots, s_\Delta\}$ such that each $s_j$ is contained in $g \cup y_j$. Since $y_1, y_2, \ldots, y_\Delta$ are pairwise disjoint, $F$ is indeed a $\Delta$-sunflower with a small core $c = g$ and disjoint petals $p_j = s_j \setminus g \subseteq y_j$.

We have a more general statement below:

**Proposition 3.7.** (Sunflower with a Small Core) Let positive real numbers $m$ and $\eta$ satisfy $m \in [n]$, $\eta \gg 1$ and $m^2 \eta \leq \frac{\ln n}{\ln m}$. A family of $m$-sets whose sparsity is $o(m \ln \eta)$ contains an $\Omega\left(\frac{n}{m^2 \ln n}\right)$-sunflower with a core of size $o(m)$.

**Proof.** Take an $(l, 2 \ln n)$-extension generator of $U$ such that $l = m^2 \eta \ln n$, and follow the same arguments as above. \qed

The proposition suggests a relationship between extension generators and sunflowers. Theorem 3.4 with Lemma 2.8 implies the existence of a sunflower in most cases covered by Lemma 3.6. One can verify that if $|U| = (\Omega(\Delta \ln \Delta))^{m^3} m^3$, there exists an $n/\Delta$-extension generator $g$ of size less than $m$. A $q$-sunflower is immediately constructible from such $g$ with Lemma 2.8. This leads to non-trivial claims such as the above. With Theorem 3.4, we have a stronger statement that if we extend $m$-sets to $l$-sets, almost every case of disjoint $y_1, y_2, \ldots, y_\Delta$ with small $g$ includes a $\Delta$-sunflower.
4. THE MONOTONE CIRCUIT COMPLEXITY OF CLIQUE

We will prove $P \neq NP$ through the intractability of the clique problem, i.e., any algorithm computing the problem takes exponential time in the number of vertices in a given graph. It is shown by the impossibility of a polynomial-sized Boolean circuit $C$ to compute the clique function $\text{CLIQUE}_{n,k}$. Thus we will separate $P/poly$ from $\text{NP}$ also.

In this section, we show that its monotone circuit complexity, the minimum circuit size to compute $\text{CLIQUE}_{n,k}$ without logical negations, is exponential with the extension generator theorem. The construction will be modified to apply to a non-monotone circuit in the next sections.

4.1. Background: Monotone Problems and Their Circuit Complexity

A De Morgan circuit is a Boolean circuit consisting of the following four types of nodes [10]: i) conjunction (non-leaf, AND), ii) disjunction (non-leaf, OR), iii) a Boolean variable (leaf, positive literal) and iv) a negated Boolean variable (leaf, negative literal). A general Boolean circuit is converted into a De Morgan circuit of almost the same size, by pushing negations toward leaves by De Morgan’s law. We consider a Boolean circuit $C$ of this form.

Each node $\alpha$ of $C$ is associated with a Boolean function $f_\alpha$ constructed inductively: If $\alpha$ is a leaf, $f_\alpha$ is logically equivalent to the literal associated with it. If $\alpha$ is a conjunction $\alpha_1 \land \alpha_2$, then $f_\alpha = f_{\alpha_1} \land f_{\alpha_2}$. Otherwise it is a disjunction $\alpha = \alpha_1 \lor \alpha_2$, so $f_\alpha = f_{\alpha_1} \lor f_{\alpha_2}$. The Boolean function computed by $C$ is $f_r(C)$ where $r(C)$ is the root of $C$.

It is well-known that there exists a generic reduction from a deterministic Turing machine (DTM) $M$ that accepts a language $L$, to a Boolean circuit $C$ [11]. The Boolean function computed by $C$ characterizes $L$, which is denoted by $f_L$. We also say that $C$ or $M$ computes a combinatorial problem $L$ if $C$ computes $f_L$, identifying the computation with the language. The problem $L$ is monotone if $f_L$ is a monotone Boolean function, i.e., $2^V \rightarrow \{0,1\}$ ($V$: variable set for $C$) such that $x_1 \subseteq x_2 \subseteq V$ means $f_L(x_1) \Rightarrow f_L(x_2)$ where the arrow expresses logical implication.

Here are examples of $f_L$ for monotone problems $L$:

1. $k$-CLIQUE: Is there a clique of size $k$ in the given graph?
   $f_L$ is the clique function $\text{CLIQUE}_{n,k}$ to satisfy [18], i.e., $f_L \Leftrightarrow \bigvee_{c \in \binom{[n]}{k}} \bigwedge_{v \in c} X_v$.

2. $k$-THRESHOLD: Does the given set have size $k$ or more?
   $f_L \Leftrightarrow \bigvee_{c \in \binom{[n]}{k}} \bigwedge_{v \in c} X_v$ where $X_v$ is the Boolean variable that is true iff the vertex $v$ exists.

3. $k$-CONNECTED COMPONENT: Does the given graph have a connected component of size $k$?
   $f_L \Leftrightarrow \bigvee_{c \in \binom{[n]}{k}} \bigwedge_{T \subseteq c} \bigwedge_{v \in T} X_v$.

4. $k$-MATCHING: Does the given graph have a matching of size $k$?
   $f_L \Leftrightarrow \bigvee_{\{v_0, \ldots, v_{2k-1}\} \subset [n]} \bigwedge_{j=0}^{k-1} X_{\{v_2j+1, v_{2j+1}\}}$.

Notice that they all depend on $n$, the problem size of $L$, which is uniquely determined by the size of the input string to the DTM. The circuit complexity $L(L)$ of $L$ is the function that maps $n$ to the minimum value of $|C|$ over all possible $C$ to compute $L$ of the problem size $n$. Here the circuit size $|C|$ is defined in [15]. The monotone circuit complexity $L^+(L)$ of $L$ is defined similarly over all possible $C$ with no leaf associated with a negative literal. The problem $L$ is polynomially computable if $L(L)$ is upper-bounded by a polynomial in $n$. It is intractable if $L(L)$ is lower-bounded by an exponential function of $n$. 

Computing Cliques Is Intractable A:19
A super-polynomial monotone circuit complexity of CLIQUE was first proven by Razborov in 1985 [1]. Soon, he also proved that of bipartite perfect matching is $\exp (\Omega (\ln^2 n))$ [5]. This means $\mathcal{L}^+ (n/2 \text{-MATCHING})$ is a super-polynomial function of $n$. Tardos noted in [12] that there exists a problem $L \in \mathcal{P}$ such that $\mathcal{L}^+ (L)$ is exponential, i.e., the gap between monotone and general circuit complexity can be very large for some problems. Alon and Boppana [2] improved Razborov’s method to show that the monotone circuit complexity of CLIQUE is in fact exponential, i.e., $\mathcal{L}^+ (1\text{-CLIQUE}) = \exp \left( \frac{\sqrt{n}}{\log n} \right)$ when $k = (n / \log n)^{2/3}$.

The above sequence of investigations is related to the $\mathcal{P} \not\subseteq \mathcal{NP}$ question; the class $\mathcal{NP}$ of problems equals $\mathcal{P}$ if and only if CLIQUE is polynomially computable. In this paper, we will show the intractability of a monotone circuit complexity of CLIQUE in fact exponential, i.e., $\mathcal{L}^+ (1\text{-CLIQUE}) = \exp \left( \frac{\sqrt{n}}{\log n} \right)$ when $k = (n / \log n)^{2/3}$.

In addition to $\mathcal{P} \not\subseteq \mathcal{NP}$, the statuses of $\mathcal{NC} \neq \mathcal{NP}$ and $\mathcal{NC} = \mathcal{NP}$ have been unknown. We will prove $\mathcal{NC} \neq \mathcal{NP}$ in Section 5 and $\mathcal{P} \neq \mathcal{NP}$ in Section 6 by incremental modification of the methodology posed in this section.

In the rest of this section, we present a new approach to show the exponential monotone circuit complexity of CLIQUE. It will prove the following proposition.

**Proposition 4.1.** $\mathcal{L}^+ (n^{1/4}\text{-CLIQUE}) > \exp (n^\epsilon)$ for a sufficiently small constant $\epsilon > 0$. □

Equivalently, we show that no monotone circuit $C$ of size at most $\exp (n^\epsilon)$ computes $\text{CLIQUE}_n, \sqrt{n}$.

### 4.2. Related Terminology

Let $\alpha$ be any node of a Boolean circuit $C$ with root $r(C)$. A node $\alpha_1$ is a descendant (node) of $\alpha$ if there is a directed path from $\alpha$ to $\alpha_1$ in $C$ and $\alpha_1 \neq \alpha$. Since $C$ is directed and acyclic, we can align the nodes in a topological order, i.e., a node order such that $\alpha$ is numbered before any descendant $\alpha_1$. The depth of $\alpha$, denoted by $\text{depth} (\alpha)$, is the maximum length of a directed path from $r(C)$ to $\alpha$. Write $\text{depth}(C)$ for the depth of $C$, the maximum length of a directed path in $C$.

Let $\mathcal{V}$ be the set of Boolean variables for $C$. Suppose that the considered problem $L$ is a graph problem, i.e., $\mathcal{V} = \{ X_e : e \in [n] \}$ where $[n]$ is the edge space over the vertex set $[n]$. When it is obvious from the context, we identify a Boolean variable $X_e \in \mathcal{V}$ with the edge $e$. A logical assignment for $C$ is denoted by $S$. It is a set of either positive literal $X_e$ or negative literal $\neg X_e$ for every $X_e \in \mathcal{V}$, but not both. It is equivalent to a graph with the vertex set $[n]$ that is input to $C$.

Define the disjunctive normal form $\text{dnf} (\alpha)$ of $\alpha$ recursively:

i) If $\alpha$ is a leaf associated with a literal $X_e$ or $\neg X_e$, $\text{dnf} (\alpha) = \{ \{X_e\} \}$ or $\{ \{\neg X_e\} \}$, respectively.

ii) If it is a conjunction $\alpha = \alpha_1 \land \alpha_2$, then $\text{dnf} (\alpha) = \{ t_1 \cup t_2 : t_i \in \text{dnf} (\alpha_i) \}$.

iii) Otherwise, it is a disjunction $\alpha = \alpha_1 \lor \alpha_2$. Define $\text{dnf} (\alpha) = \text{dnf} (\alpha_1) \cup \text{dnf} (\alpha_2)$.
Computing Cliques Is Intractable

A term is a family of terms, i.e., a family of sets of literals. We say that a term is at \( \alpha \) if \( t \in dnf(\alpha) \). It may contain a contradiction \( \{X_e, \neg X_e\} \) for some \( X_e \in V \). A term at the root \( r(C) \) is said to be global. Clearly, the circuit \( C \) returns true to \( S \) if and only if \( S \) contains a global term of \( C \).

For a given node \( \alpha_0 \) in \( C \), construct a subgraph \( C' \) of \( C \) by the following process: Let \( \alpha \) be the current node. Started at \( \alpha = \alpha_0 \), recursively traverse the subtrees rooted at both children of \( \alpha \) if it is a conjunction. If it is a disjunction, traverse the subtree rooted at one chosen child.

By the definition of \( dnf(\alpha) \), the process constructs a term \( t_0 \) at \( \alpha_0 \) and \( t \) at every visited \( \alpha \). We say that the \( t \) generates \( t_0 \) in \( C \). This notion is illustrated in Fig. 3. The constructed subgraph of \( C \) is a derivation graph of \( t_0 \). Choose and fix a derivation graph of every global term \( t_0 \in dnf(r(C)) \).

Note that the process can visit a node \( \alpha \) more than once, creating a possibly different term at \( \alpha \) per visit. We can reconstruct \( t_0 \) into a minimal term by choosing a same term at \( \alpha \); it belongs to \( dnf(\alpha_0) \) also. By our convention, a global term \( t_0 \) means a minimal term at the root \( r(C) \) with a specified derivation graph. There uniquely exists \( t \) at every node \( \alpha \) in \( C \) that generates \( t_0 \) in \( C \), or no such \( t \) exists at \( \alpha \).

To prove Proposition 4.1, we falsely assume a monotone circuit \( C \) of size at most \( \exp(n^{5\epsilon}) \) to compute \( k\text{-CLIQUE} \) where \( l = \sqrt{n} \). A global term will be found containing no \( k \)-clique. We call it a shift, which shows the impossibility of monotone computation.

Let \( c \in \binom{[n]}{k} \) be a \( k \)-clique. A term \( t \in dnf(\alpha) \) at \( \alpha \) is said to be a dominant at \( \alpha \) if it is contained in \( \binom{c}{2} \) for some \( k \)-clique. We say that \( c \) is generated at \( \alpha \), if there exists a dominant \( t \subseteq \binom{c}{2} \) at \( \alpha \).

Use the following notation in addition to the above.

\[ q, \lambda_c, l_y, \lambda_y : \text{real numbers} \]
\[ q = n^{5\epsilon}, \lambda_c = n^\epsilon, l_y = n/q, \text{ and } \lambda_y = n^{10\epsilon}, \]
\[ g : \text{\( (l_y, \lambda_y) \)-extension generator(s) of cliques \( c \) generated at } \alpha, \]
\[ \text{CLIQUEGENERATORS} : \text{an algorithm to find } g \text{ at all } \alpha, \]
\[ C_{\alpha} : \text{family of a minority of } k\text{-cliques \( c \) excluded at } \alpha, \]

A literal is a Boolean variable or its negation. A conjunction of literals is called product term in logic. When it is viewed in combinatorics as a product of variables in a polynomial over the binary finite field, it is called monomial [9]. In this paper, we especially emphasize its set theoretical side; a literal conjunction is thought of as a set of edges (positive literals) and non-edges (negative literals) over the vertices \([n]\). In this regard, we simply call it a term identifying it with the literal set.
\( y \in \binom{[n]}{l_y} \): valid set(s) of \( g \) found by CLIQUEGENERATORS, such that

- the family of \( y \) span \([n]\) rather than \([n] \setminus g\) as in Corollary \ref{cor:clique.generators}
- \( \mathcal{Y} \): the family of \( y \) valid for all \( g \) at any \( \alpha \),

\text{SHIFT}: an algorithm to construct a shift,

\text{BLOCKEDEDGES}, \text{LOCALSHIFT}: algorithms called by \text{SHIFT},

\( \sigma = (g, g_1, g_2, \alpha) \): quadruple(s) such that \( g \) is a generator found at \( \alpha \),
and \( g_i \) at its children,

\( Q, Q_0 \): a family of \( \sigma \) constructed by Algorithm \text{SHIFT},

\( y = (y_1, y_2, \ldots, y_q) \): a split of \([n]\) such that \( y_j \in \mathcal{Y} \),

\( t(y) \): shift, a global term constructed for \( y \) containing no \( k \)-cliques,

\( t(\sigma) \): a term constructed by \text{LOCALSHIFT} for each \( \sigma \in Q_0 \), and

\text{vertex}(\cdot): the vertex set incident to the argument edge set.

4.3. Overview of the Shift Method

With the given hypothetical monotone circuit \( C \), we construct \( t(y) \) to show its impossibility.

As its first phase, we take extension generators \( g \) of \( k \)-cliques \( c \) generated at each node \( \alpha \); repeatedly find \( g \subset c \) until the number of remaining \( c \) is small enough. This is done by Algorithm CLIQUEGENERATORS presented later.

We have \( (l_y, \lambda_y) \)-generators \( g \) of \( c \) at every \( \alpha \). Since \( \lambda_y \gg \ln |C| \), there are a majority of \( l_y \)-sets \( y \) valid for all \( g \) found by the algorithm. By Lemma \ref{lem:clique.generators} there exists a \( q \)-tuple \( y = (y_1, y_2, \ldots, y_q) \) of such \( y \) satisfying the following properties:

(i) \( y_1, y_2, \ldots, y_q \) are mutually disjoint and \( y_1 \cup y_2 \cup \cdots \cup y_q = [n] \) (i.e., a split of \([n]\) defined in Section \ref{sec:clique.generators})
(ii) For every \( \alpha, g \) and \( j \in [q] \), there exists a \( k \)-clique \( c \) generated at \( \alpha \) such that \( g \subset c \subset g \cup y_j \) (Corollary \ref{cor:clique.generators}).

Here generators \( g \) have small sizes by the corollary. Finding such a split common for all \( \alpha \) and \( g \) completes the first phase of the shift construction. It is a process to normalize cliques \( c \) generated at \( \alpha \) with the same \( y \) over the circuit \( C \).

The second phase actually constructs the shift inductively on the depth of a node. To illustrate it, let us assume that the terms \( t \) at nodes \( \alpha \) with depth 2 are small, say, \( |t| = O(n^2) \). Consider \( t \) contained in \( k \)-cliques \( c \), i.e., \( t \) are dominants at \( \alpha \). By the first normalization phase, \( c \) are generated by \( g \) so that the above (i)-(ii) are satisfied. It means at each \( \alpha \), we can choose any one of \( y_1, y_2, \ldots, y_q \), independently of the choices at the other \( \alpha \).

Choose \( y_1 \) at all \( \alpha \) with depth 2. There exists a clique \( c \) contained in \( g \cup y_1 \) by (ii). It is generated at \( \alpha \) so \( \binom{c}{2} \) contains a dominant \( t \in \text{dnf}(\alpha) \). Let \( d = t \setminus \binom{c}{2} \). With the assumption that \( t \) are small, we can find an edge set \( z_1 \) of size \( n^{11/6} \) inside \( y_1 \) such that:

(a) the removal of \( z_1 \) leaves no \( n^{1/5} \)-cliques in \( y_1 \), and
(b) the number of \( \alpha \) with depth 2 such that \( d \cap z_1 \neq \emptyset \) is much smaller than the total number of \( \alpha \) with depth 2.

The existence of such \( z_1 \) will be proven by Lemma \ref{lem:clique.remove} with Lemma \ref{lem:shift.compare}.
At every $\alpha$ such that $d \cap z_1 \neq \emptyset$, change the choice from $y_1$ to $y_2$ at such $\alpha$.

Continue for $y_2, y_3, \ldots$. After the above, every $d$ is disjoint with $z = z_1 \cup z_2 \cup \cdots \cup z_q$.

We illustrate the above in Fig. 4. Perform it for other $k$-cliques generated at $\alpha$ with depth 2.

Now consider a node $\alpha$ with depth 1. Assume for simplicity that it is a conjunction $\alpha = \alpha_1 \land \alpha_2$. We have the terms $t_i$ and generators $g_i$ at $\alpha_i$ with depth 2 constructed previously. Let $t = \left( t_1 \cap \binom{z_1}{2} \right) \cup \left( t_2 \cap \binom{z_2}{2} \right)$, the union of the subsets of $t_i$ that are yet to be shifted. (The subsets $t_i \setminus \binom{z_i}{2}$ has been already shifted, i.e., they are disjoint with $z = z_1 \cup z_2 \cup \cdots$ containing no undesired cliques. This will be illustrated in Fig. 7.) Apply the process for depth 2 to these $t$ at depth 1. Use the spaces $y_{n^*+1}, y_{n^*+2}, \ldots, y_{2n^*}$, and also $d = t \cap \left( \binom{y_1}{2} \cup \binom{y_2}{2} \right)$ instead of $d = t \setminus \left( \binom{z_2}{2} \right)$. We cut $t$ into $t \cap \binom{y_2}{2}$ again.

Finally at the root $r(C)$ with depth 0, do the same as depths 1 and 2. We have constructed a term $t$ such that only $t \cap \binom{y}{2}$ possibly contains a $k$-clique. However, $g$ is small as found in the first phase, so the obtained global term contains no $k$-cliques. It is our shift $t(y)$. Conclude that such $C$ does not exist.

It is straightforward to see that the above shift method can be applied to $C$ with a poly-log depth. In summary, it dynamically creates:

— a split $y = (y_1, y_2, \ldots, y_q)$ of $[n]$ such that (i)-(ii), common for all the nodes,
— blocked edge sets \( z_1, z_2, \ldots, z_q \) such that (a)–(b), and
— a global term \( t(y) \) such that \( t(y) \cap z = \emptyset \) where \( z = z_1 \cup z_2 \cup \cdots \cup z_q \).

The algorithm \textsc{Shift} will construct \( t(y) \) by calling \textsc{BlockedEdges} and \textsc{LocalShift}. The sub-algorithm \textsc{BlockedEdges} chooses \( y_j \) and \( z_j \) for each \( \alpha \), based on which \textsc{LocalShift} finds local subsets \( t(\sigma) \) of \( t(y) \).

In the next subsection, we present its complete proof in a formalized description so that it is applicable to a non-monotone circuit. The most important information to ensure its correctness is the currently considered subsets \( d = t \cap \binom{g^1,g^2}2 \setminus \binom{y}2 \) of \( t \) to be shifted. We abstract it by defining \( \sigma = (g, g^1, g^2, \alpha) \) and \( d(\sigma) = \binom{g^1,g^2}2 \setminus \binom{y}2 \). This also removes the constraint of bounded depth for the monotone case: Treat \( \sigma \) just as nodes \( \alpha \). The number of \( \sigma \) is asymptotically smaller than \( e^q \). We will construct a shift for an arbitrary monotone circuit.

4.4. A Proof of Proposition 4.1 with Extension Generators

Fix a given circuit \( C \) of size at most \( e^{n^{1/2}} \) to compute \( k\text{-CLIQUE} \). Perform the following in a reverse topological order of nodes \( \alpha \): Let \( C_\alpha = \emptyset \) if \( \alpha \) is a leaf. Otherwise, \( C_\alpha = C_{\alpha_1} \cup C_{\alpha_2} \) where \( \alpha_i \) are the two children of \( \alpha \). Find generators \( g \) by:

\textbf{Algorithm CliqueGenerators}

Collect the \( k \)-cliques \( c \not\in C_\alpha \) generated at \( \alpha \). Regarding \( c \) as vertex sets in \([n]\), take their \((l_y,\lambda_y)\)-extension generator \( g \). Exclude all \( c \) such that \( c \not\supset g \).

Repeat until the number of remaining \( k \)-cliques \( c \) is less than \( n!e^{-\lambda_c} \). When the loop is finished, add the finally remaining \( k \)-cliques to \( C_\alpha \).

After its completion, store in \( \mathcal{Y} \) the sets \( y \in \binom{n}{l_y} \) that are valid for every \( g \) found at any \( \alpha \).

We have
\[
|g| = O \left( \frac{\lambda_c}{\ln n} \right) = O \left( \frac{n^e}{\ln n} \right) \quad \text{and} \quad \kappa (C_\alpha) = \Omega (\lambda_c) . \tag{21}
\]

When we take \( g \) of \( c \) at \( \alpha \), there are at least \( \binom{n}{l_y} e^{-\lambda_c} \)-\( k \)-cliques. The size of \( g \) is \( O \left( \frac{\lambda_c}{\ln n} \right) \) by the extension generator theorem. Since less than \( n!e^{-\lambda_c} \) \( k \)-cliques are added to \( C_\alpha \) at each \( \alpha \), the final size of \( C_\alpha \) is bounded by \( |C| \cdot \binom{n}{l_y} e^{-\lambda_c} = e^{n^{1/2}} \cdot \binom{n}{l_y} e^{-\lambda_c} = \binom{n}{l_y} e^{-\Omega(\lambda_c)}. \) Thus (21) holds true.

By Theorem 2.2, there are at most \( \left( O \left( \frac{n^{l_y}}{\ln n} \right) \right) |C| = e^{O(\lambda_c)} \) pairs of generators \( g \) and node \( \alpha \) at which \( g \) is found. Since \( g \) are \((l_y,\lambda_y)\)-generators such that \( \lambda_y \gg \lambda_c \), the number of \( y \in \binom{n}{l_y} \) valid for all \( g \) at any \( \alpha \) is \( \binom{n}{l_y} (1 - e^{-\lambda_y + O(\lambda_c)}) = \binom{n}{l_y} (1 - e^{-\Omega(\lambda_y)}) \). Thus the complement sparsity of \( \mathcal{Y} \) is \( \Omega (\lambda_y) \). In other words,

\[
\kappa (\overline{\mathcal{Y}}) = \Omega (\lambda_y) \iff \kappa (\mathcal{Y}) = - \ln \left( 1 - e^{-\Omega(\lambda_y)} \right) \iff |\mathcal{Y}| = \binom{n}{l_y} \left( 1 - e^{-\Omega(\lambda_y)} \right) . \tag{22}
\]

Find a split \( y = (y_1, y_2, \ldots, y_q) \) of \([n]\) such that \( y_j \in \mathcal{Y} \) for every \( j \in [q] \). Lemma 2.8 suggests the existence of such \( y \): Enumerate all the splits \( y \) of \([n]\) such that \( |y_j| = t_y = n/q \). Remove ones such that \( y_j \not\in \mathcal{Y} \) for any \( j \in [q] \). The remaining tuples have sparsity at least \( - \ln \left( 1 - qe^{-\lambda_y} \right) = - \ln \left( 1 - e^{-\Omega(\lambda_y)} \right) \), so there exists such \( y \). If \( q \) does not divide \( n \), disregard the last \( n \mod q \) vertices of \([n]\) = \{1,2,\ldots,n\}, i.e., consider only \( k \)-cliques in the space \([q \lfloor n/q \rfloor] \subset [n] \).
Algorithm SHIFT

Inputs:
1. Given monotone circuit $C$.
2. Family $C_\alpha$ of $k$-cliques excluded at each node $\alpha$ of $C$.
3. Split $y = (y_1, y_2, \ldots, y_q)$ of $[n]$ such that $y_j \in \mathcal{Y}$ for each $j \in [q]$.

Output: A global term $t(y)$ of $C$.

begin
1. Perform initialization;
   1-1. Construct a family $Q = Q_0$ of quadruples $\sigma = (g, g_1, g_2, \alpha)$ incident to $k$-cliques $c$ at any node $\alpha$. /* $\sigma$ is incident to $c$ at $\alpha$ if they satisfy Conditions I—IV. */
   1-2. for each $\sigma = (g, g_1, g_2, \alpha) \in Q_0$ and $j \in [q]$ do
      1-2-1. $c^* \leftarrow$ a $k$-clique that is not in $C_\alpha$, and is generated at $\alpha$ such that $g \subset c^* \subset g \cup y_j$:
      1-2-2. $f_j(\sigma) \leftarrow$ a quadruple $\sigma^* = (g, g_1', g_2', \alpha) \in Q_0$ incident to $c^*$ at $\alpha$, for some sets $g_i \subset \mathcal{V}$:
         /* $\sigma^* = f_j(\sigma)$ represents switch from a clique $c$ at $\alpha$ incident to $\sigma$, to $c^*$ incident to $\sigma^*$ such that $g \subset c^* \subset g \cup y_j$. There exists such $\sigma^*$ by Lemma 4.2 (ii). */
   1-3. end for
2. Call Algorithm BLOCKEDGES (Fig. 6) to determine $y(\sigma) \in \{y_1, y_2, \ldots, y_q\}$, $f(\sigma) \in \{f_1(\sigma), f_2(\sigma), \ldots, f_q(\sigma)\}$ and $z_j \in \binom{[n]}{\eta}/$ for all $\sigma \in Q_0$ and $j \in [q]$;
   /* As explained in the overview, BLOCKEDGES determines which of $y_1, y_2, \ldots, y_q$ is most relevant for $\sigma$ to annihilate undesired small cliques. The choice is stored in $y(\sigma)$ where $\sigma$ acts as a generalized node. */
   /* Based on $f_j$ at Step 1-2-2, it also decides $f(\sigma) \in \{f_1(\sigma), f_2(\sigma), \ldots, f_q(\sigma)\}$ and blocked edge sets $z_j \in \binom{[n]}{\eta}/$, $j \in [q]$. */
3. With $f$ determined by Step 2, call Algorithm LOCALSHIFT (Fig. 5) to construct terms $t(\sigma)$ for each $\sigma \in Q_0$;
4. return $t(\sigma)$ such that $\sigma = (\emptyset, g_1, g_2, r(C)) \in Q_0$ incident to a $k$-clique not in $C_{r(C)}$;
   /* A majority of $k$-cliques are in $C_{r(C)}$ at the root $r(C)$ due to $\mathcal{H}$. The generator size is zero by Theorem 3.4. Thus $g = \emptyset$ is the only generator found by CLIQUEGENERATORS at $r(C)$. */
end

/* As explained in the overview, BLOCKEDGES determines which of $y_1, y_2, \ldots, y_q$ is most relevant for $\sigma$ to annihilate undesired small cliques. The choice is stored in $y(\sigma)$ where $\sigma$ acts as a generalized node. */
/* Based on $f_j$ at Step 1-2-2, it also decides $f(\sigma) \in \{f_1(\sigma), f_2(\sigma), \ldots, f_q(\sigma)\}$ and blocked edge sets $z_j \in \binom{[n]}{\eta}/$, $j \in [q]$. */

Run Algorithm SHIFT described in Fig. 5 with this $y$ as an input. It returns the shift $t(y)$. In what follows, we prove that it is actually a term at the root $r(C)$ containing no $k$-clique.

Let $\sigma = (g, g_1, g_2, \alpha)$ be a quadruple such that $g \cup g_1 \cup g_2 \subset [n]$. It is said to be incident to a $k$-clique $c$ at $\alpha$ if it satisfies the following four conditions:

I. The $k$-clique $c \not\subset C_\alpha$ is generated at $\alpha$ such that $g \subset c$ is a generator found by CLIQUEGENERATORS at $\alpha$.

II. If $\alpha$ is a conjunction $\alpha_1 \land \alpha_2$, each $g_i$ is a generator found by CLIQUEGENERATORS at $\alpha_i$, such that $g_i \subset c$.

III. If $\alpha$ is a disjunction $\alpha_1 \lor \alpha_2$, then $g_1 = g_2$, which is a generator found by CLIQUEGENERATORS at either $\alpha_i$ where $c \supset g_i$ is generated.

IV. If $\alpha$ is a leaf of $C$, $g_1 = g_2 = \text{vertex}(e)$ where $e$ is the positive literal (edge) associated with $\alpha$.

Step 1-1 of SHIFT creates the family $Q_0 = \mathcal{Q}$ of $\sigma$ incident to $c$ at any node. The family $Q_0$ remains the same throughout the construction, while $\mathcal{Q}$ decreases its size dynamically. Observe basic properties of $Q_0$.
Algorithm BlockedEdges

Inputs:
1. Split $y = (y_1, y_2, \ldots, y_q)$ of $[n]$ input to SHIFT.
2. Family $Q = Q_0$ constructed by Step 1-1 of SHIFT.
3. Mapping $f_j(\sigma), j \in [q]$ constructed by Step 1-2-2 of SHIFT.

Outputs: (i) Mapping $y : Q_0 \rightarrow \{y_1, y_2, \ldots, y_q\}$, (ii) mapping $f : Q_0 \rightarrow Q_0$, and (iii) blocked dge sets $z_j \subset \binom{y_j}{2}$ for $j \in [q],$

begin

1. for $j \leftarrow 1$ to $q$
do

1-1. for each $\sigma \in Q$ do

*/ $y_j$ is chosen as $y(\sigma)$ and $f_j(\sigma)$ temporarily for the remaining $\sigma$ in the current $Q$. */

1-2. for each $\sigma \in Q$ and edge set $z_j \subset \binom{y_j}{2}$ of size $|z_j| = n^{1/6}$ do

1-2-1. if i) removal of $z_j$ leaves no $n^{1/3}$-cliques in $\binom{y_j}{2}$, and ii) $z_j \cap d(f(\sigma)) \neq \emptyset$ then create a pair $(z_j, \sigma)$;

*/ This step is to find $z_j$ satisfying the conditions (a) and (b) in the overview. */

*/ As defined by (23), $d(f(\sigma)) = \bigcup_{g=1}^2 (g \cup g^* \cup g_2)$ if $f(\sigma) = \sigma^* = (g, g^*, g_2, \alpha).$ */

1-3. end for

1-4. Find and fix $z_j$ incident to the minimum number of $(z_j, \sigma)$;

1-5. $Q(z_j) \leftarrow \{ \sigma \in Q : d(f(\sigma)) \cap z_j \neq \emptyset\};$

1-6. $Q_j \leftarrow Q \setminus Q(z_j)$ and $Q \leftarrow Q(z_j);

2. end for

end

Fig. 6. To Determine $z_j, y(\sigma)$ and $f(\sigma)$

Fig. 7. To Shift $d(\sigma) = \binom{g_1 \cup g_2}{2} \setminus \binom{g_2}{2}$ to Construct $t(\sigma)$. 
Algorithm LOCALSHIFT

Inputs:
1. \(Q_0\) constructed by Step 1-1 of SHIFT.
2. \(f(\sigma)\) determined by BLOCKED EDGES.

Output: A term \(t(\sigma)\) for every \(\sigma \in Q_0\).

\[\begin{array}{l}
\text{begin} \\
/* Do the following for each } \sigma = (g, g_1, g_2, \alpha) \in Q_0 \text{ in a reverse topological order. Inductively, we have constructed a term } t(\sigma_i) \text{ for every } \sigma_i \in Q_0 \text{ incident to a k-clique at a child of } \alpha. */ \\
1. \text{if } \alpha \text{ is a leaf of } C \text{ associated with a positive literal } x \text{ (i.e., edge } x \in \binom{[n]}{2}) \text{ then } t(\sigma) = \{x\}; \\
2. \text{else} \\
2-1. \sigma^* = (g, g_1, g_2, \alpha) \leftarrow f(\sigma); \\
2-2. \text{if } \alpha \text{ is a conjunct } \alpha_1 \land \alpha_2 \text{ then} \\
2-2-1. \text{for } i \leftarrow 1, 2 \text{ do } \sigma_i \leftarrow (g_i^*, g_1, g_2, \alpha_i) \in Q_0 \text{ for some } g_i, g_1, g_2 \subset [n]; \\
2-2-2. \text{return } t(\sigma_1) \cup t(\sigma_2); \\
2-3. \text{else } /* \alpha \text{ is a disjunction } \alpha_1 \lor \alpha_2 */ \\
2-3-1. \text{Without loss of generality, let } g_i^* = g_i^2 \text{ be a generator constructed by CLIQUE GENERATORS at } \alpha_1; \\
2-3-2. \text{This is due to Condition III since } \alpha \text{ is a disjunction. */} \\
2-3-3. \text{for } i \leftarrow 1, 2 \text{ do } \sigma_i \leftarrow (g_i, g_1, g_2, \alpha_i) \in Q_0 \text{ for some } g_1, g_2 \subset [n]; \\
2-3-4. \text{return } t(\sigma_1); \\
2-4. \text{end if} \\
\text{end}
\end{array}\]

Fig. 8. To Construct a term \(t(\sigma)\) at \(\alpha\)

Lemma 4.2. The family \(Q_0\) of \(\sigma\) satisfies the following two:

(i) \(|Q_0| < e^{O(n^3)} \ll e^n\).

(ii) For each k-clique \(c \subseteq C\), generated at \(\alpha\), and generator \(g \subseteq c\) found by CLIQUE GENERATORS at \(\alpha\), there exists \(\sigma = (g, g_1, g_2, \alpha) \in Q_0\) incident to \(c\).

Proof. (i): Since \(|g|\) and \(|\alpha|\) are bounded as \(\binom{n}{2}\) and \(|C| \leq e^{n^2}\), there are at most \(\binom{n^2}{2} e^{O(n^2)}\) such \(\sigma = (g, g_1, g_2, \alpha) \in Q_0\) by Theorem 2.2.

(ii): Fix any such \(c\) and \(g\). There exists a dominant \(t \in dnf(\alpha)\) contained in \(\binom{\alpha}{2}\) since \(c\) is generated at \(\alpha\). Suppose \(\alpha\) is a conjunction \(\alpha_1 \land \alpha_2\). The family \(dnf(\alpha_i), i \in \{1, 2\}\) contains a dominant that is a subset of \(t\), by the definition of \(dnf(\cdot)\). The clique \(c\) is generated at both \(\alpha_i\). Algorithm CLIQUE GENERATORS finds \(g_i \subset c\) at each \(\alpha_i\) since \(c \notin C \supseteq C_{\alpha_i}\). Thus there exists such a quadruple \((g, g_1, g_2, \alpha) \in Q_0\). It is shown similarly when \(\alpha\) is a disjunction or a leaf. \(\square\)

Here are what Algorithm SHIFT does:
1. It constructs the shift \(t(y)\) with generators \(g\) such that \(\sigma = (g, g_1, g_2, \alpha) \in Q_0\). Step 3 calls Algorithm LOCALSHIFT described in Fig. 8. It returns a term \(t(\sigma)\) for each \(\sigma \in Q_0\), which is the output of a recursive step for finding \(t(y)\). As illustrated in Fig. 7, \(t(\sigma)\) is constructed so that edges in \(\binom{\alpha}{2}\) are changed into those incident to some \(g_j\). These edges are shifted as said in Section 4.3.

2. For each \(\sigma = (g, g_1, g_2, \alpha)\), let

\[
d(\sigma) \overset{\text{def}}{=} \left( g_1 \cup g_2 \right) \setminus \left( g \frac{3}{2} \right)
\tag{23}
\]
We focus on shifting edges in \( d(\sigma) \subseteq \binom{y}{2} \setminus \binom{z}{2} \) at \( \alpha \). Fig. 7 explains how `LOCALSHIFT` inductively constructs \( t(\sigma) \): We have terms \( t(\sigma_i) \) at the children \( \alpha_i \) of \( \alpha \) such that \( \sigma_i = (g_i, g_i, 1, \ldots, 1, \alpha_i) \). Join \( g_i \) to gather what is yet to be shifted. We initially preserve all such cases \( (g_1, g_2, \alpha) \) as \( \sigma \) in \( Q_0 \).

3. Step 2 of `SHIFT` calls `BLOCKEDEDGES` to map every \( \sigma \in Q_0 \) to some \( y_j \) denoted by \( y(\sigma) \), and to \( \sigma^* \in Q_0 \) denoted by \( f(\sigma) \). The purpose of the mappings is to remove key edges to form an undesired clique that intersects with \( y_j \) \( y \).

4. `SHIFT` systematically switches a \( k \)-clique \( c \) to another \( c^* \) at \( \alpha \). For each given \( \sigma = (g, g_1, g_2, \alpha) \in Q_0 \), Step 1-2-1 of `SHIFT` finds \( c^* \notin C_\alpha \) such that \( g \subseteq c^* \subseteq g \cup y_j \). Loop 1 of `BLOCKEDEDGES` chooses most relevant \( y_j \) as \( y(\sigma) \) and \( f_j(\sigma) \) as \( f(\sigma) \). \( f_j \) are found by 1-2-2 of `SHIFT`.) We will see that these choices eliminate key edges of undesired cliques. The edge set \( d(\sigma) \cap \binom{y}{2} \setminus \binom{z}{2} \) is shifted to \( d(\sigma^*) \cap \binom{c^*}{2} \setminus \binom{z}{2} \).

5. Loop 1 of `BLOCKEDEDGES` determines an edge set \( z_j \subset \binom{y}{2} \) whose removal leaves no \( n^{1/5} \)-cliques in \( y_j \), such that \( z_j \cap d(f(\sigma)) = \emptyset \) for all \( \sigma \). The effect of annihilating smaller cliques from \( y_j \) remains until the end of process due to the disjointness of \( y_1, y_2, \ldots, y_q \). By 1-5 of `BLOCKEDEDGES`, any \( \sigma \in Q \) such that \( z_j \cap d(f(\sigma)) = \emptyset \) is stored in \( Q(z_j) \). It means any edge \( e \in z_j \) exists only in \( d(f(\sigma)) = \binom{y_j}{2} \setminus \binom{z}{2} \) such that \( \sigma \in Q(z_j) \). The family \( Q(z_j) \) is updated as the new \( Q \) by Step 1-6, and then Loop 1 chooses another \( y(\sigma) = y_j, j > j' \) for \( \sigma \in Q(z_j) \). As a result, the edges in \( z_j \) remain non-existent in any \( t(\sigma) \setminus \binom{z}{2} \) throughout the process. (Lemmas 4.5 and 4.6)

6. Every time Loop 1 of `BLOCKEDEDGES` increments its index \( j \), the family \( Q \) keeps reducing its size exponentially (Lemma 4.8 and Corollary 4.9). It becomes empty in the final step \( j = q \). Hence no edges in \( \bigcup_{j=1}^{q} z_j \) exists in the shift \( t(y) \), creating no \( k \)-cliques. We will show it in Lemma 4.10 based on \( t(y) \cap z = \emptyset \), where \( z \colon= \bigcup_{j=1}^{q} z_j \) (Corollary 4.7).

Put \( N = \binom{n}{2} \) that is the size of the edge space \( \binom{[n]}{2} \). Prove the following general statement first.

**Lemma 4.3.** Let \( r \in [n] \) and \( L \in [N] \) be integers such that \( r \ll n \) and \( rL \gg N \ln \frac{N}{r} \). The family \( \{ S \in \binom{[n]}{N-L} : \text{edge set } S \text{ contains no } r \text{-clique} \} \) forms a majority in \( \binom{[n]}{N-L} \) with sparsity \( -\ln \left( 1 - e^{-\Theta(Lr^2/N)} \right) \).

**Proof.** Let \( R = \binom{r}{2} \). The number of \( (N-L) \)-edge sets containing a particular \( r \)-clique is \( \binom{N-L-R}{N-L} = \binom{N-R}{L} = \binom{N}{L} e^{\Theta(-\frac{L^2}{N})} = \binom{N}{N-L} e^{\Theta(-\frac{L^2}{N})} \) by Lemma 2.4. The sparsity of \( (N-L) \)-edge sets containing any \( r \)-clique is no more than \( \Theta \left( \frac{LR}{N} \right) - \ln \left( \frac{n}{r} \right) = \Theta \left( \frac{Lr^2}{N} \right) - \Theta \left( r \ln \frac{n}{r} \right) = \Theta \left( \frac{Lr^2}{N} \right) \gg \Theta \left( r \ln \frac{n}{r} \right) \gg 1, \) due to Theorem 2.2, \( rL \gg N \ln \frac{N}{r} \) and \( \frac{n}{r} \gg 1 \). The \( (N-L) \)-edge sets containing no \( r \)-cliques form a majority whose sparsity is \( -\ln \left( 1 - e^{-\Theta(Lr^2/N)} \right) \). □
With the mapping $f$ constructed by BLOCKED-EDGES, define
\[ \Phi \overset{\text{def}}{=} \bigcup_{\sigma \in Q_0} d(f(\sigma)). \]  

(24)

Show the following lemmas.

**Lemma 4.4.** $d(f(\sigma)) \subset \binom{\emptyset \cup y(\sigma)}{2} \setminus \binom{g}{2}$ for every $\sigma \in Q_0$.

**Proof.** Fix $\sigma$ and let $\sigma^* = f(\sigma) = (g, g_1^*, g_2^*, \alpha)$. Step 1-2-1 of SHIFT and Step 1-1 of BLOCKED-EDGES together choose a $k$-clique $c^*$, such that $\sigma^*$ is incident is $c^*$ at $\alpha$ and $g \subseteq c^* \subseteq y \cup y(\sigma)$. The sets $g_i^*$ are included in $c^*$ since $\sigma^*$ is incident to $c^*$. Thus,
\[ d(\sigma^*) = (g_1^* \cup g_2^*) \setminus \binom{g}{2} \subset (g \cup y(\sigma)) \setminus \binom{g}{2}. \]

The lemma follows. \[ \Box \]

**Lemma 4.5.**
\[ \Phi \cap \hat{z} = \emptyset \]

**Proof.** Fix each $j \in [q - 1]$ and $\sigma \in Q_0$. Observe two claims.

Claim 1: If $d(f(\sigma)) \cap z_j \neq \emptyset$, then $y(\sigma) = y_j$.

Proof: $z_j$ is a subset of $\binom{y_j}{2}$. Every edge in $d(f(\sigma))$ is incident to a vertex in $y(\sigma)$ by Lemma 4.4. The subspace $y(\sigma)$ must be $y_j$ since $d(f(\sigma)) \cap \binom{y_j}{2} \neq \emptyset$, i.e., there exists an edge $e \in d(f(\sigma))$ both of whose end points are in $y_j$. \[ \Box \]

Claim 2: If $d(f(\sigma)) \cap z_j \neq \emptyset$, then $y(\sigma) \neq y_j$.

Proof: $y(\sigma) = y_j$ is decided by Step 1-1 of BLOCKED-EDGES when $\sigma$ belongs to the current $Q$. By Step 1-5, $\sigma \in Q$ is stored in $Q(z_j)$ due to $d(f(\sigma)) \cap z_j \neq \emptyset$. Step 1-6 stores it in new $Q$ so that $y(\sigma) \neq y_j$ by Step 1-1. \[ \Box \]

Therefore, $d(f(\sigma)) \cap z_j = \emptyset$. It means that
\[ \forall j \in [q - 1] \forall \sigma \in Q_0, d(f(\sigma)) \cap z_j \neq \emptyset, \]
\[ \Rightarrow \Phi \cap \hat{z} = \bigcup_{\sigma \in Q} d(f(\sigma)) \cap \hat{z} = \emptyset, \]
proving the lemma.

**Lemma 4.6.** For each $\sigma \in Q_0$, LOCALSHIFT returns a term $t(\sigma)$ at $\alpha$ such that
\[ t(\sigma) \setminus \binom{g}{2} \subseteq \Phi. \]

(25)

**Proof.** Prove it inductively in a reverse topological order of the nodes $\alpha$ of $C$. Fix a given quadruple $\sigma = (g, g_1, g_2, \alpha) \in Q_0$. The basis occurs when $\alpha$ is a leaf of $C$ associated with a positive literal $x$, i.e., an edge $x \in \binom{[n]}{2}$. Algorithm CLIQUEGENERATORS collects all $c$ generated at $\alpha$. Every such $k$-clique contains $x$ in $\binom{[n]}{2}$, so $g \supseteq \text{vertex}(x)$. LOCALSHIFT correctly returns $t(\sigma) = \{x\} \in dnf(\alpha)$ such that (25).

For induction step, we consider a conjunction $\alpha = \alpha_1 \land \alpha_2$ since a disjunctive case is shown similarly. Assume true for $\alpha_i$ and prove true for $\alpha$.

Since $g$ is a generator found at $\alpha$ by CLIQUEGENERATORS, there exists a $k$-clique $c^* \not\subseteq C_{\alpha}$ generated at $\alpha$ such that $g \subseteq c^* \subseteq y \cup y_j$ for every $j \in [q]$ by Corollary 3.5. Step 1-2-1 of SHIFT and 1-1 of BLOCKED-EDGES find this $c^*$ for $y(\sigma) \not\subseteq \emptyset$. The clique $c^*$ is generated at $\alpha$, so there exists $\sigma^* = (g, g_1^*, g_2^*, \alpha) \in Q_0$ incident to $c^*$ by Lemma 4.2(ii).
Step 2-1 of LOCALSHIFT correctly finds $\sigma^* = f(\sigma)$. The clique $c^*$ is generated at the two children $\alpha_i$ of $\alpha$. There exist $\sigma_i = (g_i, y_i, g_i, \alpha_i) \in Q_0$ incident to $c^*$ for some sets $g_i, y_i, g_i, \alpha_i \subseteq [n]$. They are chosen by Step 2-2-1 of LOCALSHIFT. By induction hypothesis, LOCALSHIFT has constructed a term $t(\sigma_i)$ at $\alpha_i$ such that (25). Its Step 2-2 returns a term at $\alpha$, i.e., $t(\sigma_i) \in dnf(\alpha) \Rightarrow t(\sigma) = t(\sigma_1) \cup t(\sigma_2) \in dnf(\alpha)$ by the definition of $dnf(\alpha)$.

To verify (25), see that

$$t(\sigma) \setminus \left(\frac{g_i}{2}\right) = \bigcup_{i=1}^{2} t(\sigma_i) \setminus \left(\frac{g_i}{2}\right) = \bigcup_{i=1}^{2} \left(\left(t(\sigma_i) \cap \left(\frac{g_i^*}{2}\right) \setminus \left(\frac{g_i}{2}\right)\right) \cup \left(t(\sigma_i) \setminus \left(\frac{g_i^*}{2}\right) \setminus \left(\frac{g_i}{2}\right)\right)\right).$$

Since $d(\sigma^*) = \left(\frac{g_i^*}{2}\right) \setminus \left(\frac{g_i}{2}\right)$,

$$\bigcup_{i=1}^{2} \left(\left(t(\sigma_i) \cap \left(\frac{g_i^*}{2}\right) \setminus \left(\frac{g_i}{2}\right)\right) \cup \left(t(\sigma_i) \setminus \left(\frac{g_i^*}{2}\right) \setminus \left(\frac{g_i}{2}\right)\right)\right) \subseteq \left(\frac{g_i^*}{2}\right) \cup \left(\frac{g_i^*}{2}\right) \setminus \left(\frac{g_i}{2}\right) \subseteq d(\sigma^*) \subseteq \Phi.$$

By induction hypothesis, $\bigcup_{i=1}^{2} t(\sigma_i) \setminus \left(\frac{g_i}{2}\right) \subseteq \Phi$. Thus $t(\sigma) \setminus \left(\frac{g_i}{2}\right) \subseteq \Phi$, proving the lemma. □

**Corollary 4.7.** $t(y) \cap \hat{\varepsilon} = \emptyset$.

**Proof.** $\hat{\varepsilon} \cap t(\sigma) \setminus \left(\frac{g_i}{2}\right) = \emptyset$ for every $\sigma \in Q_0$ from the previous two lemmas. As noted below Step 4 of SHIFT, the only generator found at the root $r(C)$ is $g = \emptyset$. The claim is true as $t(y) = t(\sigma) \setminus \left(\frac{g_i}{2}\right)$ for $\sigma$ chosen by the step such that $g = \emptyset$. □

**Lemma 4.8.** $|Q(z_j)| \ll |Q|$ after Step 1-5 of BLOCKED-EDGES for each $j \in [q]$.

**Proof.** Let $e$ be an edge in $z_j \cap d(f(\sigma))$ at Step 1-2-1 (ii) of BLOCKED-EDGES. It creates at most $\binom{[y_j]-1}{2}$ pairs $(z_j, \sigma)$; choose $|z_j| - 1$ edges of $z_j$ in the space $\left(\frac{y_j}{2}\right) \setminus \{e\}$. The number of such $e$ per $\sigma \in Q$ is no more than $|d(f(\sigma))| \leq \binom{g_i^*+g_i^*}{2} < n^{2c}$ since we know $|g_i^*| = O\left(\frac{\lambda n}{m}\right) < n^{2c}$ by (21). We have the identity $\binom{\left(\frac{y_j}{2}\right)}{2} = \binom{\left(\frac{y_j}{2}\right)}{2} \binom{\left(\frac{y_j}{2}\right)-1}{2}$. As a result, there are at most

$$M = \binom{\left(\frac{y_j}{2}\right)-1}{2} \left|d(f(\sigma))\right| < \frac{|z_j| n^{2c} \left(\left(\frac{y_j}{2}\right)\right) \left(\left(\frac{y_j}{2}\right)-1\right)}{|z_j|}$$

such pairs $(z_j, \sigma)$ incident to each $\sigma \in Q$. The total number of $(z_j, \sigma)$ constructed by Step 1 is $M |Q|$ or less.

Put $r < n^{1/5}$, $L \leftarrow |z_j| = n^{11/6}$ and $N \leftarrow \binom{|y_j|}{2} = \Theta(n^{2-10c})$ to apply to Lemma 4.3. Since $rL/N \ln \sqrt{N}$ is $\Theta(n^{2-10c})$, we have a family of $k$-edge sets with sparsity $o(1)$, such that the removal of each element leaves no $r$-cliques in the space $y_j$.

Therefore there exists a $k$-edge set $z_j \subset \left(\frac{y_j}{2}\right)$ incident to

$$\frac{M |Q| \left(\frac{y_j}{2}\right)}{\binom{\left(\frac{y_j}{2}\right)}{2} e^{-o(1)} \left.\frac{|z_j|}{|Q|}\right) = O \left(n^{-\frac{1}{6}+12c} |Q|\right) \ll |Q|,$$
or less $\sigma \in \mathcal{Q}$ in the family of constructed pairs $(z_j, \sigma)$. Step 1-5 chooses these $\sigma$ as $\mathcal{Q}(z_j)$ so $|\mathcal{Q}(z_j)| \ll |\mathcal{Q}|$ holds true. □

**Corollary 4.9.** For each $j \in [q]$, 
$$\mathcal{Q}_j = \{\sigma \in \mathcal{Q}_0 : y(\sigma) = y_j\} \quad \text{and} \quad |\mathcal{Q}_j| < 2^{-j} |\mathcal{Q}_0|,$$
after **BlockedEdges terminates.**

**Proof.** The first statement is clearly true by Steps 1-1 and 1-6 of BlockedEdges; the family $\mathcal{Q}_j$ consists of all $\sigma$ that choose $y_j$ as $y(\sigma)$ after the algorithm finishes.

Every time Loop 1 of BlockedEdges increments $j$, it reduces $|\mathcal{Q}|$ to less than its half by Lemma 4.8. Thus $|\mathcal{Q}| < 2^{-j} |\mathcal{Q}_0|$ after Step 1-6 with loop index $j$. The second statement follows this and $\mathcal{Q}_j \subseteq \mathcal{Q}$. □

**Lemma 4.10.** The shift $t(y)$ contains no $k$-clique.

**Proof.** As seen, $t(y)$ is $t(\sigma) \setminus \left(\frac{y}{2}\right)$ at the root of $C$ for some $\sigma \in \mathcal{Q}_0$ such that $g = \emptyset$. Noting the fact, observe the following two claims.

**Claim 1:** $\left(\frac{y}{2}\right) \cap t(\sigma) \setminus \left(\frac{y}{2}\right) = \emptyset$ for every $\sigma \in \mathcal{Q}_0$.
**Proof:** The size of $\mathcal{Q}_0$ is $e^{O(n^2)} \ll e^q$ by Lemma 4.2 (i). The family $\mathcal{Q}_0$ is empty since $|\mathcal{Q}_j| < 2^{-j} |\mathcal{Q}_0|$ for $j \in [q]$. There exists no $\sigma \in \mathcal{Q}_0$ such that $y(\sigma) = y_q$. By Lemma 4.4, $\Phi \circ d(f(\sigma))$ contains no edge both of whose end points are in $y_q$. Thus $t(\sigma) \setminus \left(\frac{y}{2}\right)$ does not intersect with $\left(\frac{y}{2}\right)$ for $\sigma \in \mathcal{Q}_0$ due to (25). The claim follows. □

**Claim 2:** For each $j \in [q]$, $\left(\frac{y}{2}\right) \cap t(y)$ contains no $n^{1/5}$-clique.
**Proof:** It is true for $j = q$ by Claim 1: since $t(y) = t(\sigma) \setminus \left(\frac{y}{2}\right)$ for some $\sigma \in \mathcal{Q}_0$. For $j \in [q - 1]$, $t(y)$ is disjoint with $z_j \subset \bar{z}$ by Corollary 4.7. The removal of $z_j$ annihilates $n^{1/5}$-cliques in $\left(\frac{y}{2}\right)$. The edge set $\left(\frac{y}{2}\right) \cap t(y)$ is free from an $n^{1/5}$-clique. □

Therefore, $t(y)$ contains no $n^{1/5}q$-clique, thus no $k$-clique either. This proves the lemma.

With the contradiction that $C$ has a global term containing no $k$-clique, the proof of Proposition 4.1 is complete.

### 4.5. Remarks

The standard proof of the super-polynomial monotone circuit complexity of CLIQUE is described in [11]. It was first invented by Razborov, then improved by Alon and Boppana.

In the proof, logical assignments $S$ called *negative examples* are constructed based on random $(l - 1)$-coloring of vertices; an edge $(v_1, v_2)$ exists if and only if the colors of $v_i$ are distinct. It is seen that a negative example contains no $k$-clique. The monotone circuit $C$ returns false to every such $S$. At a node $\alpha$, cut terms short into the core of a sunflower. Call $S$ for which the modified circuit returns true *false positive*. Argue that a false positive $S$ is newly created at $\alpha$, only if every petal of a sunflower contains an edge of a term that does not belong to $S$. Its probability is sufficiently small so that false positives are maintained as a minority among the $(l - 1)^n$ negative examples throughout the process. At the root of the circuit, it is impossible that short terms say no to the majority of remaining negative examples but the false positives. This scheme is generalized to so-called method of approximation.

Both the approximation based on false positive and the shift method shorten terms at $\alpha$ essentially, either by sunflowers or extension generators. There is a difference in
the inductive invariants. The former maintains the property that \( C \) returns false for a majority of negative examples, whereas the latter shows that the union of related generators \( g \) still contains a \( k \)-clique \( c \notin C_C \).

The method of approximation focuses on the truth of negative examples constructed before the cutting algorithm starts. In contrast, the shift method dynamically creates a counter example \( t(y) \). The global term \( t(y) \) is disjoint with the blocked edge set \( z = z_1 \cup z_2 \cup \cdots \cup z_q \) whose removal annihilates smaller cliques, i.e., the algorithm \textsc{Shift} empties \( z \) in \( t(y) \). We will use this feature for a non-monotone case.

The method of approximation is applicable to the problem of perfect matching \([5]\). The result implies \( e^{O(n^2)} \) monotone circuit complexity of \textsc{k-Matching} for \( l \in [n^\epsilon, n/2] \cap \mathbb{Z} \) where \( \gamma \) is a given constant.

4.6. Constraint on \( \sigma \) for \( C \) with Bounded Depth
A quadruple \( \sigma = (g,g_1,g_2,\alpha) \in Q_0 \) is said to be regular if \( g \subset Y(\sigma) \), where

\[
Y(\sigma) \overset{\text{def}}{=} y_1 \cup y_2 \cup \cdots \cup y_{(\sigma)} = \bigcup_{i \in [n]} y_i.
\]

In this subsection, we show that the shift \( t(y) \) can be constructed with only regular \( \sigma \) when \( \text{depth}(C) \) is bounded by a poly-log function of \( n \). The observation will play an important role in the further discussions to handle a non-monotone circuit.

With a given split \( y = (y_1,y_2,\ldots,y_q) \) of \( [n] \), the rank of a set \( s \subset [n] = \{1,2,\ldots,n\} \), denoted by \( \text{rank}(s) \), is the largest integer \( j \in [n] \) such that \( s \cap y_j \neq \emptyset \) if \( s \neq \emptyset \); otherwise it is zero. In addition, the rank of \( \sigma = (g,g_1,g_2,\alpha) \in Q_0 \) is that of \( g \). It is denoted by \( \text{rank}(\sigma) \) also. Notice that \( \text{rank}(\sigma) = 0 \) for \( \sigma \) chosen by Step 4 of \textsc{Shift}.

Modify \textsc{Shift} so that its step 2 calls \textsc{BlockedEdges2} in Fig.9. The primary difference from \textsc{BlockedEdges} is that there should be no \( \sigma \in Q_0 \) such that \( y(\sigma) = y_j \) and \( \text{rank}(\sigma) > j \); due to the constraint that only regular \( \sigma \) are allowed. We split \( Q_j \) into families \( Q_j^{(1)}, Q_j^{(2)},\ldots,Q_j^{(j)} \) containing \( \sigma \) with different ranks. Step 1-4 of \textsc{BlockedEdges2} finds

\[
Q_j^{(r)} \overset{\text{def}}{=} \{ \sigma \in Q_0 : y(\sigma) = y_j \land \text{rank}(\sigma) = r \}.
\]

Let

\[
q' \overset{\text{def}}{=} n^{4\epsilon} \ll q = n^{5\epsilon}.
\]

For each \( r \in [2\text{depth}(C) \cdot q'] \cup \{0\} \), the families \( Q_r^{(r)}, Q_{r+1}^{(r)},\ldots,Q_{r+q'}^{(r)} \) reduce their sizes exponentially as the lower index \( j = r + i \) increases, just as \( Q_j \) in Corollary 4.9.

Confirm it in the following lemma.

**Lemma 4.11.** For each \( r \in [2\text{depth}(C) \cdot q'] \cup \{0\} \) and \( i \in [q - r] \cup \{0\} \), Step 1-4 of \textsc{BlockedEdges2} correctly computes \( Q_{r+1} \) defined by (26) so that

\[
|Q_{r+1}^{(r)}| \leq 2^{-i} |Q_r^{(r)}| \quad \text{where} \quad Q_r^{(r)} \overset{\text{def}}{=} \{ \sigma \in Q_0 : \text{rank}(\sigma) = r \}.
\]

**Proof.** Before Step 1-3, there exists an edge set \( z_j \in \left( \frac{y_j}{n^{1/6}} \right) \) whose removal annihilates \( n^{1/5} \)-cliques in \( y_j \), and such that \( |Q_r^{(r)}(z_j)| \ll |Q_j^{(r)}| \) for each \( r \leq j \), by the argument identical with the proof Lemma 4.8. Not only that, the number of such \( z_j \) is at least \( \left( \frac{y_j}{n^{1/6}} \right) (1 - n^{-10\epsilon}) \): Exclude \( z_j \) found so far. As long as there are \( \left( \frac{y_j}{n^{1/6}} \right) n^{-10\epsilon} \) remaining \( z_j \), we can continue the same process to find another \( z_j \) such that \( Q_j^{(r)}(z_j) \)
Algorithm BlockedEdges2
Inputs and Outputs: Same as BlockedEdges.
begin
1. for \( j \leftarrow 1 \) to \( q \) do
   1-1. for each \( \sigma \in Q \) such that \( \text{rank} \ (\sigma) \leq j \) do \( y(\sigma) \leftarrow y_j \) and \( f(\sigma) \leftarrow f_j(\sigma) \);
   1-2. for each \( r \in [j] \) and \( z_j \in \left( \binom{n}{r} \right) \) do
       \( Q_j^{(r)} \leftarrow \{ \sigma \in Q : \text{rank} \ (\sigma) = r \} \);
       and \( Q_j^{(r)}(z_j) \leftarrow \{ \sigma \in Q : \text{rank} \ (\sigma) = r \land z_j \cap d(f(\sigma)) \neq \emptyset \} \);
   1-3. Find and fix \( z_j \) whose removal annihilates \( n^{1/5} \)-cliques in \( y_j \), and such that
       \( \left| Q_j^{(r)}(z_j) \right| < \frac{1}{4} \left| Q_j^{(r)} \right| \) for all \( r \in [j] \cup \{0\} \);
   */ Choose any \( z_j \) if such one does not exist. */
   1-4. for each \( r \in [j] \) do \( Q_j^{(r)} \leftarrow Q_j^{(r)} \setminus Q_j^{(r)}(z_j) \);
       */ It is the family of \( r \) with rank \( r \), whose \( y(\sigma) \) has been decided as \( y_j \). */
   1-5. \( Q \leftarrow Q \setminus \bigcup_{r=1}^{q} Q_j^{(r)} \);
2. end for
end

Fig. 9. To Determine \( z_j, y(\sigma) \) and \( f(\sigma) \) for Regular \( \sigma \)

is sufficiently small. (The proof of Lemma 4.8 works even if the number of available \( z_j \)
is \( \left( \binom{n}{q} \right)n^{-10c} \).) When it stops, we have found a family of \( z_j \) whose sparsity is at most
\(- \ln (1 - n^{-10c}) \).

In other words, the number of \( z_j \) such that \( \left| Q_j^{(r)}(z_j) \right| \ll \left| Q_j^{(r)} \right| \) before Step 1-3 is less
than \( \left( \binom{n}{q} \right)n^{-10c} \). Find such a family of \( z_j \) for every \( r \in [j] \cup \{0\} \) and join them. Its size
is no more than \( \left( \binom{n}{q} \right) q \cdot n^{-10c} = \left( \binom{n}{q} \right)n^{-5c} \). The step 1-3 correctly finds one \( z_j \)
that does not belong to it, i.e., \( z_j \) such that \( \left| Q_j^{(r)}(z_j) \right| \ll \left| Q_j^{(r)} \right| \) for all \( r \in [j] \cup \{0\} \).

Having confirmed that \( \left| Q_j^{(r)}(z_j) \right| < \frac{1}{4} \left| Q_j^{(r)} \right| \) at 1-3, prove the inequality \( \left| Q_{r+1}^{(r)} \right| \leq 2^{-i} \left| Q^{(r)} \right| \) similarly to Corollary 4.9. Right after Step 1-4 when the loop index is \( j \), the ratio
\( \left| Q_j^{(r)}(z_j) \right| / \left| Q_j^{(r)} \right| \) is at most \( \frac{1}{4} \frac{1}{4^j} = \frac{1}{4} < \frac{1}{2} \). The family \( Q_j^{(r)}(z_j) \) becomes \( Q_j^{(r+1)}(z_j) \)
at Step 1-2 with loop index \( j + 1 \). The final \( Q_j^{(r)} \) is decided by Step 1-5 below, thus
(27) \( \left| Q_{j+1}^{(r)} \right| < \frac{1}{2} \left| Q_j^{(r)} \right| \). This inductively shows \( \left| Q_{r+1}^{(r)} \right| \leq 2^{-i} \left| Q^{(r)} \right| \).

The lemma follows. \( \Box \)

Since \( \hat{Q}_0 = e^{O(n^2)} \), the family \( Q_{r+1} \) is empty when \( i = q' = n^{4c} \). It means that each
\( \sigma \) with rank \( r \leq 2\text{depth} \ (C) \cdot q' \) satisfies
\( y(\sigma) \in \{ y_r, y_{r+1}, \ldots, y_{r+q'} \} \), so \( \sigma \) is regular.

Based on the above facts, let us see that LocalShift constructs \( t(y) \) with only
regular \( \sigma \) for \( C \) with poly-log depth. At the root \( \alpha = r(C) \), the only \( \sigma \) used for \( t(y) \) is
By (27), rank whose depth is zero.

Step 2-1 of LOCALSHIFT chooses \( \sigma^* = (g, g_1^*, g_2^*, \alpha) \) incident to a \( k \)-clique \( c^* \subset g \cup y (\sigma) \). By (27), \( \text{rank} (g_i^*) \leq \text{rank} (c^*) \leq \text{rank} (g \cup y (\sigma)) \leq q' \). Thus the rank of \( \sigma_i \), chosen by Step 2-2-1-2-3 of LOCALSHIFT is no more than \( q' \). This is a general statement for \( \alpha \) with depth 1.

Inductively, assume \( \text{rank} (\sigma) \leq q' \cdot \text{depth} (\alpha) \) for every \( \sigma \) occurring at Step 2-1 of LOCALSHIFT in the construction of \( t (y) \). Its \( y (\sigma) \) satisfies (27) so that \( \sigma \) is regular. Step 2-1 chooses \( \sigma^* \) incident to \( c^* \subset g \cup y (\sigma) \) with rank \( \text{rank} (g) + q' \leq q' \cdot \text{depth} (\alpha) + 1 \).

Thus, \( \text{rank} (\sigma_i) \) at Step 2-2-1-2-2-3 does not exceed \( q' (\text{depth} (\alpha) + 1) \leq q' \cdot \text{depth} (\alpha) \).

Formally, the following lemma is proven by induction on \( \text{depth} (\alpha) \).

**Lemma 4.12.** Let \( C \) be a monotone circuit \( C \) with poly-log depth to compute \( n^{1/4} \).

**Clique.** Run Shift with BLOCKED EDGES 2 on \( C \). For every node \( \alpha \) and \( \sigma \in Q_0 \) at \( \alpha \) with rank \( r \in [2 \cdot \text{depth} (C) \cdot q'] \cup \{ 0 \} \), Algorithm LOCALSHIFT constructs \( t (\sigma) \) with only regular \( \sigma' \in Q_0 \) such that \( \text{rank} (\sigma') \leq q' \cdot \text{depth} (\alpha) \) in its recursive calls. \( \square \)

As a result, the shift \( t (y) \) is constructed with only regular \( \sigma \) if \( \text{depth} (C) \) is bounded by a poly-log function.

5. A PROOF OF \( \text{NC} \neq \text{NP} \)

In this section, we show \( \text{NC} \neq \text{NP} \) by proving the following proposition.

**Proposition 5.1.** Let \( k \in \lceil n^{\gamma} \cdot n - n^{\gamma} \rceil \cap \mathbb{Z} \) for a given constant \( \gamma \in (0, 1) \). There exists a constant \( \epsilon \in (0, 1) \) sufficiently smaller than \( \gamma \), such that every Boolean circuit computing \( k \)-CLIQUE with depth \( O (n^\epsilon) \) has size \( \exp (\Omega (n^\epsilon)) \).

The main difficulty of its proof is how to construct a shift without a contradiction, i.e., such that \( \{ e, \neg e \} \not\subset t (y) \) for any \( e \in \binom {\binom {n} {2}} {2} \). We will define a notion called negative tail \( z \subset \binom {n} {2} \) to handle non-monotone terms outside a considered clique \( c \). We will find a good negative tail \( z \) such that the shift method works without creating a contradiction.

5.1. Preparation

Now the given \( C \) is a non-monotone De Morgan circuit with bounded size and depth. It returns true to a truth assignment \( S \) if and only if it contains \( \binom {c} {2} \) for some \( c \in \binom {n} {2} \).

Consider all such \( S \). Recall that \( S \) is a set of literals where positive literal \( e \in \binom {n} {2} \) is distinguished from a negative literal \( \neg e \). However, if we say the negative literal set of a term \( t \), it means the set of edges \( e \in \binom {\binom {n} {2}} {2} \) such that \( \neg e \in t \).

Denote by \( z \subseteq \binom {n} {2} \) an edge set such that \( z \setminus \binom {c} {2} \) is the negative literal set of \( S \supseteq \binom {c} {2} \). We call such \( z \) a negative tail of \( c \). A pair \( (c, z) \) determines a truth assignment \( S \); its positive literal set is \( \binom {c} {2} \cup \left( \binom {n} {2} \setminus z \right) \) and negative literal set \( z \setminus \binom {c} {2} \). It is said to be a clique-tail pair. There exists a global term at \( r (C) \) contained in every such \( S \), denoted by \( t_0 (c, z) \). We choose unique \( t_0 (c, z) \) arbitrarily for each \( (c, z) \).

Without loss of generality, assume a Boolean circuit \( C \) to compute \( k \)-CLIQUE of size no more than \( \exp (n^{\gamma}) \) and depth \( n^{\gamma} \). For each negative tail \( z \subseteq \binom {n} {2} \), consider all the clique-tail pairs \( (c, z) \). As defined in Section 4.2 with Fig. 3, there exists either a unique term \( t \) to generate the global term \( t_0 (c, z) \) at each node \( \alpha \), or no such term at \( \alpha \). We say that the subset

\[
d = t \cap \binom {c} {2}
\]
of $t$ is a $z$-dominant at $\alpha$. We will treat $z$-dominants equivalent to dominants in Proposition 4.1. Most of the proof steps perform toward a good choice of negative tail $z$, such that the algorithm $\text{SHIFT}$ works on $z$-dominants.

To prove Proposition 5.1 it suffices to show it for the case $k = \sqrt[4]{n}$. It is not difficult to see that another $l \in [n^{\gamma}, n - n^\gamma]$ is reduced to this case.\footnote{For other clique sizes $l \in [n^{\gamma}, n - n^\gamma]$, consider only $c$ containing some fixed set $\hat{c} \subset [n]$ of size less than $k$. In addition, choose a subset $[n']$ of $[n] \setminus \hat{c}$. Focus on clique-tail pairs $(c, z)$ such that $z \supseteq ([n] \setminus [n'])$. Consider cliques $c' = c \setminus \hat{c}$ in the space $[n']$ only. With proper choices of $\hat{c}$ and $n'$, another $k$ is always reduced to the case $k = \sqrt[4]{n}$.}

Fix each negative tail $z \subseteq \binom{[n]}{2}$. Run the algorithm $\text{CLIQUEGENERATORS}$ on $z$-dominants $d$ in place of dominants at every $\alpha$. This constructs $\mathcal{C}_\alpha(z)$, $\mathcal{Y}(z)$ and $\mathcal{Q}_0(z)$ similarly to $\mathcal{C}_\alpha$, $\mathcal{Y}$ and $\mathcal{Q}_0$, respectively. Use the same notation as in Section 4.4 except for these. Especially,

$$q = n^{5e}, \lambda_c = n^e, l_y = \frac{n}{q}, \lambda_y = n^{10e} \quad \text{and} \quad q' = n^{4e}$$

remain the same.

It is noteworthy that a $z$-dominant can be empty at some $\alpha$. Analogous to the monotone case, for each negative tail $z \subseteq \binom{[n]}{2}$, we define that $c$ is generated at $\alpha$ if there exists a $z$-dominant $d$ at $\alpha$ contained in $\binom{z}{2}$. This is equivalent to the existence of a term $t \in \text{DNF}(\alpha)$ that generates $t_0(c, z)$ in $C$. An example for an empty $z$-dominant occurs at a leaf $\alpha$ associated with an negative literal $\neg e$ such that $e \in z$. The term $\{\neg e\} \in \text{DNF}(\alpha)$ can generate $t_0(c, z)$ such that $e \notin \binom{z}{2}$ in $C$; thus $d = \emptyset$ at $\alpha$.

Also notice that the algorithm $\text{CLIQUEGENERATORS}$ acts on $k$-cliques $c$ generated at $\alpha$. It is possible that $q$ is non-empty even if $d = \emptyset$.

The algorithms and related terminologies for the monotone case are well-defined over $z$-dominants, except for some minor adjustments. We will clarify the difference as we construct a shift over $z$-dominants step by step.

5.2. How to Apply the Shift Method to a Non-Monotone Circuit

For the given non-monotone circuit $C$ and each negative tail $z \subseteq \binom{[n]}{2}$, we have constructed $\mathcal{C}_\alpha(z)$, $\mathcal{Y}(z)$ and $\mathcal{Q}_0(z)$ similarly to the monotone case. Run the algorithm $\text{SHIFT}$ over $z$-dominants. Inductively maintain the relation

$$\left(t(\sigma) \setminus \binom{g}{2}\right) \cap z \subseteq \Phi,$$

instead of (25). Find a global term $t(y)$ such that

$$t(y) \cap z' \cap z = \emptyset,$$

(28)

rather than $t(y) \cap z' = \emptyset$. Here $z'$ is a blocked edge set over $z$-dominants constructed by $\text{BLOCKEDEDGES}$.

Let us require that the removal of the initially chosen negative tail $z$ leave no $k$-clique in the edge space $\binom{[n]}{2}$. By Lemma 4.3 there are a majority of such $z$. We have the mapping

$$z \mapsto z'$$

from a negative tail $z$ to the resulting blocked edge set $z'$. We now want a fixed point of $z \mapsto z'$, i.e., a negative tail $z$ mapped to the same $z' = z$ that is chosen $\text{BLOCKEDEDGES}$.

If we can always find such a fixed point, we have a proof of Proposition 5.1. Algorithm $\text{SHIFT}$ would construct $t(y)$ over $z$-dominants satisfying the following:
(a) \( t(y) \cap z = \emptyset \) due to (28) with \( z' = z \),
(b) \( t(y) \) contains no positive literal of an edge in \( z \) by (a), and
(c) no negative literal of an edge outside \( z \), since we constructed \( t(y) \) only from the
global terms \( f_0(c, z) \) for the fixed negative tail \( z \), and
(d) \( t(y) \) such that (a) contains no \( k \)-cliques since the removal of \( z \) annihilates \( k \)-cliques
in the edge space.

Therefore \( t(y) \) is a shift, a global term of \( C \) free from a contradiction and \( k \)-clique.
This means to the impossibility of the circuit \( C \) to compute \( k \)-CLIQUE, proving Proposition 5.1.

We may produce many blocked edge \( z' \) from a negative tail \( z \). Thus, we consider the relation \( (z, z') \) rather than the mapping \( z \mapsto z' \). Its fixed point is a diagonal element \((z, z)\).

Our proof in the next subsection consists of 7 steps. The first four finds a fixed point
of the relation \((z, z')\):

**Step 1:** Construct a family \( Z_1 \) of tuples \((z, y) = (z, y_1, y_2, \ldots, y_q)\).

Enumerate all the possible negative tails \( z \) and splits \( y \) of \( |n| \) such that Conditions (i)
and (ii) in Section 4.3 hold true. The family forms a majority.

**Step 2:** Find and fix \( y \) incident to the maximum number of \((z, y) \in Z_1 \). Consider only the negative tails \( z \) incident to this \( y \).

**Step 3:** Construct a family \( Z_2 \) of \( q \)-tuples \( z = (z_1, z_2, \ldots, z_q) \).

Use a space-proportional split defined in Section 2.5.2 to split \( z \) into a majority of
\((z_1, z_2, \ldots, z_q)\).

**Step 4:** Fix \( z \in Z_2 \) similarly to BLOCKEDEDGES2. The edge set \( z = z_1 \cup z_2 \cup \cdots \cup z_q \) is the fixed point we find.

Use a similar strategy to the monotone case to determine \( z_1, z_2, \ldots, z_q \) one by one out
of the choices preserved in the previous steps. Choose \( z_j \) that avoids \( d(f(z)) \cap z_j \neq \emptyset \)
for any \( \sigma \in \mathcal{Q}_0(z) \).

**Step 5:** With the negative tail \( z \) fixed by Step 4, run Algorithm SHIFT on \( z \)-dominants
to construct \( t(\sigma) \) for every \( \sigma \in \mathcal{Q}_0(z) \). This determines a shift \( t(y) \) at the root \( y(C) \).

**Step 6:** Show that the constructed shift \( t(y) \) contains no contradiction.

**Step 7:** Show that \( t(y) \) contains no \( k \)-clique.

Steps 5-7 perform just as discussed above.

When Step 3 splits \( z \) into a majority of \( z = (z_1, z_2, \ldots, z_q) \) space-proportionally, it is
with a split \( X = (X_1, X_2, \ldots, X_q) \) of the whole edge space \( \binom{|n|}{2} \). (Its components are
\( X_j = (y_j \times (y_1 \cup y_2 \cup \cdots \cup y_{j-1})) \cup \binom{y_j}{2} \).) Thus \( z_j \subset X_j \) rather than \( z_j \subset \binom{y_j}{2} \) in the
monotone case. It is to annihilate contradictions over \( \binom{|n|}{2} \) in addition to undesired small cliques.

An algorithm similar to BLOCKEDEDGES2 determines \( z_1, z_2, \ldots, z_q \) in the order.
Here we have the following edge direction problem: Due to \( z_j \subset X_j \) rather than
\( z_j \subset \binom{y_j}{2} \), an edge may be considered TWICE. This possibly creates \( \sigma = (g, g_1, g_2, \alpha) \)
violating our rule \( d(f(\sigma)) \cap z = \emptyset \) equivalent to Condition (b) in Section 4.3. For
equivalent, suppose that \( y(\sigma) = y_1 \) and \( g \cap y_2 \neq \emptyset \). When the algorithm decides
\( z_2 \subset X_2 = (y_2 \times y_1) \cup \binom{y_2}{2} \), it may conflict with the above \( \sigma \); due to \( g \cap y_2 \neq \emptyset \), the
edge set \( d(f(\sigma)) \) may intersect with \( z_2 \) although \( \sigma \) has already decided \( y(\sigma) = y_1 \).

To prevent it, we restrict \( \sigma \) to be regular. By the observation in Section 4.6, regular \( \sigma \)
are sufficient to construct a shift for \( C \) with bounded depth. One can check that the
above edge direction problem does not occur with this constraint on \( \sigma \).
5.3. A Proof of Proposition 5.1

Given a circuit $C$ of size at most $e^{n^2}$ and depth $n^2$ to compute $k$-CLIQUE, perform the following 7 steps to construct a global term $t(y)$.

**Step 1:** Construct a family $Z_1$ of tuples $(z, y) = (z, y_1, y_2, \ldots, y_q)$.

Consider each negative tail $z \subset \binom{n}{2}$ of size $n^{11/6}$ whose removal leaves no $n^{1/5}$-cliques in $\binom{n}{2}$. Running CLIQUE_GENERATORS over $z$-dominants, this determines $\mathcal{Y}(z)$. Construct every possible $y = (y_1, y_2, \ldots, y_q)$ with $\mathcal{Y}(z)$ as in the monotone case; enumerate all the splits $y$ of $|n|$ such that $|y_j| = n/q$ and $y_j \in \mathcal{Y}(z)$ for $j \in [q]$. Store all such $(z, y)$ in $Z_1$.

**Step 2:** Find and fix $y$ incident to the maximum number of $(z, y) \in Z_1$. Consider only the negative tails $z$ incident to this $y$.

**Step 3:** Construct a family $Z_2$ of $q$-tuples $z = (z_1, z_2, \ldots, z_q)$.

Each obtained $z_j$ is meant to be a blocked edge set acting similarly to the monotone case as explained in Section 5.2. Follow the sub-steps below.

**Step 3-1:** Split $z$ into $z$ space-proportionally.

For each $j \in [q]$, define

$$X_j \overset{\text{def}}{=} S_j \setminus S_j - 1, \quad \text{where} \quad S_j = \left( \frac{y_1 \cup y_2 \cup \cdots \cup y_j}{2} \right),$$

where $S_0 = \emptyset$. In other words, $X_j = (y_j \times (y_1 \cup y_2 \cup \cdots \cup y_{j-1})) \cup \left( \frac{y_j}{2} \right)$. This determines the split

$$X = (X_1, X_2, \ldots, X_q)$$

of $\binom{n}{2}$. Split $z$ into $z = (z_1, z_2, \ldots, z_q)$ space-proportionally to $X$. (The definition of space-proportional split and related claims are in Section 2.5.2 and Lemma 2.9.) The family of all such $z$ is $Z_2$.

**Step 3-2:** Define related objects.

Each $z \in Z_2$ determines the negative tail $z$ by the mapping

$$z \mapsto z = z_1 \cup z_2 \cup \cdots \cup z_q.$$  \hspace{1cm} (29)

Identify $z$ with $z$: If we say $z$-dominant, it means $z$-dominant such that $z \mapsto z$. Also, put $C_{\alpha}(z) = C_{\alpha}(z)$, $\mathcal{Y}(z) = \mathcal{Y}(z)$ and $Q(z) = Q(z)$. Let

$$Q_0 \overset{\text{def}}{=} \bigcup_{z \in Z_2} Q(z) \quad \text{and} \quad Q = Q_0.$$  

The latter will change dynamically as in the monotone case.

**Step 3-3:** Show some properties of $Z_2$.

For each $z \in Z_2$, define $\sigma = (g, g_1, g_2, \alpha) \in Q(z)$ incident to $c \not\in C_{\alpha}(z)$ similarly to the monotone case: if and only if $c$ and $\sigma$ satisfy Conditions I–IV in Section 4.4 over $z$-dominants instead of dominants, and V. If $\alpha$ is a leaf of $C$ associated with a negative literal $-e$, $e \in \binom{n}{2}$, then $g_1 = g_2 = \emptyset$.

Observe properties of $Z_2$.

**Lemma 5.2.** The following four statements hold true.

(i) $y_j \in \mathcal{Y}(z)$ for each $z \in Z_2$ and $j \in [q]$.
(ii) For each \( z \in Z_2 \), \( c \not\in C_\alpha(z) \) generated at a node \( \alpha \), and generator \( q \subset c \) found by CLIQUEGENERATOR running on \( z \)-dominants at \( \alpha \), there exists \( \sigma = (q, g_1, g_2, \alpha) \in Q(z) \) incident to \( c \).

(iii) \( |z_j| = |X_j| \left( 2n^{-1/6} - o \left( n^{-1/6} \right) \right) \) for \( j \in [q] \). Especially, the smallest value among \( |z_j| \) is \( |z_1| = n^{11/6} - 10\epsilon - o \left( n^{11/6} - 10\epsilon \right) \).

(iv) The complement sparsity of \( Z_2 \) is \( \Omega(\lambda_y) \), i.e.,

\[
|Z_2| = \prod_{j=1}^{q} \left( \frac{|X_j|}{|z_j|} \right) \left( 1 - e^{-\Omega(\lambda_y)} \right).
\]

**Proof.** (i): The negative tail \( z \) determined by \( z \) with (29) is the first component of \((z, y)\) that belongs to \( Z_1 \). The tuple satisfies \( y_j \in \mathcal{Y}(\lambda_y) \) for every \( j \in [q] \) due to the construction by Step 1.

(ii): This is the counterpart of Lemma 4.2(ii) for this \( C \), and is proven by the same argument.

(iii): \( z = (z_1, z_2, \ldots, z_q) \) are constructed by space-proportional splits of \( z \). The ratio of the term size \( |z_j| \) to the space size \( |X_j| \) is equal to

\[
\frac{|z|}{\binom{n}{2}} = \frac{n^{11/6}}{n(n-1)/2} = 2n^{-1/6} - o \left( n^{-1/6} \right).
\]

Thus \( |z_j| = |X_j| \left( 2n^{-1/6} - o \left( n^{-1/6} \right) \right) \), leading to the first statement of (iii). Especially, \( |z_1| = |z| \left( \frac{|y_1|}{\binom{n}{2}} \right) = n^{11/6} - 10\epsilon - o \left( n^{11/6} - 10\epsilon \right) \). The second statement of (iii) follows it.

(iv): The complement sparsity of \( Z_1 \) is \( \Omega(\lambda_y) \). For, Step 1 finds \( z \in \binom{\binom{n}{2}}{\binom{n}{11}/n} \) of complement sparsity \( \Theta(n^{7/30}) \) with Lemma 4.3 as in Lemma 4.6. For each negative tail \( z \), we have constructed \( y \) of complement sparsity \( \Omega(\lambda_y) \) with Lemma 2.8 by the same way\(^9\) to find the input \( y \) to **SHIFT** in Section 4.4. The number of obtained \((z, y)\) is at least \( \binom{\binom{n}{2}}{\binom{n}{11}/n} \prod_{j=1}^{q} \left( \frac{n^{11/6}}{n/q} \right)^{n/2} \times \frac{n}{n/q} \times \left( 1 - e^{-\Omega(n^{7/30})} \right)^{n/2} \left( 1 - e^{-\Omega(\lambda_y)} \right) = 1 - e^{-\Omega(\lambda_y)} \).

Thus the complement sparsity of \( Z_1 \) is \( \Omega(\lambda_y) \).

It means that the complement sparsity of all \( z \) found by Step 2 is \( \Omega(\lambda_y) \) also\(^{10}\). Since \( z \) are constructed by space-proportional splits of \( z \) such that \( q \ln n \ll \lambda_y \), their complement sparsity is \( \Omega(\lambda_y) \) by Lemma 2.9. This proves Statement (iv). \( \square \)

---

\(^9\)Here \( |z_j| \) may have a constant deviation from \( |X_j| \cdot |z|/\binom{\binom{n}{2}}{\binom{n}{11}/n} \) due to the floor or ceiling function applied in the proof of Lemma 2.9. It does not affect the ratio \( |z_j|/|X_j| = 2n^{-1/6} - o \left( n^{-1/6} \right) \).

\(^{10}\)Put \( r = n^{1/3} \), \( L = |z| = n^{11/6} \) and \( N \leftarrow \binom{\binom{n}{2}}{\binom{n}{11}/n} \) to apply to Lemma 4.3. Since \( rL/N \ln \frac{N}{rL} \gg 1 \), we have a family of \( k \)-edge sets with complement sparsity \( \Theta \left( \frac{L^2}{N} \right) = \Theta \left( n^{7/30} \right) \), such that the removal of each element annihilates \( r \)-cliques in \( \binom{\binom{n}{2}}{\binom{n}{11}/n} \).

\(^{11}\)The family \( \mathcal{Y}(\lambda_y) \) of valid sets \( y \in \binom{\binom{n}{2}}{\binom{n}{11}/n} \) has complement sparsity \( \Omega(\lambda_y) \) by (22). By Lemma 2.8, the family of \( y = (y_1, y_2, \ldots, y_q) \), \( y_j \in \mathcal{Y}(\lambda_y) \) such that \( y_j \) are pairwise disjoint has the same complement sparsity bound.

\(^{12}\)The number of \((z, y)\) \in \( Z_1 \) is \( \binom{\binom{n}{2}}{\binom{n}{11}/n} \prod_{j=1}^{q} \left( \frac{n^{11/6}}{n/q} \right)^{n/2} \times \left( 1 - e^{-\Omega(\lambda_y)} \right)^{n/2} \left( 1 - e^{-\Omega(\lambda_y)} \right) = 1 - e^{-\Omega(\lambda_y)} \). There exists at least one \( y \) incident to \( \binom{\binom{n}{2}}{\binom{n}{11}/n} \left( 1 - e^{-\Omega(\lambda_y)} \right) \) negative tails \( z \).
Algorithm BLOCKEDEdges3

Inputs:
1. Family $Z_2$ of $z = (z_1, z_2, \ldots, z_q)$ constructed by Step 3-1 of this proof.
2. Family $\hat{Q}_0$ of $\sigma$ constructed by Step 3-2 of this proof.
3. Mapping $f_j(\sigma), j \in [q]$ constructed by Step 1 of SHIFT.

Outputs: (i) Uniquely determined $z \in Z_2$, (ii) mapping $y : \hat{Q}_0 \rightarrow \{y_1, y_2, \ldots, y_q\}$, and (iii) mapping $f : Q_0 \rightarrow \hat{Q}_0$.

begin
1. for $j \leftarrow 1$ to $q$ do
   1-1. for each $\sigma \in Q$ such that $\text{rank} (\sigma) \leq j$ do $y (\sigma) \leftarrow y_j$ and $f (\sigma) \leftarrow f_j (\sigma)$;
   1-2. $U_j \leftarrow$ family of $z_j$ incident to tuples $(z_{j+1}, z_{j+2}, \ldots, z_q)$ of sparsity $\lambda (j)$ in the space $\Pi_{i=j+1}^q (\hat{X}_{j,i})$;
      /* We have current tuples $(z_j, z_{j+1}, \ldots, z_q)$ (projections of $z \in Z_2$) incident to the already fixed
       $(z_1, z_2, \ldots, z_{j-1})$, which satisfy the $j-1$st invariant condition. By this step, we focus on $z_j$ inci-
       dent to a sufficiently large number of $(z_{j+1}, z_{j+2}, \ldots, z_q)$. */
   1-3. for each $r \in [j]$ and $z_j \in U_j$ do
      $Q_j^{(r)} \leftarrow \{ \sigma \in Q : \text{rank} (\sigma) = r \};$
      and $Q_j^{(r)} (z_j) \leftarrow \{ \sigma \in Q : \text{rank} (\sigma) = r \wedge z_j \wedge d (f (\sigma)) \neq \emptyset \};$
   1-4. Find and fix $z_j$ such that $|Q_j^{(r)} (z_j)| < \frac{1}{2} |Q_j^{(r)}|$ for all $r \in [j] \cup \{0\}$;
      /* Choose any $z_j$ if such one does not exist. */
   1-5. for each $r \in [j] \cup \{0\}$ do $Q_j^{(r)} \leftarrow Q_j^{(r)} \setminus Q_j^{(r)} (z_j);$  /* It is the family of $\sigma$ with rank $r$, whose $y (\sigma)$ has been decided as $y_j$. */
   1-6. $Q \leftarrow Q \setminus \bigcup_{r=0}^j Q_j^{(r)}$;
2. end for
end

Fig. 10. Determining $z_j, y (\sigma)$ and $f (\sigma)$ for a Non-Monotone Circuit $C$ of Bounded Depth

Step 4: Fix $z \in Z_2$ similarly to BLOCKEDEdges2. The edge set $z$ to which $z$ is mapped by (29) is our choice of negative tail, a fixed point of the relation $(z, z')$ discussed in Section 5.2.

Perform the following sub-steps.

Step 4-1: Run Algorithm BLOCKEDEdges3 in Fig. 10 to determine $z$ uniquely.

BLOCKEDEdges3 is a variant of BLOCKEDEdges2 discussed in Section 4.6. Its difference is Step 1-2 to maintain a sparsity lower bound for the tuples of remaining components. After Step 1-6 is finished with loop index $j$, the following $j$th invariant conditions are satisfied.

(a) There are tuples $(z_{j+1}, z_{j+2}, \ldots, z_q)$ incident to the $(z_1, z_2, \ldots, z_j)$ fixed so far, whose sparsity is at most $\lambda (j) \overset{def}{=} - \ln \left(1 - \frac{j+1}{q^r}\right)$.

(b) $|Q_j^{(r)}| \leq 2^{-j+r} |Q^{(r)}|$, for each $r = 0, 1, 2, \ldots, \min (j, 2 \text{depth} (C) \cdot q')$, where
$Q_j^{(r)} = \{ \sigma \in Q_0 : y (\sigma) = y_j \wedge \text{rank} (\sigma) = r \}$ as in (26) is computed by Step 1-5, and $Q^{(r)} = \{ \sigma \in Q_0 : \text{rank} (\sigma) = r \}$ is the same as in Lemma 4.11.

(c) $d (f (\sigma)) \cap \left(\bigcup_{i=1}^j z_i\right) = \emptyset$ for every $\sigma \in Q_j$ where $Q_j = \{ \sigma \in Q_0 : y (\sigma) = y_j \}$ is the same as in Corollary 4.9.

These are intended to achieve the following qualitative properties.
(a): In deciding \( z_1, z_2, \ldots, z_q \), Condition (a) ensures that there are sufficient number of remaining tuples \((z_{j+1}, z_{j+2}, \ldots, z_q)\) attached to the \((z_1, z_2, \ldots, z_j)\) fixed so far. Before the loop starts, we have a majority of \( z = (z_1, z_2, \ldots, z_q) \) by Lemma 5.2(iv).

(b): This is similar to Corollary 4.9 for Proposition 4.1, with which we will have (27) as in Section 4.6. It claims that the number of currently considered \( \lambda \) reduces exponentially for each rank \( r \), satisfying the regularity constraint.

(c): With the invariant, we will be able to confirm \( d(f(\sigma)) \cap z = \emptyset \) for all regular \( \sigma \). It is the key condition to achieve \( t(\mathbf{z}) \cap \emptyset \neq \emptyset \), i.e., the blocked edge set \( z \) is emptied in \( t(y) \). As noted in Section 5.2 we are searching for such a fixed point \( z \).

**Step 4-2:** Show that Loop 1 of BLOCKEDGEOES3 continues correctly until the final step \( j = q \).

**Lemma 5.3.** For each \( j \in [q] \), Loop 1 of BLOCKEDGEOES3 chooses \( z_j \) satisfying the three \( j^{th} \) invariant conditions.

**Proof.** Verify each of the three conditions below.

(a): Prove by induction on \( j = 0, 1, 2, \ldots, q \). The base case \( j = 0 \) occurs before the first step \( j = 1 \). It is true since we have \( Z_2 \) of sparsity \( -1/e^{ln(\lambda)} < \lambda(0) \). Assume true for \( j - 1 \) and prove true for \( j \).

Before the \( j^{th} \) step, we have \((z_j, z_{j+1}, \ldots, z_q)\) of sparsity \( \lambda(j - 1) \) satisfying (a) by induction hypothesis. The size of \( U_j \) at Step 1-2 of BLOCKEDGEOES3, or the number of \( z_j \) incident to \((z_{j+1}, z_{j+2}, \ldots, z_q)\) of sparsity \( \lambda(j) \), is at least \( (\binom{\lfloor X_j \rfloor}{j}) \frac{1}{q^2} \). Otherwise, there would be too many \((z_j, z_{j+1}, \ldots, z_q)\) NOT incident to \((z_0, z_1, \ldots, z_{j-1})\); their sparsity would be less than

\[
- \ln \left[ \left(1 - \frac{1}{q^2} \right) \left(1 - e^{-\lambda(j)} \right) \right] \leq - \ln \left[ \left(1 - \frac{1}{q^2} \right) j + 1 \right]
\]

contradicting the \( j - 1^{st} \) invariant condition (a). Hence \( |U_j| \geq \left( \binom{\lfloor X_j \rfloor}{j} \right) \frac{1}{q^2} \). Notice that \( |z_j| \) is sufficiently large for the sparsity \( ln q^2 \) by Lemma 5.2(iii), so \( U_j \neq \emptyset \). Step 1-4 of BLOCKEDGEOES3 always chooses one such \( z_j \in U_j \).

(b): For each \( r \leq 2 \text{depth}(C) \cdot q' \), prove \( |Q_j^{(r)}| \leq 2^{-j+r} |Q^{(r)}| \) by induction on \( j = r, r + 1, \ldots, q \). The basis \( j = r \) is true due to \( Q_j^{(r)} \subseteq Q^{(r)} \) \( \Rightarrow |Q_j^{(r)}| \leq 2^{-j+r} |Q^{(r)}| \). This also holds true for \( j = r = 0 \) by putting \( Q_0^{(r)} = Q^{(r)} \).

Assume true for \( j \) and prove true for \( j + 1 \). The induction step is shown similarly to Lemmas 4.8 and 4.11. It suffices to show that Step 1-4 correctly chooses \( z_j \in U_j \) such that \( |Q_j^{(r)}(z_j)| < \frac{1}{2} |Q_j^{(r)}| \) for all \( r \in [j] \cup \{0\} \). If it is true, the ratio \( \frac{|Q_j^{(r)}(z_j)|}{|Q_j^{(r)}|} \) is at most \( \frac{1}{3} \) after Step 1-5 when the loop index is \( j \). The family \( Q_j^{(r)}(z_j) \) becomes \( Q_j^{(r)} \) at Step 1-3 with loop index \( j + 1 \). The final \( Q_j^{(r)} \) is decided by Step 1-5 below, thus \( |Q_j^{(r)}(z_j)| < \frac{1}{2} |Q_j^{(r)}| \), leading to \( |Q_j^{(r)}| \leq 2^{-j+r} |Q^{(r)}| \), the \( j^{th} \) invariant condition (b).
At Step 1-4, there exists $z_j \in U_j$ such that $|Q_j^{(r)}(z_j)| \ll |Q_j^{(r)}|$ for each fixed $r \in [j] \cup \{0\}$ by the same argument as Lemma 4.8. Not only that, as in Lemma 4.11, the number of $z_j$ such that $|Q_j^{(r)}(z_j)| \ll |Q_j^{(r)}|$ is no more than $\binom{n^{y_j}}{|z_j|} \frac{1}{n^{y_j}}$; otherwise we can continue the process to find another $z_j$ such that $|Q_j^{(r)}(z_j)| \ll |Q_j^{(r)}|$ among the remaining ones. Thus, there exists $z_j$ such that $|Q_j^{(r)}(z_j)| \ll |Q_j^{(r)}|$ for all $r \in [j] \cup \{0\}$.

Step 1-4 correctly chooses such $z_j \in U_j$.

The induction step follows, proving Condition (b).

(c): When the loop index is $j$, Steps 1-4 and 1-5 determine $y(\sigma)$ of all $\sigma \in Q_j$ as $y_j$, since $Q_j = \bigcup_{r \in [j] \cup \{0\}} Q_j^{(r)}$. For every such $\sigma$, the edge set $d(f(\sigma))$ does not intersect with $z_j$ due to the two steps. By Lemma 4.4, $d(f(\sigma)) \subset \binom{g^{y_j}(\sigma)}{2} \setminus \binom{q_j}{2}$, i.e., every edge in it is incident to $y(\sigma) = y_j$. No edge in $\bigcup_{i=1}^{j-1} z_i \subset \bigcup_{i=1}^{j-1} X_i$ is, thus $d(f(\sigma)) \cap \left( \bigcup_{i=1}^{j-1} z_i \right) = \emptyset$. □

**Corollary 5.4.** After BLOCKED-EDGES3 terminates, the statements

(i) $y(\sigma) = y_j$ for some $j \in \{\text{rank}(\sigma), \text{rank}(\sigma) + 1, \ldots, \text{rank}(\sigma) + q'\}$, and

(ii) $d(f(\sigma)) \cap z = \emptyset$

hold true for every $\sigma \in \hat{Q}_0$ such that $\text{rank}(\sigma) \leq 2\text{depth}(C) \cdot q'$.

**Proof.** (i): The condition of Step 1-1 constrains $y(\sigma)$ to be $y_j$ such that $j \geq \text{rank}(\sigma)$. Due to the invariant (b) and the size $|Q_j^{(r)}| \leq |\hat{Q}_0| = e^{O(n^r)}$, the family $Q_j^{(r)}$ is empty since $q' = n^{2r} \gg n^r$ (by the same argument as Corollary 4.9). Therefore, $j$ of $y(\sigma) = y_j$ has to be one of $\text{rank}(\sigma), \text{rank}(\sigma) + 1, \ldots, \text{rank}(\sigma) + q'$.

(ii): Let $\sigma \in Q_j^{(r)} \subset \hat{Q}_0$ such that $r \leq 2\text{depth}(C) \cdot q'$. It is regular by (i). We have

$$d(f(\sigma)) \subset \binom{g \cup y(\sigma)}{2} \subset \binom{Y(\sigma)}{2} = \binom{y_1 \cup y_2 \cup \cdots \cup y_j}{2},$$

with Lemma 4.4.

Any edge in $z_i \subset X_i$ such that $i > j$ is incident to a vertex in $y_{j+1} \cup y_{j+2} \cup \cdots \cup y_q$. Thus $d(f(\sigma))$ is disjoint with $\bigcup_{i=j+1}^{q_j} z_i$. By this and the invariant condition (b), $d(f(\sigma)) \cap (\bigcup_{i=1}^{j} z_i) = d(f(\sigma)) \cap z = \emptyset$. □

**Step 5:** With the negative tail $z$ fixed by Step 4, run Algorithm SHIFT on $z$-dominants to construct $t(\sigma)$ for every $\sigma \in \hat{Q}_0(z)$. This determines a shift $t(y)$ at the root $r(C)$.

The only differences from the construction in Proposition 4.1 are:

— Step 2 of SHIFT calls BLOCKED-EDGES3 instead of BLOCKED-EDGES2.

— Step 1 of LOCALSHIFT returns $t(\sigma) = \{x\}$ even if $x$ is a negative literal $-e, e \in \binom{[n]}{2}$.

---

This uses Lemma 5.2(iii). By this we have $\binom{|X_j|-1}{|z_j|-1} = \binom{|X_j|}{|z_j|} \ll \binom{|X_j|}{|z_j|}$, similarly to $\binom{\binom{|X_j|}{2}}{\binom{|z_j|}{2}} = \binom{|z_j|}{|z_j|} \ll \binom{|z_j|}{|z_j|}$ in Lemma 4.8.
**Step 6:** Show that the constructed shift \( t(y) \) contains no contradiction.

We show that \( t(y) \) is contradiction-free, i.e., there is no edge \( e \in \binom{[n]}{2} \) such that \( \{e, \neg e\} \subset t(y) \). It suffices to show that

\[
t(y) \cap z = \emptyset.
\] (30)

If it is true, there is no positive literal \( e \) in the term \( t(y) \) that belongs to the negative tail \( z \). On the other hands, we dealt with only \( z \)-dominants in Step 5. There exists no negative literal \( \neg e \in t(y) \) such that \( e \in \binom{[n]}{2} \) \( \setminus \) \( z \). Thus (30) implies \( t(y) \) is contradiction-free.

We show (30) with Corollary 5.4. Prove a claim similar to Lemma 4.12.

**Lemma 5.5.** Let \( \sigma = (g, g_1, g_2, \alpha) \) be any quadruple in \( \mathcal{Q}_0 \) occurring in a recursive call of \textsc{LocalShift} caused by Step 4 of \textsc{Shift}. Its rank is at most \( q' \cdot \text{depth}(\alpha) \).

**Proof.** By induction on the depth of \( \alpha \). The basis \( \text{depth}(\alpha) = 0 \) occurs in the root call of \textsc{LocalShift} issued by Step 4 of \textsc{Shift}. The rank of \( \sigma = (\emptyset, g_1, g_2, r(C)) \) chosen by the step is 0. The lemma holds for the basis \( \text{depth}(\alpha) = 0 \).

Assume true for \( \text{depth}(\alpha) \) and prove true for \( \text{depth}(\alpha) + 1 \). By induction hypothesis, we have a regular \( \sigma \) whose rank is at most \( q' \cdot \text{depth}(\alpha) \) at Step 2-1 of \textsc{LocalShift}. The quadruple \( \sigma^* = f(\sigma) \) considered by the step is incident to \( c^\ast \) chosen by Step 1-2-1 of \textsc{Shift} such that \( y(\sigma) = y_j \). This \( k \)-clique \( c^\ast \) is contained in \( g \cup y(\sigma) \). By Corollary 5.4 (i), \( y(\sigma) = y_j \) such that \( j \in \{\text{rank}(\sigma), \text{rank}(\sigma) + 1, \ldots, \text{rank}(\sigma) + q'\} \). The sets \( g_j^\ast \) of \( \sigma \) is a subset of \( c^\ast \subset g \cup y(\sigma) \) at Step 2-1 of \textsc{LocalShift}. The quadruples \( \sigma_i \) chosen by Step 2-2-1 or 2-3-2 have rank no more than

\[
\text{rank}(\sigma) + q' \leq q' \cdot \text{depth}(\alpha) + q' = q'(1 + \text{depth}(\alpha)).
\]

This is true because of the depth bound \( n^v \) of \( C \); the above never exceeds the largest possible \( j \) that is \( q = n^v \).

The depth of a child \( \alpha_j \) is at least \( \text{depth}(\alpha) + 1 \). The lemma holds for any node of depth \( \text{depth}(\alpha) + 1 \) occurring in a recursive call to construct \( t(y) \). This proves the induction step. \( \square \)

Let \( \mathcal{Q}_{(y)} \) be the family of \( \sigma \in \mathcal{Q}_0 \) occurring in the construction of \( t(y) \). Every \( \sigma \) in the family satisfies \( \text{rank}(\sigma) \leq 2\text{depth}(C') \cdot q' \) by the lemma. Thus \( d(\sigma) \cap z = 0 \) by Corollary 5.4 (ii).

Similarly to the monotone case, let

\[
\Phi = \bigcup_{\sigma \in \mathcal{Q}_{(y)}} d(f(\sigma)).
\] (31)

By Lemma 5.5 and Corollary 5.4 (ii), we have

\[
\forall \sigma \in \mathcal{Q}_{(y)}, d(f(\sigma)) \cap z = \emptyset \Rightarrow \Phi \cap z = \emptyset.
\] (32)

Show the relation

\[
\binom{[n]}{2} \cap t(\sigma) \setminus \binom{q}{2} \subseteq \Phi \quad \text{for each } \sigma \in \mathcal{Q}_{(y)}.
\] (33)

similarly to Lemma 4.6. It is the generalization of (25) for the non-monotone case as \( \binom{[n]}{2} \) filters the negative literals. The proof slightly differs from Lemma 4.6 in its base case at a leaf node \( \alpha \) associated with a literal \( x \). The following argument proves it: Step 1 of \textsc{LocalShift} returns \( x \) as \( t(\sigma) \). If \( \alpha \) is associated with a negative literal \( x = \neg e, e \in \binom{[n]}{2} \), then \( \binom{[n]}{2} \cap t(\sigma) \setminus \binom{q}{2} \) is empty so (33) is true. If \( x \) is a positive literal...
\( e \in \binom{n}{2} \), we have \( t(\sigma) = \{e\} \) and \( \sigma = (g, \text{vertex}(e), \text{vertex}(e), g) \) since \( \sigma \) is incident to a clique at the leaf \( \alpha \). Whether \( e \) belongs to \( \{e\} \setminus \binom{\alpha}{2} = d(f(\sigma)) \subset \Phi \) or \( \binom{\alpha}{2} \), the relation (33) is true.

The induction step for (33) is shown by the same argument as Lemma 4.6. Therefore, \( z \cap t(\sigma) \setminus \binom{\alpha}{2} = \emptyset \) for every \( \sigma \in Q_t(y) \) by (32) and (33). Since \( t(y) = t(\sigma) \) for \( \sigma = (0, g_1, g_2, r(C)) \) chosen by Step 4 of SHIFT, it proves (30).

**Step 7:** Show that \( t(y) \) contains no \( k \)-clique.

Again, (30) implies it since the removal of edge set \( z \) leaves no \( n^{1/5} \)-clique in \( \binom{n}{2} \).

There exists a global term \( t(y) \in DNF(r(C)) \) that contains neither a contradiction nor \( k \)-clique. We conclude that such a circuit \( C \) to compute \( k \)-CLIQUE does not exist. This completes the proof of Proposition 5.1.

Thus we have:

**Corollary 5.6.** \( \text{NC} \neq \text{NP} \).

### 5.4. Non-Naturalness of the Proof

One common objection to a proof of big hardness through circuit complexity is so-called *natural* proof [14]. A natural argument against a major hardness would create a Boolean circuit \( B \) that breaks a pseudo random number generator. Suppose that there is a proof \( P \) that a polynomial-sized Boolean circuit \( A \) cannot compute a problem in the target complexity class. If \( P \) is natural, it is *constructive*, i.e., it designs a property \( P \) nodes in \( A \) should satisfy, checks it inductively from the leaves, and at the root of \( A \), it draws a contradiction due to the confirmed property \( P \). This type of argument itself creates a polynomial-sized circuit \( B \); it is discussed along the node structure of \( A \).

By the *largetness* of natural \( P \), the property \( P \) computed by \( B \) has a statistical significance, i.e., \( P \)’s occurrence probability is different from that in the pseudo random case. The algorithm \( B \) can be used to break a pseudo random number generator in polynomial time. Thus such \( P \) is unlikely to exist.

The shift method finds a fixed point of the mapping \( z \rightarrow z' \) (\( z' \): a negative tail, \( z' \): resulting blocked edge set) at the root of \( A \) only. It is based on counting, creating no \( B \) as above. In [14] it is also pointed out that a counting-based discussion could be non-natural.

The precise condition the shift method checks is the following: there exists an edge set \( z \), a) of size \( n^{1/6} \), b) whose removal annihilates \( n^{1/5} \)-cliques, and c) such that there is a global term \( t_0 \), c)-1 whose positive literal set is disjoint with \( z \), and c)-2 whose negative literal set is contained in \( z \).

This condition is not inductively maintained in the non-monetone shift method. The construction of \( t_0 \) for given \( z \) is inductive, but is not aware of Condition c)-2). It is for a monotone case allowing \( t \) to contain contradictions. A relevant edge set \( z \) is chosen at the root of \( A \) statistically in the framework of the Hamming space.

Hence a circuit \( B \) to break a random number generator is not created.

### 6. FLATTENING THE SHIFT FOR GENERAL \( C \)

In this section, we further modify the construction to show an exponential size lower-bound of \( C \) without the constraint on depth(\( C \)). We prove the following theorem.

\[^{14} \text{For its clarity at a leaf node \( \alpha \) associated with literal \( x \), suppose \( x \) is a negative literal, or a positive literal \( x = e \) such that \( e \notin z \). Then \( z \cap t(\sigma) \setminus \binom{\alpha}{2} = z \cap (\{x\} \setminus \binom{\alpha}{2}) \) is empty. If \( x = e \in z \), any \( z \)-dominant at \( \alpha \) is contained in a generated clique \( c \) such that \( e \in \binom{\alpha}{2} \). Therefore, \( g \supset \text{vertex}(e) \) meaning \( z \cap t(\sigma) \setminus \binom{\alpha}{2} = \emptyset \).}
\]
THEOREM 6.1. Let \( k \in [n^\gamma, n - n^\gamma] \cap \mathbb{Z} \) for a given constant \( \gamma \in (0, 1) \).

\[
\mathcal{L}(k\text{-CLIQUE}) = \exp(\Omega(n^\epsilon))
\]

holds true for a constant \( \epsilon > 0 \) sufficiently smaller than \( \gamma \).

For simplicity, assume we are given a polynomial-sized circuit \( C \) to compute \( k\text{-CLIQUE} \) for \( k = n^{1/4} \). Let \( C_t \) be the derivation graph of a term \( t \in \text{DNF}(\sigma) \) as defined in Section 4.2. Denote by \( C' \subseteq C \) a subgraph of \( C \) that is a Boolean circuit, say a sub-circuit of \( C \). Let \( \alpha (C') \subseteq C' \) be the sub-circuit of \( C' \) rooted at \( \alpha \) containing all its descendents. Such \( C' \subseteq C \) may also denote its node set if it is clear from the context. Let \( B \subseteq C \) be a node set whose elements are said to be masked.

Right now, the gap between Proposition 5.1 and Theorem 6.1 is the depth of \( C_t(\gamma) \): It is unbounded for the theorem so that we cannot guarantee the existence of a shift \( t(y) \) constructed with regular \( \sigma \) only. If there are possibly non-regular \( \sigma \in \mathcal{Q}(\gamma) \), the statement \( d(f(\sigma)) \cap z = \emptyset \) in Corollary 5.4 is not necessarily true.

Here are our observations to fill the gap.

(a) There exists a node \( \alpha \) in \( C_t(\gamma) \) such that \( \frac{1}{2} |C_t(\gamma)| < \frac{1}{3} |C_t(\gamma)| - \frac{1}{2} \leq |\alpha(C_t(\gamma))| \leq \frac{2}{3} |C_t(\gamma)| \).

Started at the root of \( C_t(\gamma) \), always traverse its larger sub-circuit only, which is rooted at a child of the current node with tie broken arbitrarily. Stop the traversal when the current sub-circuit has size \( \frac{2}{3} |C_t(\gamma)| \) or less for the first time. In the next to the last step, there are at least \( \frac{2}{3} |C_t(\gamma)| \) nodes. Thus there are \( \left( \frac{2}{3} |C_t(\gamma)| - 1 \right) / 2 \geq \frac{1}{3} |C_t(\gamma)| - \frac{1}{2} \) nodes in \( \alpha(C_t(\gamma)) \). It is greater than \( \frac{1}{4} |C_t(\gamma)| \) if \( C_t(\gamma) \) has a sufficiently large size, say at least 10.

(b) View \( \alpha(C_t(\gamma)) \) and \( C_t(\gamma) \setminus \alpha(C_t(\gamma)) \) as two disjoint sub-circuits of \( C_t(\gamma) \).

This is a recursive step to “flatten \( C_t(\gamma) \) substantially”. The term \( t(\sigma) \) at every node in \( \alpha(C_t(\gamma)) \) has been constructed flat so far, i.e., everywhere in \( t(\sigma) \) uses only regular quadruples with bounded ranks inductively. Our construction will perform on \( \alpha(C_t(\gamma)) \) first. If there is any node \( \alpha' \) in \( C_t(\gamma) \setminus \alpha(C_t(\gamma)) \) whose child \( \alpha'' \) is in \( \alpha(C_t(\gamma)) \), LOCALSHIFT does NOT join the term \( t(\sigma) \) at \( \alpha'' \) with \( t(\sigma) \) at \( \alpha' \); since another term at \( \alpha'' \) has been already included in \( \alpha(C_t(\gamma)) \). Such \( \alpha'' \) is masked for \( \alpha' \).

This way we only process nodes in \( C_t(\gamma) \) that do not belong to \( \alpha(C_t(\gamma)) \), achieving the disjointness.

(c) Recursively flatten the two sub-circuits substantially. It continues for only \( O(\log n) \) depth in the recursion tree.

Because of (a), each of \( \alpha(C_t(\gamma)) \) and \( C_t(\gamma) \setminus \alpha(C_t(\gamma)) \) has size at most \( 3/4 \) of \( |C_t(\gamma)| \). In the subsequent recursive steps, the current sub-circuits where masked nodes are excluded reduce their sizes exponentially.

(d) As a result, the constructed \( t(y) \) is flat.

We will show that the quadruples are regular everywhere in \( t(y) \) so that it is contradiction-free.

Based on the above idea, run Algorithm FLATTEN in Fig. 11. It is a recursive algorithm to perform on a sub-circuit \( C' \subseteq C \) with a recursion level \( i \), which returns a reconstructed circuit \( C'' \) to compute the same Boolean function as \( C' \) (Fig. 12). We have the following lemma on it.

LEMMA 6.2. Assume that a polynomial-sized circuit \( C \) computes \( k\text{-CLIQUE} \). The Boolean circuit \( C = \text{FLATTEN}(C, \emptyset, \ln^2 n) \) satisfies the following two:
Algorithm Flatten($C', B, i$)

Inputs:
1. Sub-circuit $C' \subseteq C$,
2. set $B$ of masked nodes in $C'$, and
3. integer $i \in \lceil \ln^2 n \rceil$.

Output: A Boolean circuit $C''$ such that $r(C'') \Leftrightarrow r(C')$.

begin
1. if $i = 1$ or $|C' \setminus B| \leq 10$ then return $C'$ and exit the algorithm else go to Step 2;
2. for each node $\alpha \in C' \setminus B$ do
   2-1. $C_{\alpha,1} \leftarrow$ Flatten($\alpha(C'), B \cap \alpha(C), i - 1$);
   2-2. $C_{\alpha,2} \leftarrow$ Flatten($C', B \cup \alpha(C'), i - 1$);
3. end for
4. return the circuit $C''$ whose root is $r(C'') = \bigvee_{\alpha \in C'} C_{\alpha,1} \land C_{\alpha,2}$;
end

Fig. 11. Reconstruction of a Circuit $C' \subseteq C$

\[\begin{array}{c}
\text{Flatten} \\
C' \\
\end{array}\]

\[\begin{array}{c}
\text{Flatten} \\
C'' \\
\end{array}\]

Fig. 12. How Flatten Works

(i) $\hat{C}$ computes $k$-CLIQUE in such a way that $\text{DNF} (r(C)) \subseteq \text{DNF} \left( r \left( \hat{C} \right) \right)$.

(ii) $|\hat{C}| = e^{O(ln^3 n)}$.

PROOF. In a recursive call occurring in Flatten($C, \emptyset, \ln^2 n$), any first parameter $C'$ of Flatten is a non-empty subgraph of $C$ due to Steps 1, 2-1 and 2-2. Show the two statements below noting them.

(i): $\hat{C}$ computes $k$-CLIQUE since

\[ r(C'') \Leftrightarrow \bigvee_{\alpha \in C'} r(C_{\alpha,1}) \land r(C_{\alpha,2}) \Leftrightarrow \bigvee_{\alpha \in C'} \alpha(C') \land r(C') \Leftrightarrow r(C'), \]

for any circuit $C'$ input to Flatten and returned $C''$. Here $r(C_{\alpha,1}) \Leftrightarrow \alpha(C')$ and $r(C_{\alpha,2}) \Leftrightarrow C'$ are assumed inductively on $i$.

To see $\text{DNF} \left( r(C'') \right) \supseteq \text{DNF} \left( r \left( C' \right) \right)$, let $t$ be any term in $\text{DNF} \left( r \left( C' \right) \right)$. There exists a node $\alpha \in C_i$ since $C_i \neq \emptyset$. The family $\text{DNF} \left( r \left( C_{\alpha,1} \right) \right)$ contains the term $t''$ at $\alpha$ to generate $t$ in $C$, assumed inductively on $i$. Also $\text{DNF} \left( r \left( C_{\alpha,2} \right) \right)$ contains $t$ inductively. Thus $t \in \text{DNF} \left( r(C'') \right)$ in Step 4.
Algorithm FlattenTerm(t, C', B, i)
Inputs:
1. Term t at r(C') in the sub-circuit C' ⊆ C,
2. set B of nodes masked in C', and
3. integer i ∈ [ln^2 n].
Output: A derivation graph of t in Flatten(α(C'), B, i).

begin
1. if i = 1 or |Ct,B| ≤ 10 then return the derivation graph of t at r(C') in C, and exit the algorithm
   else go to Step 2;
   /* Ct,B is the derivation graph of t ∈ DNF(r(C')) in C, from which the nodes in B and incident edges are removed. So |Ct,B| ≤ |C' \ B|. */
2. Find a node α ∈ Ct,B such that both α (Ct,B) and Ct,B \ α (Ct,B) have sizes at most \( \frac{3}{2} |Ct,B| \);
   /* As in the observation (a). Here α (Ct,B) is the subgraph of Ct,B rooted at α containing all its descendants. */
3. t ← the term at α to generate t in C';
4. Cα,1,t ← FlattenTerm(t', α (C'), B ∩ α (C'), i − 1);
5. Cα,2,t ← FlattenTerm(t, C', B ∪ α (C'), i − 1);
   /* Synchronized with Steps 2-1 and 2-2 of Flatten. */
6. return the subgraph of C' at Step 4 of Flatten, such that Cα,1 ∩ Cα,2 ← Cα,1,t ∩ Cα,2,t for the α found by the above 1, and Cα,1,t ∩ Cα,2,t ← \( \emptyset \) for every other α ∈ C';
end

Fig. 13. For a Derivation Graph of a Given Term t with Small Flattened Depth

(ii): It suffices to show that there exists a constant \( \gamma_1 > 0 \) such that \( |Flatten(C', B, i)| \leq e^{\gamma_1 \ln n} \) for every sub-circuit \( C' \subseteq C \) and \( i \in [ln^2 n] \). Prove it by induction on \( i \) with an obvious basis \( i = 1 \). The returned circuit \( C'' \) consists of \( C_{α,1}, C_{α,2} \) and extra conjunctions and disjunctions to construct \( r(C'') \) in Steps 4. By induction hypothesis, its total number of nodes is bounded by

\[
|Flatten(C_1, B', i - 1)| + |Flatten(C_2, B', i - 1)| + O(|C|) \\
\leq 2e^{(i-1)\gamma_1 \ln n} + O(|C|) < e^{i\gamma_1 \ln n},
\]

for a sufficiently large choice of \( \gamma_1 \). □

In Fig. 13 Algorithm FlattenTerm(t, C', B, i) constructs a derivation graph of the term t at r(C'), C' ⊆ C in the sub-circuit Flatten(C', B, i) of \( \hat{C} \).

Now here is our process to construct a shift t(y) for the general circuit C of polynomial size to compute k-CLIQUE.

Step A. Construct the circuit \( \hat{C} = Flatten(C, \emptyset, ln^2 n) \). Find a derivation graph of \( t_0(c, z) \) in \( \hat{C} \) for every k-clique c and negative tail z ∈ \( \binom{n}{n^1/4} \) with FlattenTerm.

Due to Lemma 6.2 (i), DNF \( r(\hat{C}) \) contains the global terms of C including every \( t_0(c, z) \) (defined in Section 5.1). Run FlattenTerm \( (t_0(c, z), C, \emptyset, ln^2 n) \) to find a derivation graph of \( t_0(c, z) \) in \( \hat{C} \).

If \( \hat{C}_t = FlattenTerm(t, C', B, i) \) is a recursive call occurring in FlattenTerm \( (t_0(c, z), C, \emptyset, ln^2 n) \) such that \( α = r(\hat{C}_t) \), then we write

\[
(c, z, α) \triangleright (t, C', B, i).
\]

Notice that \( r(\hat{C}_t) = r(C'') \) where \( C'' = Flatten(C', B, i) \).
The relation \((34)\) determines the following two objects.

(a) \(C_{t,B}: \) As in Step 1 of FLATTENTERM, it is the derivation graph of the term \(t\) at \(r(C)\) in \(C\), from which the nodes in \(B\) and incident edges are removed.

(b) \(t_B: \) Let \(t'\) be the term at a node \(\alpha \in C_{t,B} \cap B\) to generate \(t\) in \(C\). Construct the subset \(t_B\) of \(t\) by replacing every such \(t'\) by \(\emptyset\). Call \(t_B\) a \(B\)-term of \(t\) at \(r(C')\) in \(C\), and also \(a\) at \(r(C'')\) in \(C\). Notice \(t \in DNF(r(C''))\) by the proof of Lemma 6.2(i).

Observe the following lemma.

**Lemma 6.3.** Suppose that \((34)\) holds. Then:

(i) The first parameter \(t\) of the recursive call FLATTEN\((t,C',B,i)\) is a term at \(r(C')\) to generate \(t_0(c,z)\) in \(C\).

(ii) \(|C_{t,B}| \leq |C| \left(\frac{1}{4}\right)^{\ln^2 n - i}\).

**Proof.** (i): Due to Steps 3 to 5 of FLATTENTERM, it is true inductively on \(i = \ln^2 n, \ln^2 n - 1, \ldots, 1\).

(ii): Show \[\log_{4/3} |C_{t,B}| \leq \log_{4/3} |C| + i - \ln^2 n\] by induction on \(i = \ln^2 n, \ln^2 n - 1, \ldots, 1\). The basis \(i = \ln^2 n\) is clearly true due to \(|C_{t,B}| \leq |C|\).

Assume true for the current recursive call of FLATTENTERM such that \((34)\). Due to the choice of \(\alpha\) by Step 2, the component sizes in the next recursive calls at Steps 4 and 5 are both bounded properly, i.e., \(\max\left\{\alpha\left(C_{t,B}\right), |C_{t,B} \setminus \alpha\left(C_{t,B}\right)|\right\} \leq \frac{3}{4}|C_{t,B}|\).

By induction hypothesis, the logarithm of its LHS with respect to base \(4/3\) is at most \[\ln \frac{3}{4} |C_{t,B}| = \ln \frac{3}{4} |C_{t,B}| - 1 \leq \log_{4/3} |C| + (i - 1) - \ln^2 n.\]

This proves the induction step. \(\square\)

**Step B. Run CLIQUEGENERATORS to construct \(C_\alpha(z), Y(z), Q(z)\) and \(\hat{Q}_0\).**

For each fixed negative tail \(z \in \binom{[n]}{n/4}\), run CLIQUEGENERATORS on \(k\)-cliques \(c\) generated at each \(\alpha\). Here it is defined that \(c\) is generated at \(\alpha\) with recursion level \(i \in [\ln^2 n]\), if \((c,z,\alpha) \triangleright (t,C',B,i)\) for some \(C' \subseteq C\), \(t \in DNF(r(C'))\), and \(B \subseteq C'\) such that \(\alpha = r(\text{FLATTEN}(C',B,i))\).

A difference from the previous cases is that the algorithm CLIQUEGENERATORS collects only \(k\)-cliques \(c\) at \(\alpha\) with a same recursion level \(i\). For simple descriptions, regard that cliques with distinct recursion levels are generated at distinct \(\alpha\). This increases the number of nodes by a factor no more than \(\ln^2 n\), which is still bounded by a polynomial in \(n\). The recursion level \(i\) is related to \(|C_{t,B}|\) by Lemma 6.3(ii). It reduces exponentially as \(i = \ln^2 n, \ln^2 n - 1, \ldots, 1\).

As in Section 5.1, construct \(C_\alpha(z), Y(z), Q(z)\) for each \(z\) and \(\hat{Q}_0 = \bigcup \hat{Q}(z)\). A quadruple \(\sigma = (g, g_i, g_2, \alpha) \in Q(z)\) is incident to a \(k\)-clique \(c\) at \(\alpha \in \hat{C}\) if the following four conditions are satisfied:

I. \(c\) is generated at \(\alpha\) with recursion level \(i\) so that \((c,z,\alpha) \triangleright (t,C',B,i)\) for some \((t,C',B,i)\).

II. \(g \subset c\) is a generator found by CLIQUEGENERATORS at \(\alpha\) with \(i\).
III. If \(|C_{t,B}| > 10\), then each \(g_k, k = 1, 2\) is a generator found by CLIQUEGENERATORS at \(r(C_{a,k,t})\) where \(C_{a,k,t}\) is the derivation graph constructed by Step 4 or 5 of FLATTENTERM \((t, C', B, i)\).

IV. If \(|C_{t,B}| \leq 10\), then \(g_1 = g_2 = \text{vertex } (t_B \cap \binom{[n]}{2})\).

(As in (33), \(\cap \binom{[n]}{2}\) filters negative literals. So \(t_B \cap \binom{[n]}{2}\) is the positive literal set of the \(B\)-term \(t_B\).

Step C. Perform Steps 1–4 in the proof of Proposition 5.1 on \(\bar{C}\).

This uniquely determines a negative tail \(z\) such that the two statements in Corollary 5.4 hold true.

Step D. Perform Steps 5 to construct \(t(y)\) with a variant of LOCALSHIFT.

LOCALSHIFT2 is a variant of LOCALSHIFT defined in the recursion tree of FLATTEN \((C, \emptyset, \ln^2 n)\). It returns a term \(t(\sigma)\) for every \(\sigma \in \mathcal{Q}_0\) incident to a \(k\)-clique \(c\) generated at a node \(\alpha \in \mathcal{C}\) as follows:

1. Let \(\sigma^* = f(\sigma)\) as Step 2-1 of LOCALSHIFT does. It is incident to the \(k\)-clique \(c^*\) generated at \(\alpha\) such that \(g \subset c^* \subset g \cup y(\sigma)\).

2. By our construction in Step B, \(c^*\) at \(\alpha\) has the same recursion level \(i\) as \(c\). There exists a term \(t\) at \(\alpha\) such that

\[
(c^*, z, \alpha) \triangleright (t, C', B, i)
\]

for some \((t, C', B, i)\) such that \(\alpha = r(\text{FLATTENTERM}(t, C', B, i))\).

3. Let \(\alpha' \in C_{t,B}\) be the node found by Step 2 of FLATTENTERM \((t, C', B, i)\). Let \(\alpha_k, k = 1, 2\) be the roots of \(C_{a,k,t}\) defined by its steps 4 and 5.

4. LOCALSHIFT2 recursively constructs \(t(\sigma_k)\) for \(\sigma_k\) incident to \(c^*\) at \(\alpha_k\). It returns \(t(\sigma) = t(\sigma_1) \cup t(\sigma_2)\) at \(\alpha\). By Lemma 6.3(ii), the current component size in the next step (i.e., \(|C_{t,B}|\) of the next step) is reduced by factor at most \(3/4\) for \(\sigma_k\).

5. The above is for an inductive step \(|C_{t,B}| > 10\). For a basis \(|C_{t,B}| \leq 10\), find the same \(c^* \subset g \cup y(\sigma)\) and \(t\) as Step 2. Return the \(B\)-term \(t_B\) of \(t\) as \(t(\sigma)\). Recall that by our definition in Step B, there exists \(\sigma = (g, g_1, g_2, \alpha)\) incident to \(c^*\) such that \(g_1 = g_2 = \text{vertex } (t_B \cap \binom{[n]}{2})\).

This completes the construction of \(t(\sigma)\). As before, the shift \(t(y)\) is \(t(\sigma)\) for \(\sigma\) chosen by Step 4 of SHIFT. Let \(\mathcal{Q}_{i(\sigma)}(i)\) be the family of \(\sigma\) occurring in Step 1 of LOCALSHIFT2 with recursion level \(i\), in the construction of \(t(y)\).

**Lemma 6.4.** The following two statements hold true on \(t(y)\).

(i) \(t(y)\) is a global term of the circuit \(\bar{C} = \text{FLATTEN}(C, \emptyset, \ln^2 n)\).

(ii) Any \(\sigma \in \mathcal{Q}_{i(\sigma)}(i)\) is regular.

**Proof.** (i): We claim that for every \(\sigma \in \mathcal{Q}_0\) incident to a clique generated at \(\alpha \in \mathcal{C}\), the returned term \(t(\sigma)\) is a \(B\)-term at \(\alpha\). Show it by induction on \(i = 1, 2, \ldots, \ln^2 n\). A basis occurs when \(|C_{t,B}| \leq 10\), due to Lemma 6.3(ii). It is clearly true since LOCALSHIFT2 returns a \(B\)-term at \(\alpha\). Assume true for \(i - 1\) and prove true for \(i\).

Let \((c^*, z, \alpha) \triangleright (t, C', B, i)\) in Step 2 of LOCALSHIFT2, and \(\alpha_k = r(C_{a,k,t})\) for \(k = 1, 2\) as in Step 3. By induction hypothesis, the terms constructed by Step 4 are:

- \(B_1\)-term \(t(\sigma_1)\) at \(\alpha_1\) where \(B_1 = B \cap \alpha(C')\), and
- \(B_2\)-term \(t(\sigma_2)\) at \(\alpha_2\) where \(B_2 = B \cup \alpha(C')\).
Now observe that after the recursive call of LOCALSHIFT2 for \( t(\sigma_1) \) is completed, a \( B_1 \)-term at every node in \( \alpha(C') \setminus B \) is constructed in \( C \) as illustrated in Fig. 14. The nodes masked in \( B_2 \setminus B_1 \) for \( t(\sigma_2) \) are no longer masked for \( t(\sigma) \) as discussed in (b).

Therefore, LOCALSHIFT2 returns \( t(\sigma) \) that is a \( B \)-term at \( r(C') \) in \( C \). It is a \( B \)-term at \( \alpha = r(\text{FLATTEN}(C', B, i)) \), proving the induction step.

(ii): The statement is the same as in Lemma 5.5 and is shown by similar logic: with the depth of \( \alpha \) replaced by \( \ln^2 \frac{n}{i} \). More precisely, prove by induction on \( i = \ln^2 \frac{n}{i}, \ln^2 \frac{n}{i} - 1, \ldots, 1 \) that every \( \sigma \in Q_{t(y)}(i) \) has rank \( q' \left( \ln^2 n - i \right) \) or less. If it is true, \( \sigma \) is regular by Lemma 5.4 (ii).

The basis \( i = \ln^2 \frac{n}{i} \) of the induction is clearly true since \( \sigma \) chosen by Step 4 of SHIFT has rank 0 with empty \( g \). Assume true for \( i \) and prove true for \( i - 1 \). Step 1 of LOCALSHIFT2 finds the \( k \)-clique \( c^* \) contained in \( g \cup y(\sigma) \). By induction hypothesis, the rank of \( c^* \) is no more than \( \text{rank}(g) + q' \left( \text{rank}(\sigma) + 1 \right) \leq q' \left( \ln^2 n - i + 1 \right) \). The first component of \( \sigma_2 \) chosen by Step 4 is contained in \( c^* \); thus \( \text{rank}(\sigma_2) \leq q' \left( \text{rank}(\sigma) + 1 \right) \leq q' \left( \ln^2 n - i + 1 \right) \). This proves the induction step for recursion level \( i - 1 \). □

**Step E. Show that \( t(y) \) is free from a contradiction and \( k \)-clique.**

\( t(y) \cap z = \emptyset \) is proven by the same logic as in Proposition 5.1. Define \( Q_{t(y)} = \bigcup_{i \in [\ln^2 n]} Q_{t(y)}(i) \) instead of (31). Corollary 5.4 with Lemma 6.4 (ii) implies \( \Phi \cap z = \emptyset \) due to the regularity of related \( \sigma \). We have (33): \( \left( \left\lfloor \frac{n}{2} \right\rfloor \right) \cap t(\sigma) \setminus \left( \left\lfloor \frac{\ell}{2} \right\rfloor \right) \subseteq \Phi \) as for the \( \text{NP} \neq \text{NC} \) case. These mean that \( t(\sigma) \cap z = \emptyset \) for every \( \sigma \in Q_{t(y)} \). Consequently, \( t(y) \cap z = \emptyset \) holds true. Notice the following: When run BLOCKED EDGES3, it is necessary to have \( |d(\sigma)| \leq n^2 \) to show Lemma 5.3 (b). In a base case \( |C_{t,B}| \leq 10 \), the size of \( d(\sigma) \) is bounded by Condition IV of Step B.

Due to \( t(y) \cap z = \emptyset \), the global term \( t(y) \) of \( \hat{C} \) is a shift containing no contradiction and \( k \)-clique. This is against the assumption that \( C \) computes \( k \)-CLIQUE, proving the impossibility of such \( C \).
It is straightforward to generalize the proof to any circuit of size at most \(e^{n^2}\) instead of just a polynomial size: The size of \(\hat{C}\) becomes \(e^{O(n^{3^2})}\) instead of \(e^{O(n^{3^2})}\), and the maximum recursion level \(n^{2^2}\) instead of \(n^{3^2}\).

The proof of Theorem 6.1 is now complete. We have:

**COROLLARY 6.5.** \(P \neq \text{NP} \). □

7. CONCLUSIONS

We have presented a proof of \(P \neq \text{NP} \) through the exponential circuit complexity of \(\text{CLIQUE}\). Some topological properties of the Hamming space have been explored, with the emphasis on the \(l\)-extension of a family of \(m\)-sets. The extension generator theorem is used to split \(|n|\) into disjoint spaces \(y_1, y_2, \ldots, y_q\) of equal size. A sunflower with a small core is constructed by the theorem, showing its relationship to extension generators.

With the developed theory, we first posed a new proof of the exponential monotone circuit complexity of \(\text{LIQUE}\). Focusing on only \(\sigma\) with the regularity constraint, we generalized the construction so that it handles a non-monotone circuit with bounded depth, leading to \(\text{NP} \supseteq \text{NC} \). Finally, we devised a way to substantially flatten the targeted circuit to show \(P \neq \text{NP} \). It has been confirmed that the proof is non-natural.

**Acknowledgement:** The author would like to thank Professors Martin F¨urer at Penn State, Osamu Watanabe at Tokyo Institute of Technology, and Eric Allender at Rutgers University for their good suggestions. Also, it has been very helpful of Professors Tadao Saito at the University of Tokyo and Hisashi Kobayashi at Princeton University to provide supports toward the submission of this paper.

**Appendix A: Proofs of Claims in Section 2.3**

**Theorem 2.2**

\[
\left| \ln \left( \frac{p}{q} \right) - q \left( \ln \frac{p}{q} + 1 - S \left( \frac{q}{p} \right) \right) - \frac{1}{2} \ln \frac{p}{2\pi q(p-q)} \right| = O \left( \frac{1}{\min(q,p-q)} \right),
\]

for \(p, q \in \mathbb{Z}\) such that \(0 < q < p\).

**PROOF.** By Stirling’s series, \(k! = \sqrt{2\pi k} \left( \frac{k}{e} \right)^k \left( 1 + O \left( \frac{1}{k} \right) \right) \) for \(k \in \mathbb{Z}_{>0}\). It implies

\[
\binom{p}{q} = \frac{p!}{q!(p-q)!} = \frac{\sqrt{2\pi p} \left( \frac{p}{e} \right)^p \left( 1 + O \left( \frac{1}{p} \right) \right)}{\sqrt{2\pi q} \left( \frac{q}{e} \right)^q \left( 1 + O \left( \frac{1}{q} \right) \right) \cdot \sqrt{2\pi (p-q)} \left( \frac{p-q}{e} \right)^{p-q} \left( 1 + O \left( \frac{1}{p-q} \right) \right)}
\]

\[
= \frac{p^p}{2\pi q(p-q)} \cdot \frac{q^q}{q^q} \cdot \frac{(p-q)^{p-q}}{(p-q)^{p-q}} \cdot r,
\]

where \(r = \frac{1+O(p^{-1})}{(1+O(q^{-1})) \cdot (1+O(p-q)^{-1})} \) is a real number such that \(|r-1| = O \left( \frac{1}{\min(q,p-q)} \right) \).

Thus it suffices to show

\[
\ln \frac{p^p}{q^q(p-q)^{p-q}} = q \left( \ln \frac{p}{q} + 1 - S \left( \frac{q}{p} \right) \right).
\]

Its left hand side is equal to

\[
p \ln p - q \ln q - (p-q) \left( \ln p + \ln \left( 1 - \frac{q}{p} \right) \right) = q \ln \frac{p}{q} - p \left( 1 - \frac{q}{p} \right) \ln \left( 1 - \frac{q}{p} \right).
\]
Computing Cliques Is Intractable

We have

\[ -(1 - x) \ln (1 - x) = (1 - x) \sum_{j \geq 1} \frac{x^j}{j} = x + \sum_{j \geq 2} \frac{x^j}{j} - \sum_{j \geq 1} \frac{x^{j+1}}{j} \]

\[ = x + \sum_{j \geq 1} \left( \frac{x^{j+1}}{j+1} - \frac{x^{j+1}}{j} \right) = x - x \cdot S(x), \]

for \( x \in (0, 1) \). Therefore,

\[ \ln \frac{p^p}{q^q(p-q)^{p-q}} = q \ln \frac{p}{q} - \frac{q}{p} \ln (1 - \frac{q}{p}) = q \ln \frac{p}{q} + p \left( \frac{q}{p} - \frac{q}{p} \cdot S \left( \frac{q}{p} \right) \right) \]

\[ = q \left( \ln \frac{p}{q} + 1 - S \left( \frac{q}{p} \right) \right), \]

proving the theorem. □

**Lemma 2.4**

\[ \binom{n-m}{l} = \binom{n}{l} e^{-\frac{lm}{n} - o \left( \frac{lm}{n} \right)}, \]

for \( m, l \in [n] \) such that \( l + m \ll n \).

**Proof.** First we see \( \ln \binom{n-m}{l} < \ln \binom{n}{l} - \frac{lm}{n} \). Its proof is found in [4] as follows:

\[ \ln \binom{n-m}{l} = \sum_{i=0}^{l-1} \frac{n-m-i}{n-i} = \prod_{i=0}^{l-1} \left( 1 - \frac{m}{n-i} \right) < \left( 1 - \frac{m}{n} \right)^l, \]

\[ \Rightarrow \ln \binom{n-m}{l} - \ln \binom{n}{l} \leq \ln \left( 1 - \frac{m}{n} \right) \]

\[ = -l \sum_{i \geq 1} \frac{1}{i} \left( \frac{m}{n} \right)^i < -\frac{lm}{n}. \]

The lower bound \( \ln \binom{n-m}{l} \geq \ln \binom{n}{l} - \frac{lm}{n} - o \left( \frac{lm}{n} \right) \) is shown similarly:

\[ \frac{\binom{n-m}{l}}{\binom{n}{l}} \geq \left( 1 - \frac{m}{n-l} \right)^l, \]

\[ \Rightarrow \ln \binom{n-m}{l} - \ln \binom{n}{l} \geq l \ln \left( 1 - \frac{m}{n-l} \right) = -l \sum_{i \geq 1} \frac{1}{i} \left( \frac{m}{n-l} \right)^i. \]

We have \( n \gg \max (l, m) \) and \( l \sum_{i \geq 1} \frac{1}{i} \left( \frac{m}{n} \right)^i = l \left( \frac{m}{n} + \frac{1}{2} \left( \frac{m}{n} \right)^2 + \cdots \right) = \frac{ml}{n} + o \left( \frac{ml}{n} \right) \). Thus,

\[ \ln \binom{n-m}{l} - \ln \binom{n}{l} \geq -l \sum_{i \geq 1} \frac{1}{i} \left( \frac{m}{n-l} \right)^i = -\frac{ml}{n} - o \left( \frac{ml}{n} \right), \]

proving the lemma. □

**Lemma 2.5**

\[ \ln \binom{l-m}{m-j} \left( \frac{m}{j} \right) \leq \ln \binom{l}{m} - j \ln \frac{j}{m} + j + \ln j + O(1), \]

for \( l, m \in [n] \) such that \( m^2 \leq l \) and \( j \in [m] \).
PROOF. Assume \( j < m \). (The case \( j = m \) is proven in the footnote below\[15\]) Note that
\[
\ln \left( \frac{l - m}{m - j} \right) = \ln \left( \frac{l}{m - j} \right) - O \left( \frac{m^2}{l} \right) = \ln \left( \frac{l}{m - j} \right) - O(1),
\]
due to Lemma 2.4 and \( l \geq m^2 \).

Approximate \( \ln \left( \frac{l - m}{m - j} \right) = \ln \left( \frac{l}{m - j} \right) - O(1) \) and \( \ln \left( \frac{m}{j} \right) \) by Theorem 2.2 as follows: For the former,
\[
\ln \left( \frac{l - m}{m - j} \right) = X_1 + Y_1 + O(1),
\]
where
\[
X_1 = (m - j) \left( \ln \frac{l}{m - j} + 1 - S \left( \frac{m - j}{l} \right) \right),
\]
and
\[
Y_1 = \frac{1}{2} \ln \frac{l}{(m - j)(l - m + j)}.
\]

For the latter,
\[
\ln \left( \frac{m}{j} \right) = X_2 + Y_2 + O(1),
\]
where
\[
X_2 = j \left( \ln \frac{m}{j} + 1 - S \left( \frac{j}{m} \right) \right),
\]
and
\[
Y_2 = \frac{1}{2} \ln \frac{m}{j(m - j)}.
\]

Then it suffices to show:
\[
X_1 + X_2 \leq X_3 - j \ln \frac{j l}{m^2} + j + o(1), \quad (35)
\]
and
\[
Y_1 + Y_2 \leq Y_3 + \ln j + O(1), \quad (36)
\]
where
\[
X_3 = m \left( \ln \frac{l}{m} + 1 - S \left( \frac{m}{l} \right) \right),
\]
and
\[
Y_3 = \frac{1}{2} \ln \frac{l}{m(l - m)}.
\]

First show (35). We have
\[
S \left( \frac{m}{l} \right) = \sum_{k \geq 1} \frac{\left( \frac{m}{l} \right)^k}{k(k + 1)} = O \left( \frac{m}{l} \right),
\]
\[
\Rightarrow \quad S \left( \frac{m - j}{l} \right) = S \left( \frac{m}{l} \right) - O \left( \frac{m}{l} \right).
\]
by the definition (4) of the function \( S \), \( j < m \) and \( m^2 \ll l \). In addition,
\[
\ln \frac{l}{m - j} = \ln \frac{l}{m} - \ln \left( 1 - \frac{j}{m} \right) = \ln \frac{l}{m} + \sum_{k \geq 1} \frac{1}{k} \left( \frac{j}{m} \right)^k,
\]
\[\text{[15]}\]When \( j = m \), LHS of the desired inequality is zero. Its RHS is
\[
R = \ln \left( \frac{l}{m} \right) - m \ln \frac{l}{m} + m + \ln m + \gamma.
\]
where \( \gamma \) is a sufficiently large constant we may choose. Theorem 2.2 means \( R = 2m + o(m) + \gamma \). It is positive when \( m \gg 1 \). If \( m = O(1) \), it is also positive with a choice of \( \gamma \) since \([2m + o(m)] = O(1)\).
by the Taylor series of natural logarithm. Thus,
\[ X_1 = (m - j) \left( \ln \frac{l}{m - j} + 1 - S \left( \frac{m - j}{l} \right) \right) \]
\[ = (m - j) \left( \ln \frac{l}{m} + \sum_{k \geq 1} \frac{1}{k} \left( \frac{j}{m} \right)^k + 1 - S \left( \frac{m}{l} \right) + O \left( \frac{m}{T} \right) \right) \]
\[ = m \left( \ln \frac{m}{T} + 1 - S \left( \frac{m}{T} \right) + O \left( \frac{m}{T} \right) \right) - j \left( \ln \frac{l}{m} + 1 - S \left( \frac{m}{T} \right) + O \left( \frac{m}{T} \right) \right) + (m - j) \sum_{k \geq 1} \frac{1}{k} \left( \frac{j}{m} \right)^k \]
\[ \leq X_3 - j \left( \ln \frac{l}{m} + 1 \right) + (m - j) \sum_{k \geq 1} \frac{1}{k} \left( \frac{j}{m} \right)^k + O \left( \frac{m^2}{T} \right) \]
\[ \quad \text{where} \quad X_3 = m \left( \ln \frac{l}{m} + 1 - S \left( \frac{m}{T} \right) \right) \quad \text{and} \quad S \left( \frac{m}{T} \right) = O \left( \frac{m}{T} \right) . \]

Observe that
\[ (m - j) \sum_{k \geq 1} \frac{1}{k} \left( \frac{j}{m} \right)^k = m \cdot \left( 1 - \frac{1}{\frac{j}{m}} \right) + m \sum_{k \geq 2} \frac{1}{k} \left( \frac{j}{m} \right)^k - j \sum_{k \geq 1} \frac{1}{k} \left( \frac{j}{m} \right)^k \]
\[ = j + \sum_{k \geq 2} \frac{j^k}{km^{k-1}} - \sum_{k \geq 1} \frac{j^{k+1}}{km^k} \]
\[ = j - \sum_{k \geq 1} \left( \frac{1}{k} - \frac{1}{k+1} \right) \frac{j^{k+1}}{m^k} < j. \]

By the above two,
\[ X_1 < X_3 - j \ln \frac{l}{m} + O \left( \frac{m^2}{T} \right) = X_3 - j \ln \frac{l}{m} + o(1). \]

On the other hand,
\[ X_2 = j \left( \ln \frac{m}{j} + 1 - S \left( \frac{j}{m} \right) \right) < j \ln \frac{m}{j} + j. \]

Therefore,
\[ X_1 + X_2 < X_3 - j \ln \frac{l}{m} + j \ln \frac{m}{j} + j + o(1) = X_3 - j \ln \frac{j l}{m^2} + j + o(1), \]
proving (35).

To show (36), see that
\[ Y_1 + Y_2 - Y_3 = \frac{1}{2} \ln \left( \frac{m}{m-j}(l-m+j) \right) + \frac{1}{2} \ln \frac{l}{j(m-j)} - \frac{1}{2} \ln \frac{l}{m(l-m)} \]
\[ = \ln \frac{m}{m-j} - \frac{1}{2} \ln j + \frac{1}{2} \ln \frac{l}{m-j} + \frac{1}{2} \ln \frac{l-m}{l-m+j} \]
\[ < - \ln \left( 1 - \frac{j}{m} \right) - \frac{1}{2} \ln j. \]

If \( j \leq m/2 \), the above is less than \(-\ln \left( 1 - \frac{2m}{l} \right) \leq \ln 2 = O(1) \). Also, if \( m \) is bounded by a constant, the maximum value of \(-\ln \left( 1 - \frac{j}{m} \right) \) is \(-\ln \left( 1 - \frac{m-1}{m} \right) \) which equals \( \ln m = O(1) \), since \( j \leq m - 1 \).

The remaining case occurs when both \( j > m/2 \) and \( m \gg 1 \) are true. They mean
\[ Y_1 + Y_2 - Y_3 - \ln j < \ln m - \frac{3}{2} \ln j < \ln m - \ln \left( \frac{m}{T} \right)^{3/2} < 0. \]
Therefore, the complement sparsity of the tuples \( (s_1, s_2, \ldots, s_{j-1}, s_j') \) constructed by the \( j^{th} \) process is at least \( \lambda_j \) if \( \gamma > 0 \), so that
\[
|s_j| - \frac{|X_j|m}{n} < 1 \quad \text{and} \quad |s_j'| - \frac{|M_j|m}{n} < 1.
\] (37)

The basis \( j = 0 \) occurs before the first step \( j = 1 \). The claim holds true since there is \( s'_0 \in U \) of size \( \frac{M_{0|m}}{n} = m \) whose complement sparsity is \( \lambda_0 = \kappa(U) \). Assume true for \( j - 1 \) and prove true for \( j \).

Before the \( j^{th} \) step, there are \( (s_1, s_2, \ldots, s_{j-1}, s_j') \) that form a majority with complement sparsity \( \lambda_{j-1} \). For every such \( j^{th} \) step, we split \( s'_{j-1} \) into \( (s_j, s_j') \) space-proportionally to \( X' \), creating a \( j^{th} \) tuple \( (s_1, s_2, \ldots, s_j, s_j') \). There is a bijection between \( j^{th} \) and \( \gamma j \) tuples; one \( \gamma j \) creates exactly one \( \gamma j \) and no other does.

By induction hypothesis and \( \gamma j \), the number of \( (s_1, s_2, \ldots, s_j, s_j') \) NOT created by the \( j^{th} \) step is at most
\[
\prod_{i=1}^{j-1} \left( \frac{|X_i|}{|s_i|} \right) \left( \frac{|M_{i-1}|}{|s_{i-1}|} \right) e^{-\gamma(j-1)} n
\]
\[
= \prod_{i=1}^{j-1} \left( \frac{|X_i|}{|s_i|} \right) \left( \frac{|M_{i}|}{|s_{i}'|} \right) e^{-\gamma(j-1)} n
\]
\[
= \sum_{i=1}^{j} \left( \frac{|X_i|}{|s_i|} \right) \left( \frac{|M_{i}|}{|s_{i}'|} \right) e^{-\gamma(j-1)} n.
\]

Therefore, the complement sparsity of \( j^{th} \) tuples is at least \( \kappa(U) - \gamma j \ln n \), proving the \( j^{th} \) lower bound \( \lambda_j \).
To show (37), we have $|s'_{j-1}| = \frac{|M_{j-1}|m}{n} + \delta$ for some real number $\delta \in (-1, 1)$ before the $j^{th}$ step by induction hypothesis. Suppose that $\delta \geq 0$. Then the above process sets

$$|s'_j| = \left\lceil \frac{|M_j|m}{n} \right\rceil$$

meaning that

$$|s_j| = |s'_{j-1}| - |s'_j| = \frac{|M_{j-1}|m}{n} + \delta - \frac{|M_j|m}{n} - \delta' = \frac{|X_j|m}{n} + \delta - \delta'$$

$$\Rightarrow \quad \left| |s_j| - \frac{|X_j|m}{n} \right| < 1.$$ 

Thus (37) holds when $\delta \geq 0$. By symmetry it is true for $\delta \leq 0$ also. This completes the proof of induction step.

After the $q^{th}$ step, the complement sparsity of $\mathcal{F}$ is at least $\kappa(U) - q \ln n = \Omega(\kappa(U))$ since $\kappa(U) \gg q \ln n$. The size of each $s_j$ is either $\left\lfloor \frac{|M_j|m}{n} \right\rfloor$ or $\left\lceil \frac{|M_j|m}{n} \right\rceil$. Hence, the constructed $\mathcal{F}$ is a space-proportional split of $U$ with complement sparsity $\Omega(\kappa(U))$. 

REFERENCES

A. A. Razborov, “Lower Bounds for the Monotone Complexity of Some Boolean Functions”, Sov. Math. Dok., Vol. 31, pp. 354–357, 1985.

N. Alon and R. B. Boppana, “The Monotone Circuit Complexity of Boolean Functions”, Combinatorica, 1, Vol. 7, pp. 1–22, 1992.

A. A. Razborov and S. Rudich, “Natural Proofs”, Journal of Computer and System Sciences, Vol. 55, pp. 24–35, 1997.

J. Fukuyama, “On the Extension of an $m$-Set Family”, Congressus Numerantium, Vol. 173, pp. 35–41, 2005.

A. A. Razborov, “Lower Bounds on Monotone Complexity of the Logical Permanent”, Matematicheskie Zametki, Vol. 37, No. 6, pp. 887–900, June, 1985.

J. Fukuyama, “On the Topology of The Hamming Distance between Set Systems”, Congressus Numerantium, Vol. 161, pp. 41–63, 2003.

J. Fukuyama, “Some New Facts about the Hamming Space”, Congressus Numerantium, Vol. 168, pp. 21–31, 2004.

R. Graham, D. Knuth, and O Patashnik, “Concrete Mathematics”, Addison-Wesley, 1994.

S. Jukna, “Extremal Combinatorics, 2nd Edition”, Springer-Verlag, 2011

S. Jukna, “Boolean Function Complexity”, Springer, 2012

C. H. Papadimitriou, “Computational Complexity”, Addison-Wesley, 1994

E. Tardos, “The Gap Between Monotone and Non-Monotone Circuit Complexity is Exponential”, Combinatorica, 4, Vol. 7, pp. 141–142, 1987.

H. Vollmer, “Introduction to Circuit Complexity”, Berlin: Springer-Verlag, 1999.

A. A. Razborov and S. Rudich “Natural proofs”. Journal of Computer and System Sciences 55: 24735, 1997.