On the Information Capacity of Nearest Neighbor Representations

Kordag Mehmet Kilic*, Jin Sima† and Jehoshua Bruck‡

*Electrical Engineering, California Institute of Technology, USA, (kkilic,bruck)@caltech.edu
†Electrical and Computer Engineering, University of Illinois Urbana-Champaign, USA, jsima@illinois.edu

Abstract—The von Neumann Computer Architecture has a distinction between computation and memory. In contrast, the brain has an integrated architecture where computation and memory are indistinguishable. Motivated by the architecture of the brain, we propose a model of associative computation where memory is defined by a set of vectors in \( \mathbb{R}^n \) (that we call anchors), computation is performed by convergence from an input vector to a nearest neighbor anchor, and the output is a label associated with an anchor. Specifically, in this paper, we study the representation of Boolean functions in the associative computation model, where the inputs are binary vectors and the corresponding outputs are the labels (0 or 1) of the nearest neighbor anchors. The information capacity of a Boolean function in this model is associated with two quantities: (i) the number of anchors (called Nearest Neighbor (NN) Complexity) and (ii) the maximal number of bits representing entries of anchors (called Resolution). We study symmetric Boolean functions and present constructions that have optimal NN complexity and resolution.

I. INTRODUCTION

Inspired by the architecture of the brain, we propose a model of associative computation where memory is defined by a set of vectors in \( \mathbb{R}^n \) (that we call anchors), computation is performed by convergence from an input vector to a Nearest Neighbor anchor, and the output is a label associated with an anchor. This paradigm relates to the ability of the brain to quickly classify objects and map those to the syntax of natural languages, for example, it is instantaneous and effortless, even for a child, to recognize a ‘dog’, a ‘cat’, or a ‘car’.

In this paper, we study the representation of Symmetric Boolean Functions in the associative computation model, where the inputs are binary vectors and the corresponding outputs are the labels (0 or 1) of the Nearest Neighbor anchors. The information capacity of a Boolean function in this model is associated with two quantities: (i) the number of anchors (called Nearest Neighbor (NN) Complexity [5]) and (ii) the maximal number of bits representing entries of anchors (called Resolution). Metaphorically speaking, we are interested in the number of ‘words’ (anchors) and the size of the ‘alphabet’ (resolution) required to represent a Boolean function.

Let \( d(a, b) \) denote the Euclidean distance between the vectors \( a, b \in \mathbb{R}^n \). The Nearest Neighbor Representation and Complexity are defined in the following manner.

Definition 1. The Nearest Neighbor (NN) Representation of a Boolean function \( f \) is a set of anchors consisting of the disjoint subsets \( (P, N) \) of \( \mathbb{R}^n \) such that for every \( X \in \{0, 1\}^n \) with \( f(X) = 1 \), there exists \( p \in P \) such that for every \( n \in N \), \( d(X, p) < d(X, n) \), and vice versa. The size of the NN representation is \( |P \cup N| \).

Namely, the size of the NN representation is the number of anchors. Naturally, we are interested in representations of minimal size.

Definition 2. The Nearest Neighbor Complexity of a Boolean function \( f \) is the minimum size over all NN representations of \( f \), denoted by \( NN(f) \).

Some 2-input examples that can be easily verified are given below. We say \( f(X) = 1 \) (or 0) and their corresponding anchors are positive (or negative).

\[
\begin{array}{c|c|c|c|c}
X_1 & X_2 & \text{AND}(X) & \text{OR}(X) & \text{XOR}(X) \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 \\
\end{array}
\]

Fig. 1. The 2-input Boolean \( \text{AND}(X) = x_1 \wedge x_2 \), \( \text{OR}(X) = x_1 \vee x_2 \), and \( \text{XOR}(X) = x_1 \oplus x_2 \) functions, their corresponding truth values, and their NN representations. The positive values and anchors are in blue while the negative values and anchors are in red.

The examples given in Fig. 1 belong to a class of Boolean functions called Symmetric Boolean Functions. Let \( |X| \) denote the number of 1s in a binary vector \( X \).

Definition 3. A Boolean function is called symmetric \( f(X) = f(\sigma(X)) \) where \( \sigma(\cdot) \) can be any permutation function. Namely, a symmetric Boolean function \( f(X) \) is a function of \( |X| \).

We emphasize that although Boolean functions are evaluated over binary vectors, the anchor points can have real valued entries. Nevertheless, this need not to be the case as we can consider all the \( 2^n \) nodes of the Boolean hypercube as anchors and assign them to \( P \) or \( N \) based on the values of \( f \). This implies that \( NN(f) \leq 2^n \) for any \( n \)-input Boolean function. Specifically, we define the Boolean Nearest Neighbor Complexity (BNN) for the case where we restrict all anchors to be binary vectors. Thus, it is easy to see that \( NN(f) \leq BNN(f) \leq 2^n \) [5]. This upper bound on the NN complexity is essentially loose for almost all Boolean functions.
Table 1

| $X_1$ | $X_2$ | $|X|$ | XOR($X$) | BNN Representation |
|-------|-------|------|---------|-------------------|
| 0     | 0     | 0    | 0       | (0 0)             |
| 0     | 1     | 1    | 1       | (0 1)             |
| 1     | 0     | 1    | 1       | (1 0)             |
| 1     | 1     | 2    | 0       | (1 1)             |

Fig. 2. The 2-input XOR functions with its optimal BNN representation. It is known that $BNN(XOR) = 4$ whereas $NN(XOR) = 3$.

The discrepancy between NN and BNN is illustrated for the XOR function in Fig. 2. The generalization of XOR to $n$ variables is called the PARITY function. PARITY($X$) is 1 if the number of 1s in $X$ is odd, i.e., $PARITY(X) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ and we have a dramatic gap between NN and BNN.

**Theorem 1** ([5]). Let $f$ be the PARITY of $n$-inputs. Then,
- $NN(f) = n + 1$
- $BNN(f) = 2^n$

The fundamental difference between NN and BNN is the number of bits that are used to represent the anchor entries. To quantify it, we define the resolution of an NN representation. Loosely speaking, the number of bits to represent a number is its resolution and the resolution of a representation is the maximum resolution for a single entry for all anchor points. We say that $A \in \mathbb{R}^{m \times n}$ is an anchor matrix if each row corresponds to an anchor $a_i$ for an NN representation of an $n$-input Boolean function with $m$ anchors. We remark that we can assume that $A \in \mathbb{Q}^{m \times n}$ without loss of generality.

**Definition 4.** The resolution ($RES$) of a rational number $a/b$ is $RES(a/b) = \lceil \max\{\log_2|a + 1|, \log_2|b + 1|\} \rceil$ where $a, b \in \mathbb{Z}$, $b \neq 0$, and they are coprime.

For a matrix $A \in \mathbb{Q}^{m \times n}$, $RES(A) = \max_{i,j} RES(a_{ij})$. The resolution of an NN representation is $RES(A)$ where $A$ is the corresponding anchor matrix.

Given the PARITY example in Theorem 1, we observe that there is a trade-off between the NN complexity and the associated resolution. The $BNN(PARITY)$ is exponential in $n$ with resolution 1 while $NN(PARITY)$ is linear in $n$ with resolution $\lceil \log_2(n + 1) \rceil$.

Our paper is inspired by the work of Hajnal, Liu and Turán [5]. Their paper provided several NN complexity results, including the construction for PARITY (see Theorem 1), however, it only proposed the suboptimal PARITY-based construction for general symmetric functions (see Proposition 1). We briefly mention some additional relevant background: Nearest neighbors classification is a well-studied topic in Information Theory and Machine Learning [3, 7]. Optimizing the representation of a set by using NN was first discussed in the context of minimizing the size of a training set, namely, finding a minimal training set that NN represents the original training set [17, 23]. The idea to represent Boolean functions via the NN paradigm was first studied in [4] where BNN was considered with distance measures that are chosen to optimize the complexity of the NN representation. This work was extended to inclusion-based similarity [18] and it was proved that it provides polynomial size representations for DNF and CNF formulas.

In this paper we focus on the study of the NN representations of symmetric Boolean functions. Specifically, we construct optimal size NN representations for symmetric Boolean functions. These functions are useful to prove many complexity results about Boolean functions [2, 9, 22] and more importantly, in the context of nearest neighbors, any $n$-input Boolean function can be interpreted as an $2^n$-input symmetric Boolean function [20]. Therefore, results on symmetric Boolean functions can provide insights about our understanding of Boolean function complexity. Additionally, we study the trade-offs between the NN complexity (the number of anchors in the NN representation) and the NN resolution (the maximal number of bits to represent entries of the anchors). A non-intuitive discovery is that for symmetric Boolean functions, optimality in complexity and resolution is achieved by anchors that are asymmetric in their entries!

### A. The NN Complexity of Symmetric Functions

The PARITY construction mentioned in Theorem 1 can be extended to arbitrary symmetric Boolean functions. The idea is to assign an anchor point to each value of $|X|$. We have the following construction of an NN representation with an anchor matrix $A$ for the given symmetric Boolean function $f(X)$.

$$
\begin{array}{c|c|c}
|X| & f(X) & A \\
\hline
0 & 0 & a_1 \\
1 & 0 & a_2 \\
2 & 1 & a_3 \\
3 & 1 & a_4 \\
4 & 1 & a_5 \\
5 & 0 & a_6 \\
\end{array}
$$

Proposition 1 ([5]). Let $f$ be a symmetric Boolean function with $n$ input. Then, $NN(f) \leq n + 1$.

Proposition 1 is proven by construction and in general, the matrix $A \in \mathbb{R}^{(n+1) \times n}$ contains the anchor $a_i = ((i - 1)/n, \ldots, (i - 1)/n)$ where $P$ and $N$ is a partition of $\{a_1, \ldots, a_{n+1}\}$. Regardless of the function itself, this construction requires $n + 1$ anchors and we call it the PARITY-based construction. In contrast, we know that there are examples of symmetric Boolean functions that require a smaller number of anchors and Proposition 1 is evidently far from being optimal.

One can notice that there is a single transition from 0s to 1s in $f(X)$ for both AND and OR when they are enumerated by $|X|$. Furthermore, the correspondence between $f(X)$ and $|X|$ is particularly useful to measure the complexity of symmetric Boolean functions in Circuit Complexity Theory [12, 13, 19]. These observations motivate us to define the notion of an interval for a symmetric Boolean function in order to measure the NN complexity.
Definition 5. Let $a \leq b \leq n$ be some non-negative integers. An interval $[a, b]$ for an $n$-input symmetric Boolean function is defined as follows:

1) $f(X)$ is constant for $|X| \in [a, b]$.
2) If $a \neq 0$ and $b \neq n$, $f(X_1) \neq f(X_2)$ for any $|X_1| \in [a, b]$ and

   - If $a > 0$ and $b < n$, $|X_2| = a - 1$ and $|X_2| = b + 1$.
   - If $a = 0$ and $b < n$, $|X_2| = b + 1$.
   - If $a > 0$ and $b = n$, $|X_2| = a - 1$.

The quantity $I(f)$ is the total number of intervals for a symmetric Boolean function $f$.

We usually refer to an interval $[a, b]$ for an $n$-input symmetric Boolean function shortly as an interval. The example function in Eq. 1 and $XOR$ have $I(f) = 3$ (see Fig.2) while $AND$ and $OR$ have $I(f) = 2$ (see Fig. 3).

It seems natural to simplify the construction in Proposition 1 by assigning an anchor to each interval to reduce the size of the representation to $I(f)$. We call this an $I(f)$ interval-anchor assignment where there is a one-to-one map between an anchor and an interval of $f(X)$. We assign the $i$th interval $[a, b]$ to the anchor $a_i$ and enumerate the beginning and the end of the interval assigned to it by $I_{a_i} + 1$ and $I_{a_i}$ respectively by using the values of $|X|$. We take $I_0 = -1$ for consistency.

The function below is an example where $f(X)$ is a 6-input symmetric function and $I(f) = 3$ and $(a_1, a_2, a_3)$ is an $I(f)$ interval-anchor assignment.

\[
\begin{array}{c|c|c|c}
|X| & AND(X) & OR(X) \\
\hline
0 & 0 & 0 \\
1 & 0 & 1 \\
2 & 1 & 1 \\
\end{array}
\]

Fig. 3. The PARITY-based constructions for 2-input AND and OR functions. Clearly, the construction given in Fig. 1 has smaller size and the same resolution.

\[
\begin{array}{c|c|c}
|X| & f(X) \\
\hline
0 & 0 \rightarrow a'_1 \\
1 & 0 \rightarrow a'_1 \\
2 & 1 \rightarrow a'_2 \\
3 & 1 \rightarrow a'_2 \\
4 & 1 \rightarrow a'_2 \\
5 & 0 \rightarrow a'_3 \\
\end{array}
\]

Unfortunately, PARITY-based extensions cannot be applied to all symmetric Boolean functions. We give an example where there is no PARITY-based extension in Appendix B. Informally, symmetric Boolean functions treat each input $x_i$ with the same importance and we show that by breaking up the symmetry in the anchor entries and taking the whole NN representation farther away from the Boolean hypercube, we can reduce the size of representation to $I(f)$.

Is $NN(f) = I(f)$ for symmetric Boolean functions? To conclude that we need a corresponding lower bound for an arbitrary symmetric Boolean function. It is still an open problem, however, we are able to prove it for periodic symmetric Boolean functions.

Definition 6. A symmetric Boolean function is called periodic if each interval has the same length, which is denoted by $T$. $T$ is also called period.

More precisely, if $f$ is a periodic symmetric Boolean function with period $T$, then for $(k-1)T \leq |X| < kT$, $f(X) = 0$ (or 1) for odd $k$ and $f(X) = 1$ (or 0) for even $k$. PARITY is a periodic symmetric function with period $T = 1$. Periodic symmetric Boolean functions are useful in other works as well (see the Complete Quadratic function in [2] where $T = 2$).

B. NN Representations of Linear Threshold Functions

A linear threshold function is a weighted summation of binary inputs fed into a threshold operation, i.e., $1 \{ \sum_{i=1}^{n} w_i x_i \geq b \}$ where $1 \{ \cdot \}$ is an indicator function that evaluates 1 if the inside expression is true and 0 otherwise. $b$ is called the threshold and without loss of generality, $w_i \in Z$. For non-constant linear threshold functions, an optimal hyperplane argument using support vectors implies that $NN(f) = 2$ and the converse also holds in that any Boolean function with $NN(f) = 2$ is a linear threshold function. These functions are interesting because they are the building blocks of neural networks.

We define an $n$-input symmetric linear threshold function as a symmetric Boolean function with $I(f) = 2$ intervals, i.e., $f(X) = 1 \{|X| \geq b\}$, namely, $w_i = 1$ for all $i$. Interestingly, depending on the value of $b$, these functions can have different orders of BNN complexities.

Theorem 2 ([15]). Let $f$ be an $n$-input symmetric linear threshold function with threshold $b$. Then,

- If $b = n/2$ and $n$ is odd, $BNN(f) = 2$.
- If $b = n/2$ and $n$ is even, $BNN(f) \leq n/2 + 2$.
- If $b = [n/3]$, $BNN(f) = 2^{\Omega(n)}$.
If all the anchors are on the Boolean hypercube, then the NN complexity can be as large as $2^{O(n)}$. Nevertheless, in this work, we show that the resolution of a 2-anchor NN representation of a linear threshold function is bounded by the resolution of the weights. This implies that there are constant resolution NN representations for all symmetric linear threshold functions. In general, for linear threshold functions, the resolution is $O(n \log n)$ [1, 8, 14].

### C. Contributions and Organizations

Our key contributions are:

- New constructions are presented for arbitrary $n$-input symmetric Boolean functions with $I(f)$ anchors and $O(\log n)$ resolution.
- The new constructions have optimal NN complexity for all periodic symmetric Boolean functions.
- There is always an optimal NN representation both in size and resolution for any $n$-input symmetric linear threshold function.
- The NN representations of some symmetric Boolean functions require the resolution to be $\Omega(\log n)$ and that it is optimal for some symmetric Boolean functions.

The organization of the paper is as follows. In Section II, we present an explicit construction and thus, an upper bound for the NN complexity of any symmetric Boolean function. In Section III, we prove a lower bound for periodic symmetric Boolean functions and conclude that $NN(f) = I(f)$ for this class of functions. Later, in Section IV, we study the resolution of our NN constructions, prove that it is $O(\log n)$ and that it is optimal for some symmetric Boolean functions. In addition, we present optimal NN representation both in size and resolution for symmetric linear threshold functions. The appendices include the details of the proofs.

### II. The Upper Bound on the NN Complexity of Symmetric Boolean Functions

In this section, we present an explicit construction for the NN representation of symmetric Boolean functions with $I(f)$ intervals.

**Theorem 3.** For an $n$-input symmetric Boolean function $f$, $NN(f) \leq I(f)$.

Moreover, a matrix $B = \mathbb{1} + \epsilon M$ where $B \in \mathbb{R}^{I(f) \times n}$ can always be used for a construction given that $\epsilon > 0$ is sufficiently small where $\mathbb{1}$ is an all-one matrix and $M \in \mathbb{R}^{I(f) \times n}$ is a full row rank matrix.

For arbitrary symmetric Boolean functions, we first derive necessary and sufficient conditions for an $I(f)$ interval-anchor assignment.

**Lemma 1.** Let $f$ be an $n$-input symmetric Boolean function and $A \in \mathbb{R}^{I(f) \times n}$ be an anchor matrix for an $I(f)$ interval-anchor assignment. Then, the following condition is necessary and sufficient for any $i \in \{2, \ldots, I(f)\}$ and $k < i$:

$$\max_{X \in \{0,1\}^n : |X| = I_{i-1}} \sum_{j=1}^{n} (a_{ij} - a_{kj}) x_j < \frac{1}{2} \sum_{j=1}^{n} (a_{ij}^2 - a_{kj}^2)$$

$$< \min_{X \in \{0,1\}^n : |X| = I_{i-1} + 1} \sum_{j=1}^{n} (a_{ij} - a_{kj}) x_j$$

(4)

In general, all $a_{ij}$s can be different compared to PARITY-based extensions. However, we cannot freely choose them. We have the following observation to prove Lemma 1.

**Proposition 2.** For any $I(f)$ interval-anchor assignment with an anchor matrix $A \in \mathbb{R}^{I(f) \times n}$, $a_{ij} > a_{i-1, j}$ for all $i \in \{2, \ldots, I(f)\}$ and $j \in \{1, \ldots, n\}$.

It is possible to simplify the necessary and sufficient conditions given in Lemma 1. We prove that looking at $(k, i)$ pairs in the form $(i-1, i)$ still provides the necessary and sufficient information compared to all $(k, i)$ pairs such that $k < i$.

**Lemma 2.** Let $f$ be an $n$-input symmetric Boolean function and $A \in \mathbb{R}^{I(f) \times n}$ be an anchor matrix for an $I(f)$ interval-anchor assignment. Then, the following condition is necessary and sufficient for any $i \in \{2, \ldots, I(f)\}$:

$$\max_{X \in \{0,1\}^n : |X| = I_{i-1}} \sum_{j=1}^{n} (a_{ij} - a_{i-1, j}) x_j < \frac{1}{2} \sum_{j=1}^{n} (a_{ij}^2 - a_{i-1, j}^2)$$

$$< \min_{X \in \{0,1\}^n : |X| = I_{i-1} + 1} \sum_{j=1}^{n} (a_{ij} - a_{i-1, j}) x_j$$

(5)

We can now prove Theorem 3 by finding explicit sets of anchors satisfying Lemma 2. In Theorem 3, the $B$ matrix is actually constructed by $b_{ij} = a_{ij} - a_{i-1, j}$ for $i \in \{2, \ldots, I(f)\}$ and $j \in \{1, \ldots, n\}$. Intuitively, as long as the entries of $B$ do not differ much, we can find anchors where the LHS and the RHS in Eq. (5) should hold. Therefore, $B = \mathbb{1} + \epsilon M$ is a valid choice for sufficiently small $\epsilon > 0$.

In Theorem 3, we remark that the full row rank $B$ matrix property is only sufficient to find a construction. For example, if $b_{ij} = 1/n$ for all $i, j$, this construction can be reduced to the PARITY-based construction given in [5].

In general, the PARITY-based approach results in $O(\log n)$ resolution. We can pick $B$ such that any symmetric Boolean function can also have $O(\log n)$ resolution with $I(f)$ anchors. We present a family of examples of $B$ with $O(\log n)$ resolution in Section IV. Such an example was already given in Appendix B.

### III. Lower Bounds on the NN Complexity of Symmetric Boolean Functions

There are various circuit complexity lower bounds on symmetric Boolean functions [15, 20, 21]. Similarly, lower bounds for the number of anchors can be proven for PARITY using $\{1, 2\}$-sign representations of Boolean functions [5, 6].


prove a more general result for periodic symmetric Boolean functions.

**Theorem 4.** For a periodic symmetric Boolean function of \( I(f) \) intervals, \( NN(f) \geq I(f) \).

To prove this Theorem, we use a necessary condition for any NN representation of a periodic symmetric Boolean function.

**Proposition 3.** Consider an NN representation of a periodic symmetric Boolean function \( f(X) \) of period \( T \) with an anchor matrix \( A \in \mathbb{R}^{m \times n} \). Assume that \( \sum_{j=1}^{T} a_{1j} \geq \sum_{j=1}^{T} a_{ij} \) for any \( i \geq 2 \). Then, no binary vectors that are closest to \( a_1 \) can have all 0s in the first \( T \) coordinates.

We first claim that the assumption in Proposition 3 can be assumed without loss of generality by first observing that the NN representations of symmetric Boolean functions are equivalent up to permutations of anchors. We prove this explicitly in the Appendix A2. Then, we find the anchor with the maximal \( T \) sum in the entries and rearrange the rows and columns so that the claim \( \sum_{j=1}^{T} a_{1j} \geq \sum_{j=1}^{T} a_{ij} \) for any \( i \geq 2 \) holds.

In addition, we prove that if \( I(f) \leq 4 \), then the NN complexity of any \( n \)-input symmetric Boolean function has the lower bound \( NN(f) \geq I(f) \). We present the proof in Appendix C based on the tools developed in Section IV.

**IV. THE RESOLUTION OF NN REPRESENTATIONS**

It is known that the PARITY-based approach results in \( \lfloor \log_2(n+1) \rfloor \) resolution constructions for symmetric Boolean functions. We claim that the construction presented in Section II can admit \( O(\log n) \) resolution. Let Moore-Penrose pseudoinverse of a matrix \( A \) be denoted by \( A^+ \).

**Theorem 5.** Suppose that we are given a rational matrix \( B \in \mathbb{Q}^{m \times n} \) such that \( B = I_{m \times n} + \epsilon I_{m \times n} \) where \( I_{m \times n} \) is an all-one matrix, \( \epsilon \) is a rational constant, and \( I_{m \times n} \) is a submatrix of the \( n \times n \) identity matrix with the first \( m \) rows. Then, \( RES(B^+) = O(\log n) \).

**Corollary 5.1.** For every \( n \)-input symmetric Boolean function, there is an anchor matrix \( A \in \mathbb{Q}^{(f) \times n} \) such that \( RES(A) = O(\log n) \).

We refer to Theorem 3 in [11] to find the entries of \( B^+ \) explicitly and to prove Theorem 5.

Now we consider the resolution of the NN representation of linear threshold functions. By a hyperplane argument, we prove the following result which can be verified algebraically.

**Theorem 6.** Let \( f \) be a non-constant \( n \)-input linear threshold function with weight vector \( w \in \mathbb{Z}^n \) and the threshold term \( b \in \mathbb{Z} \). Then, there is a 2-anchor NN representation with resolution \( O(RES(w)) \). In general, the resolution is \( O(n\log n) \).

**Corollary 6.1.** Let \( f \) be an \( n \)-input symmetric linear threshold function with threshold \( b \). Then, there is a 2-anchor NN representation with resolution \( O(1) \).

Hence, symmetric linear threshold functions have 2 intervals and optimal NN representations in size and resolution. It is easy to verify that

- \( a_1 = (0, 0, 0, -1/2, -1, \ldots, -1) \)  
- \( a_2 = (2, 2, 2, 3/2, 1, \ldots, 1) \)

is always an NN representation where the number of 0s in \( a_1 \) (and 2s in \( a_2 \)) is \( b - 1 \).

However, if the number of intervals is at least 3, the resolution is lower bounded by \( \Omega(\log n) \) for some symmetric Boolean functions and the construction presented in Section II has optimal resolution by Corollary 5.1.

**Theorem 7.** Let \( f \) be an \( n \)-input symmetric Boolean function where \( f(X) = 1 \) for \( |X| = \lfloor n/2 \rfloor + 1 \) and 0 otherwise. Then, any 3-anchor NN representation has \( \Omega(\log n) \) resolution.

There is an anchor matrix \( A \) for the function in Theorem 7 such that \( RES(A) = O(\log n) \) via PARITY-based extensions. To prove Theorem 7, we first prove that all but two symmetric Boolean functions with 3 intervals require an interval-anchoring assignment. We then use the necessary and sufficient condition in Lemma 2 to prove the resolution lower bound.

**Lemma 3.** Let \( f \) be an \( n \)-input symmetric Boolean function with 3 intervals. Then, any \( I(f) \)-anchoring NN representation of \( f \) needs to be an interval-anchoring assignment except for the function \( f(X) = 1 \) (or 0) for \( |X| \in \{0, n\} \) and 0 (or 1) otherwise.

An alternative NN representation for the function in Lemma 3 is described in Appendix C.

Finally, we use a circuit theoretic lower bound for a linear threshold function to give a lower bound on the NN representation size depending on resolution in Appendix D.

**V. CONCLUDING REMARKS**

We study the information capacity of symmetric Boolean functions in the associative computation model. The information capacity consists of two quantities: NN complexity and resolution. Namely, the number of anchors in the NN representation and the number of bits required to represent the anchors. Specifically, we prove an upper bound on the NN complexity \( NN(f) \leq I(f) \) and that this bound is tight for periodic symmetric Boolean functions. The upper bound of \( I(f) \) on the NN complexity is based on a new NN construction with resolution \( O(\log n) \). We prove that our upper bound on resolution is tight for some symmetric Boolean functions. Surprisingly, to achieve optimality, the construction must be asymmetric in the entries of the anchors, contrary to the properties of symmetric Boolean functions. In addition, optimal constructions in size \( (NN(f) = 2) \) and resolution (a constant) are given for symmetric linear threshold functions.

For any symmetric Boolean function \( f \), we conjecture that \( NN(f) = I(f) \). While linear threshold functions have \( NN(f) = 2 \), the resolution is bounded by \( O(n\log n) \). An intriguing open problem is to find NN representations for all
linear threshold functions of polynomial NN complexity and logarithmic (or constant) resolution.

ACKNOWLEDGEMENT

This research was partially supported by the Carver Mead New Adventure Fund.

REFERENCES


Proof of Proposition 2. Consider two Boolean vectors \( X = (x_1, \ldots, 0, \ldots, x_n) \) and \( X' = (x_1, \ldots, 1, \ldots, x_n) \) where they only differ at \( t \)th location. Assume that this occurs at the boundary for an interval, i.e., \( X \) and \( X' \) are closer to \( a_{i-1} \) and \( a_i \), respectively. Then,

\[
d(a_{i-1}, X)^2 - d(a_i, X)^2 < 0 \tag{8}
\]

\[
d(a_{i-1}, X')^2 - d(a_i, X')^2 > 0 \tag{9}
\]

Subtracting both inequalities, we get \( d(a_{i-1}, X)^2 - d(a_{i-1}, X')^2 = (d(a_i, X)^2 - d(a_i, X')^2) < 0 \). Since \( X \) and \( X' \) only differ at the \( t \)th location, we get \( 2(a_{i-1} - 1) = 2(a_i - 1) < 0 \), hence, \( a_{i-1} > a_i \).

Proof of Lemma 1. We begin by writing the simplest necessary and sufficient condition for an \( f(I) \) interval-anchored assignment. Consider any two anchors \( a_i \) and \( a_k \) and assume that \( |X| \in [I_{i-1} + 1, I_i] \) so that \( X \) is closer to \( a_i \).

\[
d(a_i, X)^2 - d(a_k, X)^2 < 0 \tag{10}
\]

which can be written as

\[
\sum_{j=1}^{n}(a_{ij} - x_j)^2 - \sum_{j=1}^{n}(a_{kj} - x_j)^2 < 0 \tag{11}
\]

for \( i \in \{2, \ldots, I(f)\} \) and \( k < i \). This implies

\[
\frac{1}{2} \sum_{j=1}^{n}(a_{ij}^2 - a_{kj}^2) < \min_{X \in \{0,1\}^{n}: |X| \in [I_{i-1}+1, I_i]} \sum_{j=1}^{n}(a_{ij} - a_{kj})x_j \tag{12}
\]

The RHS term is minimized when \( |X| = \sum_{j=1}^{n}x_j = I_{i-1} + 1 \) by Proposition 2 because each \( a_{ij} - a_{kj} > 0 \). Then,

\[
\frac{1}{2} \sum_{j=1}^{n}(a_{ij}^2 - a_{kj}^2) < \min_{X \in \{0,1\}^{n}: |X| = I_{i-1} + 1} \sum_{j=1}^{n}(a_{ij} - a_{kj})x_j \tag{13}
\]

We similarly do the analysis for the anchors \( a_i \) and \( a_k \) but this time assuming that \( |X| \in [I_{k-1} + 1, I_k] \) so that \( X \) is closer to \( a_k \). We still take \( k < i \). Then,

\[
\frac{1}{2} \sum_{j=1}^{n}(a_{kj}^2 - a_{ij}^2) < \min_{X \in \{0,1\}^{n}: |X| = I_{k-1} + 1} \sum_{j=1}^{n}(a_{kj} - a_{ij})x_j \tag{14}
\]

Now, the RHS is maximized when \( |X| = I_k \) by Proposition 2. By combining (13) and (15), we complete the proof.

Proof of Lemma 2. It is obvious that this is necessary if we apply \( k = i - 1 \). We will show that we can use the Eq. (5) to obtain Lemma 1 by summing them telescopically until some fixed \( k < i \).

\[
\max_{X \in \{0,1\}^{n}: |X| = I_{i-1}} \sum_{j=1}^{n}(a_{ij} - a_{(i-1)j})x_j < \frac{1}{2} \sum_{j=1}^{n}(a_{ij}^2 - a_{(i-1)j}^2)
\]

\[
< \min_{X \in \{0,1\}^{n}: |X| = I_{i-1} + 1} \sum_{j=1}^{n}(a_{ij} - a_{(i-1)j})x_j
\]

\[
< \min_{X \in \{0,1\}^{n}: |X| = I_k} \sum_{j=1}^{n}(a_{kj} - a_{(i-1)j})x_j
\]

\[
< \min_{X \in \{0,1\}^{n}: |X| = I_k} \sum_{j=1}^{n}(a_{(k+1)j} - a_{kj})x_j
\]

Let us combine the inequalities and focus on the RHS.

\[
\sum_{l=k+1}^{i} \min_{X \in \{0,1\}^{n}: |X| = I_{l-1}} \sum_{j=1}^{n}(a_{lj} - a_{(l-1)j})x_j \tag{16}
\]

We can replace the constraint sets of all of the minimization expressions with \( I_{l-1} + 1 \) because

\[
\min_{X \in \{0,1\}^{n}: |X| = I_{l-1} + 1} \sum_{j=1}^{n}(a_{lj} - a_{(l-1)j})x_j
\]

by Proposition 2 and \( l \leq i \). Then, we combine the objective functions telescopically again so that

\[
\frac{1}{2} \sum_{j=1}^{n}(a_{ij}^2 - a_{kj}^2) < \min_{X \in \{0,1\}^{n}: |X| = I_{i-1} + 1} \sum_{j=1}^{n}(a_{ij} - a_{kj})x_j \tag{17}
\]

The LHS can be handled in a similar manner. Then, we can obtain Lemma 1 exactly.

Proof of Theorem 3. Let us define a matrix \( B \in \mathbb{R}^{I(f) \times n} \) where \( b_{ij} = a_{(i+1)j} - a_{ij} \). Rewriting the condition in Lemma 2 using the \( B \) matrix and transforming \( i \) to \( i+1 \), we get

\[
\max_{X \in \{0,1\}^{n}: |X| = I_i} \sum_{j=1}^{n}b_{ij}x_j < \frac{1}{2} \sum_{j=1}^{n}(-b_{ij}^2 + 2b_{ij} \sum_{k=1}^{i} b_{kj} + b_{ij}a_{ij})
\]

\[
< \min_{X \in \{0,1\}^{n}: |X| = I_{i+1}} \sum_{j=1}^{n}b_{ij}x_j \tag{19}
\]
where the middle term is computed by the identity 
\[ a_{ij} = a_{ij} + \sum_{k=1}^{n-1} b_{kj} = a_{ij} + b_{ij} + \sum_{k=1}^{n} b_{kj}. \]

\[ \frac{1}{2} \sum_{j=1}^{n} (a_{i+1,j}^2 - a_{ij}^2) = \frac{1}{2} \sum_{j=1}^{n} b_{ij} (b_{ij} + 2a_{ij}) \]

\[ = \frac{1}{2} \sum_{j=1}^{n} \left( b_{ij}^2 + 2b_{ij} (a_{ij} - b_{ij} + \sum_{k=1}^{n} b_{kj}) \right) \]

\[ = \frac{1}{2} \sum_{j=1}^{n} \left( -b_{ij}^2 + 2b_{ij} \sum_{k=1}^{n} b_{kj} + 2b_{ij} a_{ij} \right) \]

Let us take a convex combination of the LHS and RHS with some \( \lambda_i \in (0,1) \) to make it equal to the middle term for each \( i \in \{1, \ldots, I(f) - 1\} \). Therefore, we want to solve \( a_1 = (a_{i1}, \ldots, a_{1n}) \) for the \( B_{A1} = c \) where

\[ (B_{A1})_i = c_i = \frac{1}{2} \sum_{j=1}^{n} b_{ij}^2 - \sum_{j=1}^{n} b_{ij} \sum_{k=1}^{n} b_{kj} \]

\[ + \lambda_i \max_{X \in \{0,1\}^n : |X|_1 = l_i} \sum_{j=1}^{n} b_{ij} x_j \]

\[ + (1 - \lambda_i) \min_{X \in \{0,1\}^n : |X|_1 = l_i + 1} \sum_{j=1}^{n} b_{ij} x_j \]

As long as this system of linear equations is consistent, we have a solution for \( a_1 \) and a construction for \( I(f) \) interval-anchor assignment. If \( B \) is a full row rank matrix, the system is guaranteed to be consistent. As long as the LHS is smaller than the RHS in Eq. (19), any full row rank \( B \) matrixes will work. Therefore, we can claim that \( B = 1 + \epsilon M \) for sufficiently small \( \epsilon > 0 \) and full row rank \( M \) can be used to construct an anchor matrix for an arbitrary symmetric Boolean function.

**2) Proofs for Section III:**

**Proposition 4.** Suppose that an anchor matrix \( A \in \mathbb{R}^{m \times n} \) is an NN representation with \( n \) anchors for an \( n \)-input symmetric Boolean function. Then, any permutation of rows and columns of \( A \) is an NN representation for the same function.

**Proof of Proposition 4.** Permutation of rows is trivial and can be done for any NN representation. For the columns, we can use the definition of symmetric Boolean functions where \( f(X) = f(\sigma(X)) \) for any permutation \( \sigma(.) \). The closest anchor to a Boolean vector \( X \) can be found by

\[ \arg \min_i d(a_i, X)^2 = \arg \min_i |X|_2 - 2(AX) + |a_i|_2^2 \]

\[ = \arg \max_i \left( 2AX - \text{diag}(AA^T) \right)_i \]

where \(|.|_2\) denotes the Euclidean norm and \( \text{diag}(M) \) is the all-zero matrix except the diagonal entries of \( M \). Let \( P \in \{0,1\}^{m \times n} \) be a permutation matrix. Then,

\[ \arg \max_i \left( 2APX - \text{diag}(AA^T) \right)_i \]

is an anchor index assigned to the same anchor type (either positive or negative) because \( f \) is symmetric. Note that Eq. (26) is equivalent to

\[ \arg \max_i \left( 2(AP)X - \text{diag}((AP)(AP)^T) \right)_i \]

\[ = \arg \max_i \left( 2(AP)X - \text{diag}(AA^T) \right)_i \]

**Proof of Proposition 3.** We use a similar idea used for Proposition 2. Assume that there is a vector \( X = (0, \ldots, 0, x_{T+1}, \ldots, x_n) \) assigned to \( e_1 \) for contradiction. Let \( X' = (1, \ldots, 1, x_{T+1}, \ldots, x_n) \). Then,

\[ d(a_1, X)^2 < d(a_1, X)^2 \quad \forall i \geq 2 \]

\[ d(a_1, X')^2 > d(a_1, X')^2 \quad \text{for some } i \geq 2 \]

We get Eq. (28) by the assumption for contradiction. Eq. (29) is obtained by the fact that \( f(X) \neq f(X') \) because we have a jump of length \( T \) for a given value of \( |X|_2 \). Subtracting both, we get

\[ \sum_{j=1}^{T} a_{ij} < \sum_{j=1}^{T} a_{ij} \]

for some \( i \geq 2 \), which is the desired contradiction.

**Proof of Theorem 4.** Let \( f \) be a periodic symmetric Boolean function with period \( T \) so that \( n = I(f)T \). Suppose that the NN representation for this function has an anchor matrix \( A \in \mathbb{R}^{m \times n} \) where

\[ \sum_{j=(k-1)T+1}^{kT} a_{kj} \geq \sum_{j=(k-1)T+1}^{kT} a_{ij} \]

for any \( k \in \{1, \ldots, m - 1\} \) and \( k < i \). This holds without loss of generality because we can rank the maximal sums of the \( k \) entries of the anchors and rearrange the rows and columns of \( A \) by Proposition 4.

Iteratively, for each \( k \in \{1, \ldots, m - 1\} \), we see that if \( X_i = 0 \) for \( i \in \{1, \ldots, kT\} \), then \( a_1, \ldots, a_k \) cannot be assigned to \( X \) by Proposition 3. Since \( n = I(f)T \) for a periodic symmetric Boolean function, if \( m < I(f) \), there will remain \( X \) vectors with different \( f(X) \) values assigned to a single anchor, leading to a contradiction.

**3) Proofs for Section IV:**

**Theorem 8 (111).** Suppose that we are given a matrix \( A \in \mathbb{R}^{m \times n} \) and column vectors \( c \in \mathbb{R}^m, d \in \mathbb{R}^n \). Then, if \( c \) is in the column space of \( A \) and the quantity \( \beta = 1 + d^T A^+ c \neq 0 \), we have

\[ (A + cd^T)^+ = A^+ + \frac{1}{\beta} c d^T A^+ \]

\[ \frac{1}{\beta} (|k|^2 v^T + k\beta)(|v|^2 k^T A^+ + h\beta) \]

\[ \frac{1}{\beta} (|k|^2 |v|^2 + \beta^2) \]

where \( v = d^T (I - A^+ A), k = A^+ c, \) and \( h = d^T A^+ \).
Proof of Theorem 5. To apply Theorem 8, we pick \( c = 1_m \) and \( d = 1_n \) where \( 1_k \) denotes the all-one vector in \( \mathbb{R}^k \) for an integer \( k \). Also, pick \( A = c I_{m,n} \). Therefore, \( A^+ = \frac{1}{2} I_{m,n} \).

We compute the pseudoinverse of \( 1_m \times n + c I_{m,n} \) as the following closed form expression.

\[
\left( 1_m \times n + c I_{m,n} \right)^+ = \left[ \frac{1}{r} I_{m \times n} - \frac{1 + n/c}{m(n-m)+1} I_{m \times m} \right]
\]

Since \( m \leq n \) and \( c \) is a rational constant, the denominator of all entries of the pseudoinverse is a function of \( O(n^2) \) (and \( O(n) \) for the numerator). Therefore, the resolution of the entries of \( B^+ \) is \( O(\log n) \).

Proof of Corollary 5.1. For the matrix \( B = 1_m \times n + c M \) in Theorem 3 with \( m = I(f) - 1 \), we pick \( c = 1/2 \) and \( M = I_{m,n} \), where \( I_{m,n} \) is a submatrix of \( n \times n \) identity matrix with the first \( m \) rows. It can be easily verified that \( B \) matrix where \( b_{ij} = a_{ij} - a_{(i-1)j} \) satisfies the necessary and sufficient condition (5) for the interval-anchor assignment. The first anchor \( a_1 \) can be obtained by \( B^+ c \) where \( c \) is given in Eq. (23).

With \( \lambda_i = 1/2 \) for all \( i \in \{1, \ldots, m\} \), each entry of \( c \) depends on the entries on \( B \) and \( n \) polynomially. By Theorem 5, \( B^+ \) has entries with at most \( O(\log n) \) resolution and therefore, we see that the entries of \( a_1 = B^+ c \) has at most \( O(\log n) \) resolution (consequently, \( a_2, \ldots, a_m \)). In conclusion, we can always obtain an interval-anchor assignment with \( O(\log n) \) resolution.

Proof of Theorem 6. Geometrically, any 2-anchor NN representation construction of linear threshold functions can be written in the following form for a real number \( c > 0 \) and an arbitrary \( x^* \in \mathbb{R}^n \) such that \( u^T x^* = b^* \) where \( b^* = b - 0.5 \).

\[
\begin{align*}
  a_1 &= x^* - c w \\
  a_2 &= x^* + c w
\end{align*}
\]

This can be seen algebraically as well. We perturb \( b \) to \( b^* \) to consider the points on the hyperplane itself. We first claim that there exists \( x^* \) such that \( RES(x^*) \leq RES(w) \).

Since \( f(X) \) is not constant, there is always a pair of binary vectors \( X' \) and \( X'' \) such that \( u^T X' < b' < u^T X'' \) where \( X'_i = X''_i \) for all \( i \in \{1, \ldots, n\} \) except for a unique \( i = k \in \mathbb{Z} \). Then, we construct

\[
\begin{align*}
  x^*_i &= X'_i \quad \text{for } i \neq k \\
  x^*_k &= \frac{b' - u^T X'}{w_k}
\end{align*}
\]

Clearly, \( x^*_i \)'s are binary except \( i = k \) where \( |b' - u^T X'| < w_k \) and therefore, \( RES(x^*_i) = \log_2 w_k + 1 \). In conclusion, \( RES(x^*) = \log_2 w + 1 \) \( \leq RES(w) \).

Picking \( c = 1 \), we see that \( RES(A) = O(RES(w)) \). Moreover, since \( w_i = 2^{O(n \log n)} \) for \( i \in \{1, \ldots, n\} \) in general \([1, 8, 14]\), we conclude that \( RES(A) = O(n \log n) \).

Proof of Theorem 7. For the \( n \)-input symmetric Boolean function given in the Theorem, we always have an interval-anchor assignment by Lemma 3. For an interval-anchor assignment, let \( B \in \mathbb{R}^{2 \times n} \) \( b_{ij} = a_{ij} - a_{ij} > 0 \) for \( i \in \{1, 2\} \) and \( j \in \{1, \ldots, n\} \). Then, let us write the necessary and sufficient conditions by Lemma 2. Without loss of generality, assume that \( 0 < b_{11} \leq b_{12} \leq \cdots \leq b_{in} \) as we can always reorder the columns of \( B \). Then, the first condition will be

\[
\max_{X \in \{0,1\}^n} X[:n/2] \sum_{j=1}^n b_{ij} x_j < \frac{1}{2} \sum_{j=1}^n b_{ij} (b_{ij} + 2a_{1j})
\]

\[
\min_{X \in \{0,1\}^n} X[:n/2] \sum_{j=1}^n b_{ij} x_j < \frac{1}{2} \sum_{j=1}^n b_{ij} (b_{ij} + 2a_{1j})
\]

We have two important inequalities based on this condition.

\[
b_{1(n - \sqrt{n})} < b_{1(\sqrt{n+1})} + \frac{1}{\sqrt{n}} b_{11} \tag{39}\]
\[
b_{1n} < 2b_{1(\sqrt{n+1})} \tag{40}\]

To prove Eq. (39), we assume the contrary such that \( b_{1(n - \sqrt{n})} \geq b_{1(\sqrt{n+1})} + \frac{1}{\sqrt{n}} b_{11} \). Also, \( b_{1(n - \sqrt{n} + j)} \geq b_{1(n - \sqrt{n})} \) for \( 1 \leq j \leq \sqrt{n} \) and \( b_{1(n+1)} \geq b_{1j} \) for \( 1 \leq j \leq \sqrt{n} + 1 \). Hence, if we sum over \( j \in \{1, \ldots, \sqrt{n}\} \), we get

\[
\sum_{j=1}^{\sqrt{n}} b_{1(n - \sqrt{n} + j)} \geq \sum_{j=1}^{\sqrt{n}} \left( b_{1(j+1)} + \frac{1}{\sqrt{n}} b_{11} \right) \tag{41}\]

Let us add \( b_{1j} \) for \( j \in \{n - \lceil n/2 \rceil + 1, \ldots, n - \sqrt{n}\} \) to both sides. Then, we get

\[
\sum_{j=n-\lceil n/2 \rceil + 1}^{n} b_{1j} \geq \sum_{j=1}^{\sqrt{n}} b_{1(j+1)} + \frac{1}{\sqrt{n}} b_{11} + \sum_{j=n-\lceil n/2 \rceil + 1}^{n-\sqrt{n}} b_{1j} \tag{42}\]

where the LHS has \( \lceil n/2 \rceil \) many terms and the RHS has \( \lceil n/2 \rceil + 1 \) many terms. This contradicts Eq. (38).

Similarly, to prove Eq. (40), we assume the contrary such that \( b_{1n} \geq 2b_{1(\sqrt{n}+1)} \geq b_{11} + b_{12} \). If we add both sides

\[
\sum_{i=n-\lceil n/2 \rceil + 1}^{n} b_{1i} \geq \sum_{i=n-\lceil n/2 \rceil + 1}^{n} b_{1i} \tag{43}\]

where the LHS has \( \lceil n/2 \rceil \) many terms and the RHS \( \lceil n/2 \rceil + 1 \) many terms, contradicting Eq. (38).

We now want to bound the middle term in Eq. (38). We divide the sum in three parts. Essentially, we want to show that the main contribution in the value of the whole summation is due to the middle term.

\[
\sum_{j=1}^{\sqrt{n}} b_{1j} (b_{1j} + 2a_{1j}) + \sum_{j=\sqrt{n}+1}^{n-\sqrt{n}} b_{1j} (b_{1j} + 2a_{1j}) + \sum_{j=n-\sqrt{n}+1}^{n} b_{1j} (b_{1j} + 2a_{1j}) \tag{44}\]
We also divide the middle term in Eq. (44) into two parts depending on whether the $b_{1j} + 2a_{1j}$ terms are positive or not. Let $\mathcal{J}^+$ denote the indices $j$ where $b_{1j} + 2a_{1j} > 0$ and $\mathcal{J}^-$ otherwise. Then, we define $S^+ = \sum_{j \in \mathcal{J}^+} (b_{1j} + 2a_{1j})$ and $S^- = \sum_{j \in \mathcal{J}^-} (b_{1j} + 2a_{1j})$ so that

$$\sum_{j = \sqrt{n}/\sqrt{n} + 1}^{n - \sqrt{n}} b_{1j}(b_{1j} + 2a_{1j}) > b_{1(\sqrt{n} + 1)}(S^+ + S^-) + b_{1(n - \sqrt{n})}(S^-)$$

and therefore, $0 < \frac{1}{\sqrt{n}} \leq |a_{ij} - a_{1j}| < \frac{O(n')}{n}$ for $i \in \{1, 2\}$ and some $j \in \{1, \ldots, n\}$. Hence, $RES(A) = r = \Omega(\log n)$.

**Proposition 5.** For an arbitrary NN representation of size at least 2, pick any positive and negative anchor, $a$ and $b$. Let $X$ and $Y$ be the set of binary vectors closest to $a$ and $b$, respectively. Then, $conv(X) \cap conv(Y) = \emptyset$ where $conv(A)$ denotes the convex hull of a set $A$.

**Proof of Proposition 5.** Let $|X| = k$ and $|Y| = l$. We assume that $X = \{X_1, \ldots, X_k\}$ and $Y = \{Y_1, \ldots, Y_l\}$. Also, let $u$ and $v$ be arbitrary convex combinations for the sets $X$ and $Y$. That is, $u = \sum_{i=1}^{k} \lambda_i X_i$ and $v = \sum_{i=1}^{l} \mu_i Y_i$ where $X_i \in X$, $Y_i \in Y$, $\lambda_i \in [0, 1]$, $\sum_{i=1}^{k} \lambda_i = 1$, $\mu_i \in [0, 1]$, and $\sum_{i=1}^{l} \mu_i = 1$.

We know that $d(a, X_i)^2 < d(b, X_i)^2$ for $i \in \{1, \ldots, k\}$ and $d(a, Y_i)^2 > d(b, Y_i)^2$ for $i \in \{1, \ldots, l\}$. Suppose for contradiction that there is a set of $\lambda$ and $\mu$ so that $u = v$.

$$d(a, u)^2 - d(b, u)^2 = ||a||^2 - ||b||^2 - 2(a - b)^T \sum_{i=1}^{k} \lambda_i X_i$$

$$= \sum_{i=1}^{k} \lambda_i (||a||^2 - 2(a - b)^T X_i - ||b||^2)$$

$$= \sum_{i=1}^{k} \lambda_i (d(a, X_i)^2 - d(b, X_i)^2) < 0$$

where we use $\lambda_i \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$. We do the same for $v$ since $u = v$, to obtain

$$d(a, v)^2 - d(b, v)^2 = \sum_{i=1}^{l} \mu_i (d(a, Y_i)^2 - d(b, Y_i)^2) > 0$$

resulting in a contradiction.

**Proof of Lemma 3.** There are only two possible anchor assignments for symmetric Boolean functions with 3 intervals besides an interval-anchor assignment: Either $a_2$ is assigned to the region $|X| \in [I_1 + 1, I_2]$ and $a_1$ & $a_3$ shares the rest (Case 1) or $a_2$ & $a_3$ shares the same region $|X| \in [I_1 + 1, I_2]$ and $a_1$ is assigned to the rest (Case 2).

**For Case 1:** We pick $X_1$ and $X_2$ assigned either both to $a_1$ or $a_2$ where $|X| = t_1 \in I_1 + 1$ and $|X^2| = t_2 > I_2$. Consider also an integer $t \in [I_1 + 1, I_2]$ independently. This is always possible except the function given in the description. However, for that function, this case implies an interval-anchor assignment as there are unique vectors both for $|X| = 0$ and $|X| = n$. We denote the set of coordinates where $X_1 = X_2 = 1$ by $S_1$, $X_1 = X_2 = 0$ by $S_2$, $X_1 = 0, X_2 = 1$ by $S_3$, and finally, $X_1 = X_2 = 0$ by $S_4$. Clearly, $t_1 = |S_1| + |S_2|$, $t_2 = |S_1| + |S_3| + |S_4| + |S_4| = n$. Consider the following as an example.

$$X_1 = (1, \ldots, 1, 1, \ldots, 1, 0, \ldots, 0, 0, \ldots, 0)$$

$$X_2 = (1, \ldots, 1, 0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0)$$
Note that the example representation given in Eq. (57) and (58) can be assumed without loss of generality by the reordering of indices. Also, \( S_3 \neq \emptyset \) is necessary by the choice of \( t_1, t_2, t_3, t_4 \) and \( t \). It is also clear that \(|S_1| = t_2 - t_1 + |S_2| \geq t - t_1 \) and \(|S_3| = t_2 - |S_1| \geq t - |S_1| \).

It can be easily verified that the convex combination \( X' = \frac{t_2 - t_1}{t_2 - t_4} X_1 + \frac{t_4 - t_1}{t_2 - t_4} X_2 \) lies on the hyperplane \(|X| = t\). We further claim that it is in the convex hull of the set of binary vectors \( X_s \) where \(|X| = t\).

Let \( Y \) be the average of all binary vectors on the hyperplane \(|X| = t\) with \( t \) many 1s at the locations of 1s of \( X_1 \), \( S_1 \) many 0s at the location of 0s of \( X_1 \). That is, we put the remaining \( t - t_1 \) many ones to the indices in \( S_2 \). It is clear that there are \( \binom{t_2}{t_1-1} \) such vectors. The average value in the \( S_3 \) region is \( K = \binom{|S_3| - 1}{t_3 - t_1} / \binom{|S_3|}{t_3 - t_1} = (t - t_1) / |S_3| \).

Similarly, we define \( Z \) as the average of all binary vectors on the hyperplane \(|X| = t\) with \( t \) many 1s for the indices in \( S_1 \) and the \( t - |S_1| \) remaining 1s will be distributed in the region \( S_3 \). The average value in the \( S_3 \) region is \( L = \binom{|S_3| - 1}{t_3 - |S_1|} / \binom{|S_3|}{t_3 - |S_1|} = (t - |S_1|) / |S_3| \).

\[
Y = (1, \ldots, 1, 1, \ldots, 1, K, \ldots, K, 0, \ldots, 0) \quad (59)
\]

\[
Z = (1, \ldots, 1, 0, \ldots, 0, L, \ldots, L, 0, \ldots, 0) \quad (60)
\]

Then, it is easy to verify that \( X' = \frac{t_2 - t_1}{t_2 - t_3} Y + \frac{t_4 - t_1}{t_2 - t_3} Z \) as \( K(t_2 - t) / (t_2 - t_1) + L(t - t_1) / (t_2 - t_1) = (t - t_1) / (t_2 - t_1) \).

For Case 2: When \( a_2 \) & \( a_3 \) share the interval \([I_1 + 1, I_2]\), all the vectors such that \(|X| = t\) for \( t \in [I_1 + 1, I_2] \) cannot be assigned to one of the anchors as this will reduce to Case 1. Therefore, \( a_2 \) & \( a_3 \) must share \(|X| = t\) for any \( t \in [I_1 + 1, I_2] \).

Let \( X_1 \neq X_2 \), and the \( S_i \) be the same as in Case 1 except \(|X_1| = |X_2| = I_1 + 1\). We can choose \( X_1 \) and \( X_2 \) closest to \( a_2 \) or \( a_3 \) respectively. In this case, \( S_2 \) and \( S_3 \) are not empty. We define \( X_3 \) for \( X_1 \& X_2 \) such that \(|X_3| = I_2 + 1 \) and \( X_3 \) has all 1s for the indices in \( S_1 \) and \( S_2 \) except the entry in the first index for \( S_2 \) is zero. Also, we keep the first entry in \( S_3 \) to be zero as well and fill the rest of the indices arbitrarily. Hence, by construction, we have \( I_2 < n - 1 \).

\[
X_3 = (1, \ldots, 1, 0, 1, \ldots, 1, 0, 1, 0, \ldots, 1) \quad (61)
\]

Similarly, by reversing the roles of 0s and 1s, we can obtain \( I_1 > 0 \) condition. Therefore, the construction of \( X_3 \) is possible whenever either \( I_1 > 0 \) or \( I_2 < n - 1 \). The only exception is when \( f(X) = 1 \) (or 0) for \(|X| \in \{0, n\} \) and 0 (or 1) otherwise.

Notice that \( X_3 \) is constructed by using \( S \) depending both on \( a_2 \) and \( a_3 \). In general, \( X_3 \) can be assigned either to \( a_2 \) or \( a_3 \). Assume that it is assigned to \( a_2 \). Otherwise, change \( X_1 \) to \( X_2 \) in the following claims: Let \( Y = X_1 \wedge X_3 \) and \( Z = X_1 \vee X_3 \). Clearly, \(|Y| \leq I_1 - 1 \) and \(|Z| \geq I_2 + 1 \) and they are both assigned to \( a_2 \). We finally claim that \(|Y| / 2 + |Z| / 2 = X_1 / 2 + X_3 / 2 \).

In conclusion, both cases contradict Proposition 5 and an interval-anchor assignment is necessary.

### B. A Counterexample for the Parity-based Extensions for NN Representations of Symmetric Boolean Functions

Let \( f(X) \) be the function given in Eq. (62). For the sake of contradiction, we assume that there exist a Parity-based extension to this symmetric function, namely, \( a_{ij} = a_{ik} \) for any \( i \in \{1, \ldots, 5\} \) and \( j, k \in \{1, \ldots, 8\} \).

<table>
<thead>
<tr>
<th>X</th>
<th>f(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

We can rewrite the necessary and sufficient conditions given in Lemma 2. Also we know that \( a_{ij} + a_{(i-1)j} \) (or \( a_{ij} - a_{(i-1)j} \)) has the same value for all \( j \in \{1, \ldots, 8\} \) and \( i \in \{2, \ldots, 5\} \). Then, we get

\[
I_{i-1}(a_{i-1} - a_{(i-1)j}) < 4(a_{i-1} - a_{(i-1)j})(a_{i+1} + a_{(i-1)j}) < (I_{i-1} + 1)(a_{i+1} - a_{(i-1)j}) \quad (63)
\]

\[
\frac{I_{i-1}}{4} < (a_{i+1} + a_{(i-1)j}) < \frac{I_{i-1} + 1}{4} \quad (64)
\]

Recall that Proposition 2 still applies here. Therefore, \( a_{ij} < a_{j} < a_{j} < a_{j} < a_{j} \) for all \( j \in \{1, \ldots, 8\} \). More explicitly, we have the following system of inequalities

\[
a_{11} < a_{21} < a_{31} < a_{41} < a_{51} \quad (65)
\]

\[
0 < a_{21} + a_{11} < 1 \quad (66)
\]

\[
1 < a_{31} + a_{21} < 2 \quad (67)
\]

\[
6 < a_{41} + a_{31} < 7 \quad (68)
\]

\[
7 < a_{51} + a_{41} < 8 \quad (69)
\]

Firstly, we multiply Eq. (66) and (68) with \(-1\) and sum all Eq. (66),(67),(68), and (69). Secondly, we multiply Eq. (67) with \(-1\) and add it to Eq. (68). Thus,

\[
0 < a_{51} - a_{11} < 1 \quad (70)
\]

\[
1 < a_{41} - a_{21} < 1.5 \quad (71)
\]

which is inconsistent with Eq. (65).

The following is a construction for the function in Eq. (62) using our techniques. We use 2 significant digits to fit the matrix here.

| 17.78 | 2.28 | -5.72 | -21.22 | -1.53 | -1.53 | -1.53 |
| 19.28 | 3.28 | -4.72 | -20.22 | -0.53 | -0.53 | 0.53 |
| 20.28 | 4.78 | -3.72 | -19.22 | 0.47 | 0.47 | 0.47 |
| 21.28 | 5.78 | -2.22 | -18.22 | 1.47 | 1.47 | 1.47 |
| 22.28 | 6.78 | -1.22 | -16.72 | 2.47 | 2.47 | 2.47 |
C. Examples of NN Representations of Symmetric Boolean Functions without an Interval-Anchor Assignment

Lemma 4. Let \( f \) be an \( n \)-input symmetric Boolean function with \( I(f) \leq 4 \). Then, \( NN(f) \geq I(f) \).

Proof. The cases when \( I(f) \in \{1, 2\} \) is trivial. When \( I(f) = 3 \), we can use Proposition 2. Now, assume that 3-anchor NN representations exist for symmetric Boolean functions with \( I(f) = 4 \). Then, the assignment of anchors to the intervals can only belong to two cases: \( a_1 \) is assigned to the first and third intervals where \( o_2 \) & \( a_3 \) shares the binary vectors for the second and fourth intervals, and vice versa. In both cases, there is a structure similar to Case 1 of the proof of Lemma 3. By following similar steps, we can obtain a contradiction to Proposition 5 and conclude that 3-anchor NN representations do not exist for symmetric Boolean functions with \( I(f) = 4 \). □

Here is a 5-input counterexample for the function described in Lemma 3 where the middle interval is shared by two positive anchors.

\[
\begin{array}{c|c}
| X | & f(X) \\
\hline
0 & 0 \\
1 & 1 \\
2 & 1 \\
3 & 1 \\
4 & 1 \\
5 & 0 \\
\end{array}
\]

\[
A' = \begin{bmatrix}
0 & 0.57 & 0.57 & 0.57 & 0.57 \\
0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\
1 & 0.43 & 0.43 & 0.43 & 0.43 \\
\end{bmatrix}
\]

(73)

D. The Relationship Between NN Representations and Linear Threshold Circuits

Another important aspect of NN representations is their place in the circuit class hierarchy. Suppose that we are given an \( m \)-anchor NN representation of a Boolean function. One can construct a linear threshold circuit of depth-3 with \( O(m^2) \) many linear threshold gates [5, 6]. The converse, however, is not known; we cannot obtain an NN representation easily given a logic circuit.

We focus on the COMPARISON (denoted by COMP) function to illustrate the size-resolution trade-off. COMP\((X, Y)\) is a linear threshold function

\[
\text{COMP}(X, Y) = \mathbb{I}\left\{ \sum_{i=1}^{n} 2^{i-1} x_i \geq \sum_{i=1}^{n} 2^{i-1} y_i \right\}
\]

which computes whether an unsigned integer is greater than or equal to another. Here, \( X \) and \( Y \) corresponds to the binary expansions of these integers.

Let OR denote Boolean OR function, AND denote Boolean AND function, and THR denote a linear threshold function.

Lemma 5. Suppose that \( f(X) \) is an \( n \)-input Boolean function with an \( m \)-anchor NN representation with the set of positive (or negative) anchors \( P \) (or \( N \)). Then, there is a logic circuit that can compute \( f(X) \) in the form OR \( \circ \) AND \( \circ \) THR of size \( |P||N| + |P| + 1 \) where \( \circ \) denotes function composition.

Proof. The label of the nearest neighbor can be found by the following formula.

\[
\text{arg min}_i (a_i, X) = \text{arg min}_i |X| - 2(AX)_i + ||a_i||^2_2
\]

(74)

\[
= \text{arg max}_i 2(AX)_i - ||a_i||^2_2
\]

(75)

where \( ||\cdot||_2 \) denotes the Euclidean norm. Let \( P \) (or \( N \)) be the set of positive (or negative) anchors \( \{a_1, \ldots, a_{|P|}\} \) (or, \( \{b_1, \ldots, b_{|N|}\} \)).

![Fig. 4. A sketch of a transformation from an NN representation to a linear threshold circuit of depth 3. The first layer consists of linear threshold gates while the second and third layer is an AND – OR network. The gates with \( w_1 \) inside with \( w_0 \) as the bias is THR\((X)\), the gates with \( \wedge \) inside is AND\((X)\), and the gates with \( \vee \) inside is OR\((X)\).]

The constructive transformation is given in Fig. 4. In the first layer, we simply compare the distances of all individual positive anchors, say \( a_i \), to all negative anchors \( \{b_1, \ldots, b_{|N|}\} \) using Eq. (74). Hence, there are \( |P||N| \) many threshold gates in the first layer. There is an AND gate corresponding to each positive anchor and the top gate is an OR gate, implying that the claimed circuit size is correct.

Assume first that \( f(X) = 1 \). The linear threshold function

\[
\mathbb{I}\left\{ 2a^T X - ||a||^2_2 \geq 2b^T X - ||b||^2_2 \right\}
\]

(76)
evaluates 1 for some \( i \in \{1, \ldots, |P|\} \) and for all \( j \in \{1, \ldots, |N|\} \) because a positive anchor must be closer to \( X \) than any negative anchor. Hence, the output of the corresponding \( i^{th} \) AND gate will be 1 and consequently, the output of the circuit is 1.

Conversely, assume \( f(X) = 0 \). Then, there always exist some \( j \) such that Eq. (76) is 0 for all \( i \) because the nearest neighbor to \( X \) is negative (i.e. in \( N \)). Therefore, all of the second layer AND gates compute 0 so that the output of circuit is 0.

To compute COMP, there is a lower bound on the number of threshold gates independent of the circuit depth given that there is a weight size constraint. Based on the lower bound and Lemma 5, we can obtain a similar bound on the NN complexity.

**Theorem 9 ([10, 16]).** Suppose that \( X = (x_1, \ldots, x_n) \) and \( Y = (y_1, \ldots, y_n) \) are binary vectors. The COMP\((X, Y)\) function can be computed by a linear threshold circuit of size \( \Omega(n/\log nW) \) where \( W \in \mathbb{Z} \) is the maximum weight size.

**Lemma 6.** The number of anchors for an NN representation of an \( n \)-input \( \text{COMP}(X, Y) \) is \( \Omega\left(\sqrt{\frac{n}{\log n + r}}\right) \) where \( r \) is the resolution of the representation.

**Proof.** Suppose that there is a NN representation of the \( \text{COMP}(X, Y) \) such that the number of anchors, say \( m \), is \( o\left(\sqrt{\frac{n}{\log n + r}}\right) \). Then, by Lemma 5, there is a depth-3 threshold circuit of size \( o\left(\sqrt{\frac{n}{\log n + r}}\right) \) computing \( \text{COMP}(X, Y) \). Since \( r = \lceil \log_2 W + 1 \rceil \), this contradicts the lower bound for \( \text{COMP}(X, Y) \) given in Theorem 9. \( \square \)