What makes a neural code convex?

Carina Curto\textsuperscript{1}, Elizabeth Gross\textsuperscript{2}, Jack Jeffries\textsuperscript{3,4}, Katherine Morrison\textsuperscript{5,1}, Mohamed Omar\textsuperscript{6}, Zvi Rosen\textsuperscript{7,1}, Anne Shiu\textsuperscript{8}, and Nora Youngs\textsuperscript{6}
August 1, 2015

\begin{flushleft}
\textsuperscript{1} Department of Mathematics, The Pennsylvania State University, University Park, PA 16802
\textsuperscript{2} Department of Mathematics, San José State University, San José, CA 95192
\textsuperscript{3} Department of Mathematics, University of Utah, Salt Lake City, UT 84112
\textsuperscript{4} Department of Mathematics, University of Michigan, Ann Arbor, MI 48109
\textsuperscript{5} School of Mathematical Sciences, University of Northern Colorado, Greeley, CO 80639
\textsuperscript{6} Department of Mathematics, Harvey Mudd College, Claremont, CA 91711
\textsuperscript{7} Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720
\textsuperscript{8} Department of Mathematics, Texas A&M University, College Station, TX 77843
\end{flushleft}

Abstract

Neural codes allow the brain to represent, process, and store information about the world. Combinatorial codes, comprised of binary patterns of neural activity, encode information via the collective behavior of populations of neurons. A code is called \emph{convex} if its codewords correspond to regions defined by an arrangement of convex open sets in Euclidean space. Convex codes have been observed experimentally in many brain areas, including sensory cortices and the hippocampus, where neurons exhibit convex receptive fields. What makes a neural code convex? That is, how can we tell from the intrinsic structure of a code if there exists a corresponding arrangement of convex open sets? Using tools from combinatorics and commutative algebra, we uncover a variety of signatures of convex and non-convex codes. In many cases, these features are sufficient to determine convexity, and reveal bounds on the minimal dimension of the underlying Euclidean space.

Contents

1 Introduction ................................. 2
  1.1 Convex neural codes .......................... 2
  1.2 Preliminaries ............................... 3
  1.3 The neural ideal ............................ 5
  1.4 Summary of main results .................... 6

2 Local obstructions to convexity .................. 10
  2.1 Receptive field relationships ................ 10
  2.2 Local obstructions .......................... 11
  2.3 Link lemmas and the proof of Theorem 1.6 11

3 Computing mandatory codewords algebraically .... 14
  3.1 Alexander duality and Hochster’s formula 14
  3.2 Using free resolutions to compute \( M_H(\Delta) \) 15

4 Combinatorial signatures of convex codes ......... 16
  4.1 Classification of convex codes on \( n \leq 4 \) neurons 16
  4.2 Codes with non-overlapping maximal codewords 21
  4.3 Linear codes and convexity .................. 21

5 Algebraic signatures of convex and non-convex codes 23
  5.1 The canonical form \( \text{CF}(J_C) \) vs. receptive field relationships \( \text{RF}(C) \) 23
1 Introduction

1.1 Convex neural codes

Cracking the neural code is one of the central challenges of neuroscience. Typically, this has been understood as finding the relationship between the activity of neurons and the stimuli they represent. More generally, neural codes must also reflect relationships between stimuli, such as proximity between locations in an environment [1]. Convex codes, comprised of activity patterns for neurons with classical receptive fields, may be the brain’s solution to this problem. These codes have been observed experimentally in many brain areas, including sensory cortices and the hippocampus. Hubel and Wiesel’s discovery in 1959 of orientation-tuned neurons in the primary visual cortex was perhaps the first example of convex coding in the brain [2]. This was followed by O’Keefe’s discovery of hippocampal place cells in 1971 [3], showing that convex codes are also used in the brain’s representation of space. Both discoveries were groundbreaking for neuroscience, and were later recognized with Nobel Prizes in 1981 [4] and 2014 [5], respectively.

Our motivating example of a convex code is, in fact, the hippocampal place cell code. A place cell is a neuron that acts as a position sensor, exhibiting a high firing rate when the animal’s location lies inside the cell’s preferred region of the environment – its place field. Figure 1 displays the place fields of four place cells recorded while a rat explored a two-dimensional environment. Each place field is an approximately convex subset of \( \mathbb{R}^2 \). Taken together, the set of all activity patterns that can arise in a population of place cells comprise a convex code for the animal’s position in space.

![Figure 1: Place fields of four CA1 pyramidal neurons (place cells) in rat hippocampus, recorded while the animal explored a 1.5m \( \times \) 1.5m square box environment. Red areas correspond to regions of space where the corresponding place cells exhibited high firing rates, while dark blue denotes near-zero activity. Place fields were computed from data provided by the Pastalkova lab, as described in [6].](image)

Like the above example, the definition of a convex code is extrinsic: a code is convex if it can be realized by an arrangement of convex open sets in some Euclidean space. How can we characterize convex codes intrinsically? In other words, in terms of neural activity patterns alone, what makes a neural code convex? If a code is not convex, how can we prove this? If a code is convex, what is the minimal dimension needed for the corresponding open sets? Identifying intrinsic features of convex and non-convex codes will enable us to infer coding properties from population recordings of neural
activity, without needing a priori knowledge of the stimuli being encoded. Understanding the structure of convex codes is also essential to uncovering the basic principles of how neural networks are organized in order to learn, store, and process information.

In this work we build on mathematical ideas first introduced in [7], including the neural ideal $J_\mathcal{C}$ and its canonical form $\text{CF}(J_\mathcal{C})$, in order to tackle questions of convexity and dimension in combinatorial neural codes. After reviewing the necessary background in Sections 1.2 and 1.3 we give an overview of the main results in Section 1.4. In Section 2, we introduce local obstructions, our main tool for proving that a code is not convex. We also prove a core theorem characterizing codes without local obstructions. In Section 3 we present an algebraic method using free resolutions of monomial ideals that enables us to quickly detect non-convex codes based on missing codewords. In Section 4, we classify all convex and non-convex codes on $n \leq 4$ neurons and find special conditions guaranteeing that a code is convex. In Section 5, we find algebraic signatures of convex and non-convex codes using the canonical form of neural ideals. Finally, in Section 6 we explore dimension bounds for convex codes stemming from Helly’s theorem, the Fractional Helly theorem, and $d$-representability of simplicial complexes.

1.2 Preliminaries

Here we briefly review some frequently used definitions. See [7, Section 2] for additional details.

A binary pattern on $n$ neurons is a string of 0s and 1s of length $n$, with a 1 for each active neuron and a 0 denoting silence; equivalently, it is a subset of (active) neurons

$$\sigma \subseteq [n] \overset{\text{def}}{=} \{1, \ldots, n\}.$$  

We will abuse notation and consider 0/1 strings of length $n$ and subsets of $[n]$ interchangeably. For example, 1101 and 0100 are also denoted $\{1, 2, 4\}$ and $\{2\}$, respectively.

A neural code on $n$ neurons is a collection of binary patterns $\mathcal{C} \subseteq 2^{[n]}$. In other words, it is a binary code of length $n$ where we interpret each binary digit as the “on” or “off” state of a neuron. This type of code is also known in the neuroscience literature as a combinatorial code [1]. The elements of a code are called codewords. For convenience, we will always assume a neural code $\mathcal{C}$ includes the all-zeros codeword, $00\cdots0 \in \mathcal{C}$; the presence or absence of this codeword has no effect on the code’s convexity or minimal embedding dimension (see Definition 1.1 below).

To any code $\mathcal{C}$, we can associate a simplicial complex $\Delta(\mathcal{C})$:

$$\Delta(\mathcal{C}) \overset{\text{def}}{=} \{\sigma \subseteq [n] \mid \sigma \subseteq c \text{ for some } c \in \mathcal{C}\}.$$  

This is the smallest abstract simplicial complex on $[n]$ that contains all elements of $\mathcal{C}$. When referring to the subsets that comprise a simplicial complex, we will use the terms elements and faces interchangeably. The dimension of a face $\sigma \in \Delta$ is defined to be $|\sigma| - 1$. Faces that are maximal under inclusion are referred to as facets. Note that any facet of $\Delta(\mathcal{C})$ must be a codeword of $\mathcal{C}$, and so these facets are also referred to as maximal codewords.

The hollow simplex on $n$ vertices is the simplicial complex containing all proper subsets of $[n]$ – i.e., the simplex missing only the top-dimensional face, $[n]$. The restriction of $\Delta$ to $\sigma$ is the simplicial complex

$$\Delta|_{\sigma} \overset{\text{def}}{=} \{\omega \in \Delta \mid \omega \subseteq \sigma\}.$$  

Another important simplicial complex we will work with is the link, $\text{Lk}_{\sigma}(\Delta)$\footnote{It is more common to write $\text{Lk}_{\Delta}(\sigma)$ or $\text{link}_{\Delta}(\sigma)$, instead of $\text{Lk}_{\sigma}(\Delta)$ (see, for example, [3]). However, because we will often fix $\sigma$ and consider its link inside different simplicial complexes, such as $\Delta|_{\sigma \cup \tau}$, it is more convenient to put $\sigma$ in the subscript.} For any $\sigma \in \Delta$, the link of $\sigma$ inside $\Delta$ is

$$\text{Lk}_{\sigma}(\Delta) \overset{\text{def}}{=} \{\omega \in \Delta \mid \sigma \cap \omega = \emptyset \text{ and } \sigma \cup \omega \in \Delta\}.$$
Let $X$ be a topological space. A collection of open sets $\mathcal{U} = \{U_1, \ldots, U_n\}$, where each $U_i \subset X$, is called an open cover. If every non-empty intersection,

$$U_\sigma \overset{\text{def}}{=} \bigcap_{i \in \sigma} U_i \quad \text{for} \; \sigma \subseteq [n],$$

is contractible\(^2\) then we say that $\mathcal{U}$ is a good cover. Given an open cover $\mathcal{U}$, the code of the cover is the neural code

$$\mathcal{C}(\mathcal{U}) \overset{\text{def}}{=} \{ \sigma \subseteq [n] \mid U_\sigma \setminus \bigcup_{j \in [n] \setminus \sigma} U_j \neq \emptyset \}.$$ 

Each codeword in $\mathcal{C}(\mathcal{U})$ corresponds to a region that is defined by the intersections of open sets in $\mathcal{U}$. By convention, the empty intersection $U_\emptyset = \bigcap_{i \in \emptyset} U_i$ equals $X$, so that $\emptyset \in \mathcal{C}(\mathcal{U})$ if and only if $\bigcup_{i \in [n]} U_i \subseteq X$. We also assume $\bigcup_{i \in [n]} U_i \subseteq X$, so that $00\cdots0 \in \mathcal{C}$ (i.e., $\emptyset \in \mathcal{C}$), in agreement with our convention.

Note that there always exists $\mathcal{U} = \{U_1, \ldots, U_n\}$ such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$ \([7, \text{Lemma 2.1}]\), but it may be impossible to choose the $U_i$s to all be convex. We thus have the following definitions:

**Definition 1.1.** Let $\mathcal{C}$ be a neural code on $n$ neurons. If there exists an open cover $\mathcal{U} = \{U_1, \ldots, U_n\}$ such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$ and the $U_i$s are all convex subsets of $\mathbb{R}^d$, then we say that $\mathcal{C}$ is convex. The smallest $d$ such that this is possible is called the minimal embedding dimension, $d(\mathcal{C})$.

If a code $\mathcal{C}$ is not convex, our convention is to set $d(\mathcal{C}) = 1$. The following example is of a convex code with minimal embedding dimension $d(\mathcal{C}) = 2$.

**Example 1.2.** Consider the open cover $\mathcal{U}$ illustrated in Figure 2a. The corresponding code, $\mathcal{C} = \mathcal{C}(\mathcal{U})$, has 10 codewords. $\mathcal{C}$ is a convex code with $d(\mathcal{C}) = 2$. The simplicial complex $\Delta(\mathcal{C})$ is depicted in Figure 2b.

Figure 2: (a) An arrangement $\mathcal{U} = \{U_1, \ldots, U_4\}$ of convex open sets. Black dots mark regions corresponding to distinct codewords. From left to right, the codewords are 0000, 1000, 1100, 1010, 1110, 0110, 0010, 0111, 0011, and 0001. (b) The simplicial complex $\Delta(\mathcal{C})$ for the code $\mathcal{C} = \mathcal{C}(\mathcal{U})$ defined in (a). The two facets, 123 and 234, correspond to the two maximal codewords, 1110 and 0111, respectively.

It is important to note that $\mathcal{C}(\mathcal{U})$ is not the same as the nerve of the cover:

$$\mathcal{N}(\mathcal{U}) \overset{\text{def}}{=} \{ \sigma \subseteq [n] \mid U_\sigma \neq \emptyset \}.$$ 

In fact, $\mathcal{N}(\mathcal{U}) = \Delta(\mathcal{C}(\mathcal{U}))$, and $\mathcal{C}(\mathcal{U}) \subseteq \mathcal{N}(\mathcal{U})$. The nerve of any cover $\mathcal{U}$ such that $\mathcal{C} = \mathcal{C}(\mathcal{U})$ can thus be recovered directly from the code as $\Delta(\mathcal{C})$, without reference to a specific cover. The celebrated Nerve lemma tells us that $\mathcal{N}(\mathcal{U})$ carries a surprising amount of topological information about the underlying space covered by the sets in $\mathcal{U}$. The following statement is a direct consequence of \([9, \text{Corollary 4G.3}]\).

\(^2\)A set is contractible if it is homotopy-equivalent to a point.\[^9\]
Lemma 1.3 (Nerve lemma). If \( \mathcal{U} \) is a good cover, then the covered space \( Y = \bigcup_{i=1}^{n} U_i \) is homotopy-equivalent to \( \mathcal{N}(\mathcal{U}) \). In particular, \( Y \) and \( \mathcal{N}(\mathcal{U}) \) have exactly the same homology groups.

Because an open cover comprised of convex sets is always a good cover, the Nerve lemma tells us that if \( \mathcal{C} = \mathcal{C}(\mathcal{U}) \) for a collection of convex open sets \( \mathcal{U} \), then \( \Delta(\mathcal{C}) \) must match the homotopy type of \( \bigcup_{i=1}^{n} U_i \). This fact was previously exploited to extract topological information about the represented space from hippocampal place cell activity \[10\].

The code \( \mathcal{C}(\mathcal{U}) \), however, contains additional information about \( \mathcal{U} \) that is not captured by the nerve alone (see \[7, Section 2.3.2\]). In particular, a non-empty intersection \( U_\sigma \) may itself be covered by some of the other open sets, so that \( U_\sigma \subseteq \bigcup_{i \in \tau} U_i \), but this information is not present in \( \mathcal{N}(\mathcal{U}) \). For example, in Figure 2a we have \( U_2 \subset U_1 \cup U_3 \) and \( U_2 \cap U_4 \subset U_3 \). Note that if \( U_\sigma \subseteq \bigcup_{i \in \tau} U_i \), the collection of open sets \( \{U_\sigma \cap U_i\}_{i \in \tau} \) forms an open cover of \( U_\sigma \). The nerve of this cover is related to the nerve of the original cover via the link: if \( \Delta = \mathcal{N}(\mathcal{U}) \), then

\[
\mathcal{N}(\{U_\sigma \cap U_i\}_{i \in \tau}) = \text{Lk}_\sigma(\Delta|_{\sigma \cup \tau}).
\]

We will make use of this fact when we discuss local obstructions in Section 2.2.

1.3 The neural ideal

To capture the full combinatorial data of \( \mathcal{C} \) algebraically, we turn to two ideals, \( I_\mathcal{C} \) and \( J_\mathcal{C} \), that were first introduced in \[7\]. Here we give only a brief review of the most relevant definitions; for more details, see \[7, Section 3\].

Consider the polynomial ring \( \mathbb{F}_2[x_1, \ldots, x_n] \), whose elements can be thought of as functions on binary patterns of length \( n \), taking values in the finite field \( \mathbb{F}_2 \). To evaluate a polynomial \( f \in \mathbb{F}_2[x_1, \ldots, x_n] \) on a binary pattern, we simply replace each indeterminate \( x_i \) with the 0/1 value of the \( i \)th neuron. For example, the polynomial \( f = x_1x_3(1 - x_2) \in \mathbb{F}_2[x_1, \ldots, x_4] \) can be evaluated on the binary patterns 1011 and 1100 to obtain \( f(1011) = 1 \) and \( f(1100) = 0 \).

Given a code \( \mathcal{C} \subseteq \mathbb{F}_2^n \), perhaps the most natural ideal to consider is the set of all polynomials that vanish on all codewords:

\[
I_\mathcal{C} \overset{\text{def}}{=} \{ f \in \mathbb{F}_2[x_1, \ldots, x_n] \mid f(c) = 0 \text{ for all } c \in \mathcal{C} \}.
\]

Note that \( \mathcal{B} = \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle \) is automatically contained in \( I_\mathcal{C} \) because of the binary nature of codewords. In fact, \( I_\mathcal{C} \) can be expressed as the sum \( I_\mathcal{C} = J_\mathcal{C} + \mathcal{B} \), where \( J_\mathcal{C} \) is the \textit{neural ideal}:

\[
J_\mathcal{C} \overset{\text{def}}{=} \langle \chi_v \mid v \in \mathbb{F}_2^n \setminus \mathcal{C} \rangle,
\]

generated by a subset of the indicator functions,

\[
\chi_v \overset{\text{def}}{=} \prod_{\{i|v_i=1\}} x_i \prod_{\{j|v_j=0\}} (1 - x_j).
\]

The functions \( \chi_v \) are examples of \textit{pseudo-monomials}, which are polynomials \( f \in \mathbb{F}_2[x_1, \ldots, x_n] \) that can be written in the form

\[
f = x_\sigma \prod_{j \in \tau} (1 - x_j),
\]

where \( x_\sigma \overset{\text{def}}{=} \prod_{i \in \sigma} x_i \) and \( \sigma, \tau \subseteq [n] \) with \( \sigma \cap \tau = \emptyset \). Pseudo-monomials come in three types:

- Type 1: \( x_\sigma \), for \( \sigma \neq \emptyset \),
- Type 2: \( x_\sigma \prod_{i \in \tau} (1 - x_i) \), for \( \sigma \neq \emptyset, \sigma \cap \tau = \emptyset \), and
• Type 3: $\prod_{i \in \tau}(1 - x_i)$, for $\tau \neq \emptyset$.

For any ideal $J \subseteq \mathbb{F}_2[x_1, \ldots, x_n]$, a pseudo-monomial $f \in J$ is called minimal if there does not exist another pseudo-monomial $g \in J$ with $\deg(g) < \deg(f)$ such that $f = hg$ for some $h \in \mathbb{F}_2[x_1, \ldots, x_n]$.

If $J$ is an ideal generated by a set of pseudo-monomials, we define the canonical form of $J$ to be the set of all minimal pseudo-monomials of $J$:

$$\text{CF}(J) \overset{\text{def}}{=} \{ f \in J \mid f \text{ is a minimal pseudo-monomial} \}.$$  

In particular, for any neural code $\mathcal{C}$ the neural ideal $J_{\mathcal{C}}$ is generated by pseudo-monomials. We will make frequent use of its canonical form, $\text{CF}(J_{\mathcal{C}})$. Since we assume $00 \cdots 0 \in \mathcal{C}$, $J_{\mathcal{C}}$ and $\text{CF}(J_{\mathcal{C}})$ contain no Type 3 pseudo-monomials. We denote the Type 1 and Type 2 pseudo-monomials of $\text{CF}(J_{\mathcal{C}})$ by $\text{CF}^1(J_{\mathcal{C}})$ and $\text{CF}^2(J_{\mathcal{C}})$, respectively, so that:

$$\text{CF}(J_{\mathcal{C}}) = \text{CF}^1(J_{\mathcal{C}}) \cup \text{CF}^2(J_{\mathcal{C}}).$$

Note that the canonical form $\text{CF}(J_{\mathcal{C}})$ can be computed algorithmically, starting from the code $\mathcal{C}$. In [7, Section 4.5], an algorithm was described that used the primary decomposition of pseudo-monomial ideals. This algorithm has since been improved [11], and software for computing $\text{CF}(J_{\mathcal{C}})$ has been made publicly available [12].

**Example 1.4 (Example 1.2 continued).** Recall the code $\mathcal{C} = \mathcal{C}(\mathcal{U})$ from Example 1.2. For this code, there are six non-codewords: 0100, 1001, 0101, 1101, 1011, and 1111. The neural ideal $J_{\mathcal{C}}$ is thus generated by the corresponding pseudo-monomials $\chi_v$:

$$J_{\mathcal{C}} = \langle x_2(1 - x_1)(1 - x_3)(1 - x_4), \ x_1 x_4(1 - x_2)(1 - x_3), \ x_2 x_4(1 - x_1)(1 - x_3), \ x_1 x_2 x_4(1 - x_3), \ x_1 x_3 x_4(1 - x_2), \ x_1 x_2 x_3 x_4 \rangle.$$  

The canonical form is $\text{CF}(J_{\mathcal{C}}) = \{ x_1 x_4, x_2(1 - x_1)(1 - x_3), x_2 x_4(1 - x_3) \}$, with $\text{CF}^1(J_{\mathcal{C}}) = \{ x_1 x_4 \}$ and $\text{CF}^2(J_{\mathcal{C}}) = \{ x_2(1 - x_1)(1 - x_3), x_2 x_4(1 - x_3) \}$.

Finally, as was shown in [7, Section 4.4], the neural ideal can be viewed as a generalization of the Stanley-Reisner ideal associated to a simplicial complex $\Delta$:

$$I_{\Delta} \overset{\text{def}}{=} \{ x_\sigma \mid \sigma \notin \Delta \}.$$  

In fact, the elements of $\text{CF}^1(J_{\mathcal{C}})$ generate the Stanley-Reisner ideal of $\Delta(\mathcal{C})$, so that $I_{\Delta(\mathcal{C})} = \langle \text{CF}^1(J_{\mathcal{C}}) \rangle$.

### 1.4 Summary of main results

To prove that a neural code is convex, it suffices to exhibit a convex realization. Our strategy for proving that a code is not convex is to show that it has a local obstruction to convexity. A local obstruction arises when the combinatorial data of the code $\mathcal{C}$ dictates that any convex realization as $\mathcal{C} = \mathcal{C}(\mathcal{U})$ would have a cover $\mathcal{U}$ that violates the Nerve lemma, and is thus not a good cover (contradicting convexity). We will define local obstructions more precisely in Section 2.

We say that a code is locally good if it has no local obstructions. The following lemma is a direct consequence of Lemma 2.4 in Section 2.2.

**Lemma 1.5.** If $\mathcal{C}$ is a convex code, then $\mathcal{C}$ is locally good.

This was also observed in [13], using slightly different language. Although we previously conjectured that the converse also holds, this appears not to be true. In recent work [14], it was shown that the code $\mathcal{C} = \{ 2345, 123, 134, 145, 13, 14, 23, 34, 45, 3, 4 \}$ is not convex – despite having no local obstructions.
Our first result states that $C$ is locally good if and only if $\sigma \in C$ for any non-contractible link $Lk_{\sigma}(\Delta(C))$. We denote the elements of a simplicial complex $\Delta$ exhibiting non-contractible links as

$$\mathcal{M}(\Delta) \overset{\text{def}}{=} \{ \sigma \in \Delta \mid Lk_{\sigma}(\Delta) \text{ non-contractible} \}.$$  

Note that $\mathcal{M}(\Delta)$ always contains all facets of $\Delta$, since the link of a facet is the empty set, which by convention is non-contractible. In fact, all elements of $\mathcal{M}(\Delta)$ are intersections of facets.

**Theorem 1.6.** A code $C$ is locally good if and only if $\mathcal{M}(\Delta(C)) \subseteq C$. If $\sigma \in \mathcal{M}(\Delta)$, then $\sigma$ is an intersection of facets of $\Delta$.

The proof is given in Section 2.3. The elements of $\mathcal{M}(\Delta(C))$ may be thought of as “mandatory” codewords: they must be included in order for a code $C$ with simplicial complex $\Delta(C)$ to be convex. The theorem also tells us that all mandatory codewords correspond to intersections of facets. To guarantee that a code is locally good, irrespective of the topology of the links $Lk_{\sigma}(\Delta)$, it thus suffices for the code to contain all intersections of maximal codewords, ensuring that $\mathcal{M}(\Delta(C)) \subseteq C$. We refer to such codes as max intersection-complete codes, and write “$C$ is max $\cap$-complete.” Strengthening this definition we obtain intersection-complete ($\cap$-complete) codes, which have the property that for any pair of codewords $\omega_1, \omega_2 \in C$, we must have $\omega_1 \cap \omega_2 \in C$. We will provide an algebraic characterization of $\cap$-complete codes in Section 5.2.

It was recently proven that all $\cap$-complete codes are convex [15]. This is not, however, a necessary condition for convexity. The following example shows that a code need not be even max $\cap$-complete in order to be convex.

**Example 1.7.** Consider the simplicial complex $\Delta$ shown in Figure 3a. The facets are 123, 134, and 145, and their intersections yield the faces 1, 13, and 14. Note that the links $Lk_{13}(\Delta) = \{2, 4\}$ and $Lk_{14}(\Delta) = \{3, 5\}$ are non-contractible, while $Lk_1(\Delta)$ is contractible (see Figure 3b). The set of faces with non-contractible links is thus $\mathcal{M}(\Delta) = \{13, 14, 123, 134, 145\}$.

The code $C = \{1, 13, 14, 45, 123, 134, 145\}$, with $\Delta(C) = \Delta$, is max $\cap$-complete, and thus locally good, but it is not $\cap$-complete because it is missing $4 = (14) \cap (45)$. Now consider the code $\hat{C} = \Delta \setminus \{1\}$, missing only the vertex 1 of $\Delta$. $\hat{C}$ is not max $\cap$-complete. However, $\mathcal{M}(\Delta(\hat{C})) \subseteq C$, and so by Theorem 1.6 we know that $\hat{C}$ is locally good. $\hat{C}$ is also, in fact, convex (see Figure 3c).

![Figure 3](image-url)

Figure 3: (a) A simplicial complex $\Delta$. The vertex 1 is an intersection of facets, but $1 \notin \mathcal{M}(\Delta)$ because $Lk_1(\Delta)$ is contractible. (b) $Lk_1(\Delta)$. (c) A convex realization of the code $\hat{C}$ from Example 1.7.

While all elements of $\mathcal{M}(\Delta)$ may be difficult to compute in general, we denote the subset of faces with non-contractible links that can be detected via homology as

$$\mathcal{M}_H(\Delta) \overset{\text{def}}{=} \{ \sigma \in \Delta \mid \dim \tilde{H}_i(Lk_{\sigma}(\Delta)) > 0 \text{ for some } i \}.$$  

Clearly, $\mathcal{M}_H(\Delta) \subseteq \mathcal{M}(\Delta)$, so if any element of $\mathcal{M}_H(\Delta(C))$ is missing from the code $C$, then $C$ is not locally good and hence non-convex. We will describe in Section 3.2 how to compute $\mathcal{M}_H(\Delta)$ using existing computational algebra software.

Collecting the above consequences of Theorem 1.6 together with several additional results, we obtain the following theorem.

---

3We omit the coefficients for homology, as they do not affect the results. The $\tilde{H}_i$ denote reduced homology groups.
Theorem 1.8. Let $\mathcal{C}$ be a code on $n$ neurons. Each of the combinatorial signatures in rows C-1, ..., C-9 in Table 1 implies the corresponding property of $\mathcal{C}$. In rows C-1 and C-5, the implications are bidirectional.

<table>
<thead>
<tr>
<th>Combinatorial signature of $\mathcal{C}$</th>
<th>Property of $\mathcal{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-1 $\mathcal{M}(\Delta(\mathcal{C})) \subseteq \mathcal{C}$</td>
<td>⇔ locally good</td>
</tr>
<tr>
<td>C-2 $\mathcal{M}_H(\Delta(\mathcal{C})) \not\subseteq \mathcal{C}$</td>
<td>⇒ non-convex</td>
</tr>
<tr>
<td>C-3 $\text{Lk}_\sigma(\Delta(\mathcal{C}))$ is non-contractible for some $\sigma \in \Delta(\mathcal{C}) \setminus \mathcal{C}$</td>
<td>⇒ non-convex</td>
</tr>
<tr>
<td>C-4 $\mathcal{C}$ is max $\cap$-complete</td>
<td>⇒ locally good</td>
</tr>
<tr>
<td>C-5 $\mathcal{C}$ is max $\cap$-complete, and $n \leq 4$</td>
<td>⇔ convex (for $n \leq 4$)</td>
</tr>
<tr>
<td>C-6 $\mathcal{C}$ is $\cap$-complete</td>
<td>⇒ convex</td>
</tr>
<tr>
<td>C-7 $\mathcal{C} = \Delta(\mathcal{C})$ (for $n &gt; 1$)</td>
<td>⇒ convex, $d(\mathcal{C}) \leq n - 1$</td>
</tr>
<tr>
<td>C-8 $1 \cdots 11 \in \mathcal{C}$, or all facets of $\Delta(\mathcal{C})$ are disjoint</td>
<td>⇒ convex, $d(\mathcal{C}) \leq 2$</td>
</tr>
<tr>
<td>C-9 $\mathcal{C}$ is max $\cap$-complete and also a linear code</td>
<td>⇒ convex, $d(\mathcal{C}) \leq 2$</td>
</tr>
</tbody>
</table>

Table 1: Combinatorial signatures of convex, non-convex, and locally good codes.

Rows C-1, C-2, C-3, and C-4 of Table 1 are all immediate consequences of Theorem 1.6. C-5 corresponds to Theorem 1.4, which is presented in Section 4.1, together with a classification of convex codes for $n \leq 4$. C-6 was recently proven in [15], while C-7 follows from a result of Tancer, which we summarize in Section 6.1. C-8 corresponds to Proposition 4.6 of Section 4.2, and C-9 corresponds to Proposition 4.9 of Section 4.3.

Although C-4 of Theorem 1.8 tells us only that max $\cap$-complete codes are locally good, C-5, C-6, C-7, C-8, and C-9 all provide special cases of max $\cap$-complete codes that are, in fact, also convex. This motivates the following conjecture:

Conjecture 1.9. If $\mathcal{C}$ is a max $\cap$-complete code, then $\mathcal{C}$ is convex.

Note that the converse to this conjecture does not hold for $n \geq 5$. In particular, the code $\hat{\mathcal{C}}$ of Example 1.7 is convex, despite the fact that it is not max $\cap$-complete.

The next theorem provides algebraic signatures that allow us to determine whether or not a given code is locally good.

Theorem 1.10. Let $J_{\mathcal{C}}$ denote the neural ideal associated to the code $\mathcal{C}$, and $\text{CF}(J_{\mathcal{C}}) = \text{CF}^1(J_{\mathcal{C}}) \cup \text{CF}^2(J_{\mathcal{C}})$ its canonical form. For any $x_\sigma \prod_{i \in \tau} (1 - x_i) \in \text{CF}^2(J_{\mathcal{C}})$, if $\text{Lk}_\sigma(\Delta|_{\sigma \cup \tau})$ is not contractible then $\mathcal{C}$ is not a convex code. In particular, if $J_{\mathcal{C}}$ exhibits any of the algebraic signatures in rows A-1, A-2, or A-3 of Table 2, then $\mathcal{C}$ is not convex. If, alternatively, A-4 or A-5 are satisfied, then $\mathcal{C}$ is locally good. Codes that satisfy signature A-5 are $\cap$-complete, and hence convex.

The proof is given in Section 5.3. The following examples illustrate the use of the algebraic signatures from Theorem 1.10.

Example 1.11. Consider the codes $\mathcal{C}_1, \ldots, \mathcal{C}_5$ in Table 3, together with their canonical forms. The code $\mathcal{C}_1$ is not convex because it exhibits signature A-1 for, e.g., $\sigma = \{1\}$, $i = 2$, and $j = 3$. We can see that $\mathcal{C}_2$ is also not convex, due to signature A-2: for $\sigma = \{1\}$ and $\tau = \{2, 3, 4, 5\}$, the edges of the graph $G_{\mathcal{C}_2}(\sigma, \tau)$ are precisely (23) and (45), so the graph is disconnected. In code $\mathcal{C}_3$, $\sigma = \{1\}$ and $\tau = \{2, 3, 4\}$ yields signature A-3, and so $\mathcal{C}_3$ is also not convex. For $\mathcal{C}_4$, since $\text{CF}^1(J_{\mathcal{C}_4})$ is empty, $x_\sigma x_\tau \notin J_{\mathcal{C}_4}$ for $\sigma = \{1\}$ and $\tau = \{2, 3\}$ and for $\sigma = \{2, 3\}$ and $\tau = \{1\}$, so A-4 applies. We can thus conclude that $\mathcal{C}_4$ is locally good. In $\mathcal{C}_5$, all the Type 2 relations $x_\sigma \prod_{i \in \tau} (1 - x_i)$ satisfy $|\tau| = 1$, so signature A-5 is satisfied. $\mathcal{C}_5$ is thus $\cap$-complete, and hence convex.
the simplicial complex data in bounds on the minimal embedding dimension for max non-local obstructions to convexity, and how can we detect them? Can non-local obstructions arise to improve the bounds on Table 5 in Section 4.1). The problem of how to use this additional information about a code in order the presence or absence of specific codewords can a

<table>
<thead>
<tr>
<th>Property of $\mathcal{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{A}$-1 $\exists x_\sigma (1-x_i)(1-x_j) \in \text{CF}^2(\mathcal{J}<em>C)$ s.t. $x</em>\sigma x_i x_j \in \mathcal{J}_C$ $\Rightarrow$ non-convex</td>
</tr>
<tr>
<td>$\mathcal{A}$-2 $\exists x_\sigma \prod_{i \in \tau} (1-x_i) \in \text{CF}^2(\mathcal{J}<em>C)$ s.t. $G</em>{\mathcal{C}}(\sigma, \tau)$ is disconnected $\Rightarrow$ non-convex</td>
</tr>
<tr>
<td>$\mathcal{A}$-3 $\exists x_\sigma \prod_{i \in \tau} (1-x_i) \in \text{CF}^2(\mathcal{J}<em>C)$ s.t. $x</em>\sigma x_i x_\tau \in \text{CF}^1(\mathcal{J}_C)$ $\Rightarrow$ non-convex</td>
</tr>
<tr>
<td>$\mathcal{A}$-4 $\forall x_\sigma \prod_{i \in \tau} (1-x_i) \in \text{CF}^2(\mathcal{J}<em>C)$, $x</em>\sigma x_\tau \notin \mathcal{J}_C$ $\Rightarrow$ locally good</td>
</tr>
<tr>
<td>$\mathcal{A}$-5 $\forall x_\sigma \prod_{i \in \tau} (1-x_i) \in \text{CF}^2(\mathcal{J}_C)$, $</td>
</tr>
</tbody>
</table>

Table 2: Algebraic signatures of convex, non-convex, and locally good codes. $G_{\mathcal{C}}(\sigma, \tau)$ is the simple graph on vertex set $\tau$ with edges $\{(ij) \in \tau \times \tau \mid x_\sigma x_i x_j \notin \mathcal{J}_C\}$.

<table>
<thead>
<tr>
<th>Code</th>
<th>$\text{CF}^1(\mathcal{J}_C)$</th>
<th>$\text{CF}^2(\mathcal{J}_C)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{C}_1 = 000, 110, 101, 011$</td>
<td>$x_1x_2x_3$</td>
<td>${x_i(1-x_j)(1-x_k) \mid i, j, k = 1, 2, 3}$</td>
</tr>
<tr>
<td>$\mathcal{C}_2 = 000000, 11100, 10011$</td>
<td>$x_1x_2x_3, x_1x_2x_5$</td>
<td>${x_i(1-x_j)(1-x_k)(1-x_{i_5})(1-x_{i_4})(1-x_{i_3}) \mid i_1 \ldots i_5 \in [5]}$</td>
</tr>
<tr>
<td>$01110, 00111$, and all codewords of weight 2</td>
<td>$x_1x_2x_3, x_1x_2x_5, x_2x_3x_5$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{C}_3 = 0000, 1110, 1011$</td>
<td>$x_1x_2x_3x_4$</td>
<td>${x_i(1-x_j)(1-x_k) \mid i, j = 2, 3, 4}$</td>
</tr>
<tr>
<td>$1110, 10011, 1010, 1001$</td>
<td></td>
<td>$x_1(1-x_2)(1-x_3), x_2x_3(1-x_1)$</td>
</tr>
<tr>
<td>$\mathcal{C}_4 = 000, 010, 001, 110, 101, 111$</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{C}_5 = 000000, 11110, 10111, 01111$</td>
<td>$x_1x_2x_5$</td>
<td>${x_i(1-x_j) \mid i \in [5]; j = 3, 4}$</td>
</tr>
<tr>
<td>$10110, 01111, 00111, 00110$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The codes in Example 1.11 and their canonical forms. Note that the indices that appear in a single element of $\text{CF}(\mathcal{J}_C)$ must always be distinct. The weight of a codeword $\sigma \in \mathcal{C}$ is the size $|\sigma|$, and equal to the number of 1s in the corresponding binary string.

Finally, we provide a new lower bound on the minimal embedding dimension $d(\mathcal{C})$. In [7], $\text{CF}(\mathcal{J}_C)$ was used together with Helly’s theorem in order to obtain a coarse bound on $d(\mathcal{C})$:

$$d(\mathcal{C}) \geq \max_{x_\sigma \in \text{CF}^1(\mathcal{J}_C)} |\sigma| - 1,$$

recalling the convention that $d(\mathcal{C}) = \infty$ if $\mathcal{C}$ is not a convex code. In Section 6, we improve upon this bound. We also show that the presence or absence of a single codeword, namely the all-ones word 11\ldots1, can cause the dimension to jump from very small to very large:

**Proposition 1.12.** Let $\mathcal{C}$ be a code on $n$ neurons, and suppose that for some $k$ with $1 \leq k < n$, $\Delta(\mathcal{C})$ contains all $k$-dimensional faces. If $11\ldots1 \in \mathcal{C}$, then $d(\mathcal{C}) \leq 2$; otherwise, $d(\mathcal{C}) > k$.

Unfortunately, all our results on dimension rely only on information about the code that is present in the simplicial complex $\Delta(\mathcal{C})$. In our classification of convex codes for $n \leq 4$, however, it is clear that the presence or absence of specific codewords can affect $d(\mathcal{C})$, even if $\Delta(\mathcal{C})$ remains unchanged (see Table 5 in Section 4.1). The problem of how to use this additional information about a code in order to improve the bounds on $d(\mathcal{C})$ remains wide open.

**Open Questions.** The results summarized above suggest several open questions. First, what are the non-local obstructions to convexity, and how can we detect them? Can non-local obstructions arise for max $\cap$-complete codes, or does Conjecture 1.9 hold? Related to this, can we characterize max $\cap$-complete codes algebraically, as we did for $\cap$-complete codes? Finally, can we further improve the bounds on the minimal embedding dimension $d(\mathcal{C})$, using information about the code that goes beyond the simplicial complex data in $\Delta(\mathcal{C})$?
2 Local obstructions to convexity

We cannot, in general, determine whether or not a neural code \( \mathcal{C} \) is convex from data in its simplicial complex \( \Delta(\mathcal{C}) \) alone. This is because for any simplicial complex \( \Delta \), there exists a convex cover \( \mathcal{U} \) such that \( \Delta = \mathcal{N}(\mathcal{U}) \) [16]. Obstructions to convexity must therefore emerge from information in the code that goes beyond what is reflected in \( \Delta(\mathcal{C}) \). As was shown in [7], this additional information is precisely the receptive field relationships, which we turn to now.

2.1 Receptive field relationships

For a code \( \mathcal{C} \) on \( n \) neurons, let \( \mathcal{U} = \{U_1, \ldots, U_n\} \) be any collection of open sets such that \( \mathcal{C} = \mathcal{C}(\mathcal{U}) \). A receptive field relationship (RF relationship) of \( \mathcal{C} \) is a pair \((\sigma, \tau)\) corresponding to the set containment

\[
U_\sigma \subseteq \bigcup_{i \in \tau} U_i,
\]

where \( \sigma \neq \emptyset \), \( \sigma \cap \tau = \emptyset \), and \( U_\sigma \cap U_i \neq \emptyset \) for all \( i \in \tau \). If \( \tau = \emptyset \), then the relationship \((\sigma, \emptyset)\) simply states that \( U_\sigma = \emptyset \). Note that relationships of the form \((\sigma, \emptyset)\) reproduce the information in \( \Delta(\mathcal{C}) \), while those of the form \((\sigma, \tau)\) for \( \tau \neq \emptyset \) reflect additional structure in the code.

The set of all RF relationships \( \{(\sigma, \tau)\} \) for a given code \( \mathcal{C} \) is denoted \( \text{RF}(\mathcal{C}) \). A minimal RF relationship is one such that no single neuron can be removed from \( \sigma \) or \( \tau \) without destroying the containment. It is important to note that \( \text{RF}(\mathcal{C}) \) is well defined, independent of the choice of open sets \( \mathcal{U} \). Moreover, \( \text{RF}(\mathcal{C}) \) can be inferred directly from the code without appealing to a realization as \( \mathcal{C}(\mathcal{U}) \) [7], as will be explained later in Section 5.1.

Example 2.1 (Example 1.2 continued). The code \( \mathcal{C} = \mathcal{C}(\mathcal{U}) \) from Example 1.2 has RF relationships \( \text{RF}(\mathcal{C}) = \{(\{1, 4\}, \emptyset), (\{1, 2, 4\}, \emptyset), (\{1, 3, 4\}, \emptyset), (\{2\}, \{1, 3\}), (\{2\}, \{1, 3, 4\}), (\{2\}, \{3\})\} \). Of these, the pairs \( (\{1, 4\}, \emptyset), (\{2\}, \{1, 3\}) \), and \( (\{2\}, \{3\}) \), corresponding to \( U_1 \cup U_4 = \emptyset, U_2 \subset U_1 \cup U_3 \), and \( U_2 \cap U_4 \subset U_3 \), are the minimal RF relationships.

2.2 Local obstructions

Local obstructions are our main tool for proving that a code is not convex. We begin with a simple example that illustrates how a code can fail to have a convex realization.

Lemma 2.2. Let \( \mathcal{C} = \mathcal{C}(\mathcal{U}) \). If for some RF relationship \( U_\sigma \subseteq U_i \cup U_j \) we have that \( U_\sigma \cap U_i \cap U_j = \emptyset \), then \( \mathcal{C} \) is not a convex code.

Proof. If \((\sigma, \tau) = \{i, j\}\) is a RF relationship, then \( U_\sigma \subseteq U_i \cup U_j \), \( U_\sigma \cap U_i \neq \emptyset \), and \( U_\sigma \cap U_j \neq \emptyset \). Since by assumption \( U_\sigma \cap U_i \cap U_j = \emptyset \), the sets \( V_i = U_\sigma \cap U_i \) and \( V_j = U_\sigma \cap U_j \) are disjoint open sets that each intersect \( U_\sigma \), and such that \( U_\sigma \subseteq V_i \cup V_j \). It follows that \( U_\sigma \) is disconnected in any open cover \( \mathcal{U} \) such that \( \mathcal{C} = \mathcal{C}(\mathcal{U}) \), and so \( \mathcal{C} \) cannot have a convex realization.

The idea illustrated here is that \( U_\sigma \) is covered by a collection of sets whose topology does not match that of \( U_\sigma \). We can generalize this problem via the definition of a local obstruction to convexity. Local obstructions arise when

\[
U_\sigma \subseteq \bigcup_{i \in \tau} U_i,
\]

for nonempty \( \sigma, \tau \subset [n] \) with \( \sigma \cap \tau = \emptyset \), but the nerve of the corresponding cover of \( U_\sigma \) by the sets \( \{U_i\}_{i \in \tau} \) is not contractible. This violates the Nerve lemma (Section 1.2) if we assume the \( U_i \)'s are all convex, allowing us to conclude that \( \mathcal{C} \) cannot be a convex code. Recalling from Section 1.2 that

\[
\mathcal{N}(\{U_\sigma \cap U_i\}_{i \in \tau}) = \text{Lk}_\sigma(\Delta|_{\sigma \cup \tau}),
\]

where \( \Delta = \mathcal{N}(\mathcal{U}) \), we have the following definition:
Definition 2.3. Let \((\sigma, \tau) \in \text{RF}(C)\), and let \(\Delta = \Delta(C)\). We say that \((\sigma, \tau)\) is a local obstruction of \(C\) if \(\tau \neq \emptyset\) and \(\text{Lk}_\sigma(\Delta|_{\sigma \cup \tau})\) is not contractible.

Note that \(\tau \neq \emptyset\) implies \(\sigma \notin C\) and \(U_\sigma \neq \emptyset\), as the definition of a RF relationship requires that \(U_\sigma \cap U_i \neq \emptyset\) for all \(i \in \tau\). Any local obstruction \((\sigma, \tau)\) must therefore have \(\sigma \in \Delta(C) \setminus C\) and \(\text{Lk}_\sigma(\Delta|_{\sigma \cup \tau})\) nonempty. The motivation for the above definition comes from the following simple consequence of the Nerve lemma, which was previously observed in [13].

Lemma 2.4. If \(C\) has a local obstruction, then \(C\) is not a convex code.

Proof. We will assume that \(C\) is a convex code with a local obstruction \((\sigma, \tau)\), and obtain a contradiction. Let \(U = \{U_1, \ldots, U_n\}\) be a collection of convex open sets such that \(C = C(U)\), and let \(\Delta = \Delta(C)\). Observe that the intersections \(U_\sigma\) and \(U_\tau\) are also convex. It follows that \(\{U_\sigma \cap U_i\}_{i \in \tau}\) is a good cover of \(U_\sigma\). By the Nerve lemma, \(\mathcal{N}(\{U_\sigma \cap U_i\}_{i \in \tau})\) must be contractible because \(U_\sigma\) is contractible. However, \(\mathcal{N}(\{U_\sigma \cap U_i\}_{i \in \tau}) = \text{Lk}_\sigma(\Delta|_{\sigma \cup \tau})\), contradicting the fact that \((\sigma, \tau)\) is a local obstruction. \(\square\)

Recall from Section 1.4 that a code is locally good if it has no local obstructions. Lemma 1.5 now immediately follows as a corollary. The following simple condition on RF relationships guarantees that a code is locally good.

Lemma 2.5. Let \(C = C(U)\). If for each \((\sigma, \tau) \in \text{RF}(C)\) we have \(U_\sigma \cap U_\tau \neq \emptyset\), then \(C\) is locally good.

Proof. \(U_\sigma \cap U_\tau \neq \emptyset\) implies \(\text{Lk}_\sigma(\Delta|_{\sigma \cup \tau})\), for \(\Delta = \Delta(C)\), is the full simplex on the vertex set \(\tau\), which is contractible. If this is true for every RF relationship, then none can give rise to a local obstruction. \(\square\)

Note that if \(11 \cdots 1 \in C\), then \(U_\sigma \cap U_\tau \neq \emptyset\) for any pair \(\sigma, \tau \subset [n]\), so \(C\) is locally good. We thus obtain as a corollary a weaker version of C-8 in Theorem 1.8.

Corollary 2.6. If \(C\) contains the all-ones codeword, then \(C\) is locally good.

2.3 Link lemmas and the proof of Theorem 1.6

As we saw in the previous section, local obstructions are detected via non-contractible links of the form \(\text{Lk}_\sigma(\Delta|_{\sigma \cup \tau})\). Figure 3 displays all possible links that can arise for \(|\tau| \leq 4\). Non-contractible links are highlighted in red. In this section, we present some useful lemmas pertaining to the contractibility of links. We then use these lemmas to prove Theorem 1.6.

In what follows, the notation

\[
\text{cone}_v(\Delta) \overset{\text{def}}{=} \{\{v\} \cup \omega \mid \omega \in \Delta\}
\]

denotes the cone of \(v\) over \(\Delta\), where \(v\) is a new vertex not contained in \(\Delta\). Any simplicial complex that can be expressed as a cone over a sub-complex is automatically contractible. In Figure 4 the only contractible link that is not a cone is L13. This is the same link that appeared in Figure 3b.

Lemma 2.7. Let \(\Delta\) be a simplicial complex on \([n]\), \(\sigma \in \Delta\), and \(v \in [n]\) such that \(v \notin \sigma\). Then \(\text{Lk}_{\sigma \cup \{v\}}(\Delta) \subseteq \text{Lk}_\sigma(\Delta|_{[n]\setminus \{v\}})\), and

\[
\text{Lk}_\sigma(\Delta) = \text{Lk}_\sigma(\Delta|_{[n]\setminus \{v\}}) \cup \text{cone}_v(\text{Lk}_{\sigma \cup \{v\}}(\Delta)).
\]

Proof. The proof follows from the definition of the link. First, observe that

\[
\text{Lk}_{\sigma \cup \{v\}}(\Delta) = \{\omega \in [n] \mid v \notin \omega, \omega \cap \sigma = \emptyset, \text{ and } \omega \cup \sigma \cup \{v\} \in \Delta\}
= \{\omega \in [n] \setminus \{v\} \mid \omega \cap \sigma = \emptyset, \text{ and } \omega \cup \sigma \cup \{v\} \in \Delta\}
\subseteq \{\omega \in [n] \setminus \{v\} \mid \omega \cap \sigma = \emptyset, \text{ and } \omega \cup \sigma \in \Delta|_{[n]\setminus \{v\}}\}
= \text{Lk}_\sigma(\Delta|_{[n]\setminus \{v\}}),
\]

\[
\text{cone}_v(\Delta) = \{\{v\} \cup \omega \mid \omega \in \Delta\}.
\]
Figure 4: All simplicial complexes on up to 4 vertices, up to symmetry. These can each arise as links, $\text{Lk}_{\sigma}(\Delta |_{\sigma \cup \tau})$, for $|\tau| \leq 4$. Red labels correspond to non-contractible complexes. Note that L13 is the only simplicial complex on $n \leq 4$ vertices that is contractible but not a cone.
which establishes that \( \text{Lk}_{\sigma \cup \{v\}}(\Delta) \subseteq \text{Lk}_{\sigma}(\Delta|_{[n]\setminus \{v\}}) \). Next, observe that

\[
\text{cone}_v(\text{Lk}_{\sigma \cup \{v\}}(\Delta)) = \{\{v\} \cup \omega \mid \omega \in \text{Lk}_{\sigma \cup \{v\}}(\Delta)\} \\
= \{\{v\} \cup \omega \mid \omega \subseteq [n], v \notin \omega, \omega \cap \sigma = \emptyset, \text{ and } \omega \cup \{v\} \cup \sigma \in \Delta\} \\
= \{\tau \subseteq [n] \mid v \in \tau, \tau \cap \sigma = \emptyset, \text{ and } \tau \cup \sigma \in \Delta\} \\
= \{\omega \in \text{Lk}_\sigma(\Delta) \mid v \notin \omega\},
\]

and

\[
\text{Lk}_\sigma(\Delta|_{[n]\setminus \{v\}}) = \{\omega \in \text{Lk}_\sigma(\Delta) \mid v \notin \omega\}.
\]

From here the second statement is clear. \( \square \)

**Corollary 2.8.** Assume \( v \notin \sigma \). If \( \text{Lk}_{\sigma \cup \{v\}}(\Delta) \) is contractible, then \( \text{Lk}_\sigma(\Delta) \) and \( \text{Lk}_\sigma(\Delta|_{[n]\setminus \{v\}}) \) are homotopy-equivalent.

**Proof.** The above lemma shows that \( \text{Lk}_\sigma(\Delta) \) can be obtained from \( \text{Lk}_\sigma(\Delta|_{[n]\setminus \{v\}}) \) by coning off the subcomplex \( \text{Lk}_{\sigma \cup \{v\}}(\Delta) \). If this subcomplex is contractible, then the homotopy type is preserved. \( \square \)

Another useful corollary follows from the one above by simply setting \( \Delta = \Delta|_{[\sigma \cup \tau \cup \{v\}]} \) and \( [n] = \sigma \cup \tau \cup \{v\} \). We immediately see that if both \( \text{Lk}_\sigma(\Delta|_{[\sigma \cup \tau \cup \{v\}]}) \) and \( \text{Lk}_{\sigma \cup \{v\}}(\Delta|_{[\sigma \cup \tau \cup \{v\}]} \) are contractible, then \( \text{Lk}_\sigma(\Delta|_{[\sigma \cup \tau]} \) is contractible. Stated another way:

**Corollary 2.9.** Assume \( v \notin \sigma \), and \( \sigma \cap \tau = \emptyset \). If \( \text{Lk}_\sigma(\Delta|_{[\sigma \cup \tau]} \) is not contractible, then \( \text{Lk}_\sigma(\Delta|_{[\sigma \cup \tau \cup \{v\}]}) \) and/or \( \text{Lk}_{\sigma \cup \{v\}}(\Delta|_{[\sigma \cup \tau \cup \{v\}]} \) is not contractible.

This corollary can be extended to show that for every non-contractible link \( \text{Lk}_\sigma(\Delta|_{[\sigma \cup \tau]} \), there exists a non-contractible “big” link \( \text{Lk}_{\sigma'}(\Delta) \) for some \( \sigma' \supseteq \sigma \). This is because vertices outside of \( \sigma \cup \tau \) can be added one by one to either \( \sigma \) or its complement, preserving the non-contractibility of the new link at each step. In other words, we have the following lemma.

**Lemma 2.10.** Suppose \( \sigma \cap \tau = \emptyset \), and \( \text{Lk}_\sigma(\Delta|_{[\sigma \cup \tau]} \) is not contractible. Then there exists \( \sigma' \supseteq \sigma \) such that \( \sigma' \cap \tau = \emptyset \) and \( \text{Lk}_{\sigma'}(\Delta) \) is not contractible.

The next results show that only intersections of facets (maximal faces under inclusion) can possibly yield non-contractible links. For any \( \sigma \in \Delta \), we denote by \( \rho_\sigma \) the intersection of all facets of \( \Delta \) containing \( \sigma \). In particular, \( \sigma = \rho_\sigma \) if and only if \( \sigma \) is an intersection of facets of \( \Delta \). It is also useful to observe that a simplicial complex is a cone if and only if the common intersection of all its facets is non-empty. (Any element of that intersection can serve as a cone point, and a cone point is necessarily contained in all facets.)

**Lemma 2.11.** Let \( \sigma \in \Delta \). Then \( \sigma = \rho_\sigma \iff \text{Lk}_\sigma(\Delta) \) is not a cone.

**Proof.** Recall that \( \text{Lk}_\sigma(\Delta) \) is a cone if and only if all facets of \( \text{Lk}_\sigma(\Delta) \) have a non-empty common intersection \( \nu \). This can happen if and only if \( \sigma \cup \nu \subseteq \rho_\sigma \). Note that since \( \nu \in \text{Lk}_\sigma(\Delta) \), we must have \( \nu \cap \sigma = \emptyset \) and hence \( \text{Lk}_\sigma(\Delta) \) is a cone if and only if \( \sigma \neq \rho_\sigma \). \( \square \)

Furthermore, it is easy to see that every simplicial complex that is not a cone can in fact arise as the link of an intersection of facets. For any \( \Delta \) that is not a cone, simply consider \( \tilde{\Delta} = \text{cone}_v(\Delta) \); \( v \) is an intersection of facets of \( \tilde{\Delta} \), and \( \text{Lk}_v(\tilde{\Delta}) = \Delta \). The above lemma immediately implies the following corollary:

**Corollary 2.12.** If \( \sigma \neq \rho_\sigma \), then \( \text{Lk}_\sigma(\Delta) \) is a cone and hence contractible. In particular, if \( \text{Lk}_\sigma(\Delta) \) is not contractible, then \( \sigma \) must be an intersection of facets of \( \Delta \).
Using the above facts about links, we can now prove Theorem 1.6 which states that: (i) $C$ is locally good if and only if $M(\Delta(C)) \subsetneq C$, and (ii) if $\sigma \in M(\Delta)$, then $\sigma$ is an intersection of facets of $\Delta$. Recall that $M(\Delta)$ is the set of $\sigma \in \Delta$ such that $\text{Lk}_\sigma(\Delta)$ is not contractible.

**Proof of Theorem 1.6.** The second statement is a direct consequence of Corollary 2.12. We now focus on the first statement. $(\Rightarrow)$ We prove the contrapositive. Suppose there exists $\sigma \in \text{Lk}_\sigma(\Delta(C))$ such that $\sigma \notin C$. Then for $\Delta = \Delta(C)$, $\text{Lk}_\sigma(\Delta)$ is not contractible and thus $(\sigma, \bar{\sigma})$ is a local obstruction. It follows that $C$ is not locally good. $(\Leftarrow)$ We again prove the contrapositive. Suppose $C$ is not locally good, and let $(\alpha, \beta)$ be any local obstruction of $C$. This means that $\alpha \cap \beta \neq \emptyset$ and $\text{Lk}_\alpha(\Delta_{|\alpha \cup \beta})$ is not contractible. By Lemma 2.10 there exists $\sigma' \supseteq \sigma$ such that $\sigma' \cap \beta = \emptyset$ and $\text{Lk}_{\sigma'}(\Delta)$ is not contractible. This means $\sigma' \in M(\Delta(C))$. Moreover, $U_{\sigma'} \subseteq U_{\sigma} \subseteq U_{\text{i} \in \beta} U_{i}$ with $\sigma' \cap \beta = \emptyset$, which implies $\sigma' \notin C$. It follows that $M(\Delta(C)) \subsetneq C$. \hfill $\Box$

## 3 Computing mandatory codewords algebraically

Recall from C-2 of Theorem 1.8 (Section 1.4) that a code $C$ is non-convex if $M_H(\Delta(C)) \subsetneq C$, where $M_H(\Delta)$ denotes the elements of $\Delta$ with non-contractible links that can be detected via homology:

$$M_H(\Delta) \overset{\text{def}}{=} \{ \sigma \in \Delta \mid \dim \bar{H}_i(\text{Lk}_\sigma(\Delta)) > 0 \text{ for some } i \}.$$

If there exists an element $\sigma \in M_H(\Delta)$ such that $\sigma \notin C$, then we can conclude that $C$ is not convex. Otherwise, all we can say is that $C$ has no local obstructions that are detectable via the dimensions of homology groups.

In this section, we explain how to compute $M_H(\Delta)$ algebraically using free resolutions of monomial ideals. This allows us to quickly identify all elements in $M_H(\Delta)$ using existing computational algebra software, such as Macaulay2 [8]. The key to computing $M_H(\Delta)$ algebraically is an application of Hochster’s formula, which we review in Section 3.1. In Section 3.2 we explain how this can be used to find $M_H(\Delta)$ and illustrate the method with an example.

### 3.1 Alexander duality and Hochster’s formula

For any simplicial complex $\Delta$ on vertex set $[n]$, the Alexander dual is the related simplicial complex:

$$\Delta^* \overset{\text{def}}{=} \{ \bar{\tau} \mid \tau \notin \Delta \},$$

where $\bar{\tau} = [n] \setminus \tau$ denotes the complement of $\tau$ in $[n]$. Note that $(\Delta^*)^* = \Delta$. Alexander duality relates the reduced homology of a simplicial complex to the cohomology of its Alexander dual:

$$\bar{H}_i(\Delta^*) \cong \bar{H}^{n-i-3}(\Delta).$$

Hochster’s formula relates the nonzero Betti numbers from a minimal free resolution of a Stanley-Reisner ideal to the reduced cohomology of restricted simplicial complexes [8]. Specifically,

$$\beta_{i-1,\sigma}(I_\Delta) = \beta_{i,\sigma}(S/I_\Delta) = \dim \bar{H}_{|\sigma|-i-1}(\Delta_{|\sigma}),$$

where $S = k[x_1, \ldots, x_n]$ and $\beta_{i,\sigma}$ refer always to Betti numbers for a minimal free resolution.

The following link lemma can be used to derive the dual version of Hochster’s formula, which is also well known [8, Corollary 1.40]. The dual formulation is more useful to us, as it allows us to compute the dimensions of all non-trivial homology groups for all links, $\text{Lk}_\sigma(\Delta)$, from a single free resolution.

---

This is all that can be concluded because $M_H(\Delta)$ is only a subset of the “mandatory” codewords $M(\Delta)$ that are necessary for a code to be locally good (Theorem 1.6). The reason $M_H(\Delta) \neq M(\Delta)$ in general is that we do not measure torsion, and moreover a homologically trivial simplicial complex also needs trivial fundamental group in order to be contractible. For example, consider the 2-skeleton of a triangulation of the Poincare homology sphere: this simplicial complex has all-vanishing reduced homology groups, but is non-contractible due to its non-trivial fundamental group [17].
Lemma 3.1. \( \text{Lk}_\sigma(\Delta) = (\Delta^*|_{\sigma})^* \).

Proof. First, observe that \( \Delta^*|_{\sigma} = \{ \tau \mid \tau \subset \sigma \text{ and } \tau \in \Delta^* \} \). The dual is thus \( (\Delta^*|_{\sigma})^* = \{ \sigma \setminus \tau \mid \tau \subset \sigma \text{ and } \tau \notin \Delta^* \} = \{ \omega \mid \omega \subset \sigma \text{ and } \sigma \setminus \omega \notin \Delta^* \} = \{ \omega \mid \omega \cap \sigma = \emptyset \text{ and } \sigma \cup \omega \in \Delta \} = \text{Lk}_\sigma(\Delta) \). \( \square \)

Lemma 3.2 (Hochster’s formula, dual version). \( \dim \tilde{H}_i(\text{Lk}_\sigma(\Delta)) = \beta_{i+2,\sigma}(S/I_{\Delta^*}) \).

Proof. Using (in order) Lemma 3.1, Alexander duality, and the original version of Hochster’s formula (above), we obtain: \( \dim \tilde{H}_i(\text{Lk}_\sigma(\Delta)) = \dim \tilde{H}_i((\Delta^*|_{\sigma})^*) = \dim \tilde{H}^{\sigma|_{\sigma}}-i-3((\Delta^*|_{\sigma})^*) = \beta_{i+2,\sigma}(S/I_{\Delta^*}) \). \( \square \)

It is important to note that if \( \sigma \) is a facet of \( \Delta \), then \( \text{Lk}_\sigma(\Delta) = \emptyset \), which is non-contractible due to nontrivial homology in degree \(-1\). Hochster’s formula thus detects facets of \( \Delta \) via the nonzero Betti numbers \( \beta_{1,\sigma} \), as these correspond to \( \sigma \) such that \( \dim \tilde{H}_{-1}(\text{Lk}_\sigma(\Delta)) > 0 \). Note also that \( \text{Lk}_\emptyset(\Delta) = \Delta \), so if \( \Delta \) itself has nontrivial reduced homology in degree \( i \), this will be detected as a nonzero Betti number \( \beta_{i+2,[n]} \), where \( \bar{\sigma} = [n] \) is the complement of \( \sigma = \emptyset \).

3.2 Using free resolutions to compute \( \mathcal{M}_H(\Delta) \)

In the last section we saw that for any simplicial complex \( \Delta \), with Alexander dual \( \Delta^* \), Hochster’s formula (Lemma 3.2) relates the homology of the links \( \text{Lk}_\sigma(\Delta) \) to the Betti numbers \( \beta_{i,\sigma}(S/I_{\Delta^*}) \) of a minimal free resolution of the Stanley-Reisner ring \( S/I_{\Delta^*} \). The set of all \( \sigma \in \Delta \) such that \( \text{Lk}_\sigma(\Delta) \) has nontrivial homology is thus given by:

\[
\mathcal{M}_H(\Delta) = \{ \sigma \in \Delta \mid \beta_{i,\sigma}(S/I_{\Delta^*}) > 0 \text{ for some } i > 0 \}.
\]

As illustrated in the following example, the nonzero \( \beta_{i,\sigma}(S/I_{\Delta^*}) \) can be read off of a minimal free resolution for the \( S \)-module \( S/I_{\Delta^*} \). This process can also be automated using standard computational algebra software such as Macaulay2 [18].

Example 3.3. Let \( \Delta \) be the simplicial complex L25 in Figure 4. The Stanley-Reisner ideal is given by \( I_\Delta = \langle x_1x_2x_4, x_2x_3x_4 \rangle \), and its Alexander dual is \( I_{\Delta^*} = \langle x_1, x_2, x_4 \rangle \cap \langle x_2, x_3, x_4 \rangle = \langle x_1x_3, x_2, x_4 \rangle \). A minimal free resolution of \( S/I_{\Delta^*} \) is:

\[
0 \leftarrow S/I_{\Delta^*} \xleftarrow{\left[ \begin{array}{ccc} x_1x_3 & x_2 & x_4 \end{array} \right]} S(-2) \oplus S(-1)^2 \xleftarrow{\left[ \begin{array}{ccc} x_2 & x_4 & 0 \\ -x_1x_3 & 0 & x_4 \\ 0 & -x_1x_3 & -x_2 \end{array} \right]} S(-3)^2 \oplus S(-2) \leftarrow S(-4) \leftarrow 0
\]

The Betti number \( \beta_{i,\sigma}(S/I_{\Delta^*}) \) is the dimension of the module in multidegree \( \sigma \) at step \( i \) of the resolution, where \( S/I_{\Delta^*} \) is step 0 and the steps increase as we move from left to right. At step 0, the total degree is always 0. For the above resolution, the multidegrees at \( S(-2) \oplus S(-1)^2 \) (step 1) are 1010, 0100, and 0001; at \( S(-3)^2 \oplus S(-2) \) (step 2), we have 1110, 1011, and 0101; and at \( S(-4) \) (step 4) the multidegree is 1111. This immediately gives us the nonzero Betti numbers:

\[
\begin{align*}
\beta_{0,0000}(S/I_{\Delta^*}) &= 1, & \beta_{1,1010}(S/I_{\Delta^*}) &= 1, & \beta_{1,0100}(S/I_{\Delta^*}) &= 1, & \beta_{1,0001}(S/I_{\Delta^*}) &= 1, \\
\beta_{2,1110}(S/I_{\Delta^*}) &= 1, & \beta_{2,1011}(S/I_{\Delta^*}) &= 1, & \beta_{2,0101}(S/I_{\Delta^*}) &= 1, & \beta_{3,1111}(S/I_{\Delta^*}) &= 1.
\end{align*}
\]

Recalling that the multidegrees correspond to complements \( \bar{\sigma} \) of faces in \( \Delta \), we can now immediately read off the elements of \( \mathcal{M}_H(\Delta) \) from the above \( \beta_{i,\sigma} \) for \( i > 0 \) as:

\[
\mathcal{M}_H(\Delta) = \{0101, 1011, 1110, 0001, 0100, 1010, 0000\} = \{24, 134, 123, 4, 2, 13, 0\},
\]

where we have used binary strings and subsets of \([n]\) interchangeably to represent faces.

See [8] Chapter 1] for more details about free resolutions of monomial ideals.
Note that the first three elements of $\mathcal{M}_H(\Delta)$ in Example 3.3, obtained from the Betti numbers $\beta_{1,*}$ in step 1 of the resolution, are precisely the facets of $\Delta$. The next three elements, 0001, 0100, and 1010, are non-maximal codewords that must be included for a code with simplicial complex $\Delta$ to be convex. These all correspond to pairwise intersections of facets, and are obtained from the Betti numbers $\beta_{2,*}$ at step 2 of the resolution; this is consistent with the fact that the corresponding links are all disconnected, resulting in non-trivial $H_0(\text{Lk}_\sigma(\Delta))$. The last element, 0000, reflects the fact that $\text{Lk}_0(\Delta) = \Delta$, and $\dim H_1(\Delta) = 1$ for $\Delta = L25$. By convention, however, we always include the all-zeros codeword in our codes (Section 1.2).

Using Macaulay2 [18], the Betti numbers for the simplicial complex $\Delta$ in Example 3.3 can be computed through the following sequence of commands (suppressing outputs except for the Betti tally at the end):

```plaintext
i1 : kk = ZZ/2;

i2 : S = kk[x1,x2,x3,x4, Degrees => {{1,0,0,0},{0,1,0,0},{0,0,1,0},{0,0,0,1}}];

i3 : I = monomialIdeal(x1*x2*x4,x2*x3*x4);

i4 : Istar = dual(I);

i5 : M = S^1/Istar;

i6 : Mres = res M; \text{[comment: this step computes the minimal free resolution]}

i7 : peek betti Mres

o7 = BettiTally{(0, {0, 0, 0, 0}, 0) => 1}
   (1, {0, 0, 0, 1}, 1) => 1
   (1, {0, 1, 0, 0}, 1) => 1
   (1, {1, 0, 1, 0}, 2) => 1
   (2, {0, 1, 0, 1}, 2) => 1
   (2, {1, 0, 1, 1}, 3) => 1
   (2, {1, 1, 1, 0}, 3) => 1
   (3, {1, 1, 1, 1}, 4) => 1
```

Each line of the BettiTally displays $(i, \{\sigma\}, |\sigma|) \Rightarrow \beta_{i,\sigma}$. This yields (in order):

$\beta_{0,0000} = 1$, $\beta_{1,0001} = 1$, $\beta_{1,0100} = 1$, $\beta_{1,0101} = 1$, $\beta_{2,0101} = 1$, $\beta_{2,1110} = 1$, $\beta_{3,1111} = 1$,

which is the same set of nonzero Betti numbers we previously obtained in Example 3.3. Recalling, again, that the multidegrees correspond to complements $\widetilde{\sigma}$, this output immediately gives us $\mathcal{M}_H(\Delta)$, exactly as in Example 3.3.

4 Combinatorial signatures of convex codes

In Sections 2 and 3, we showed how local obstructions can be used to prove that neural codes are not convex. In this section, we present a collection of combinatorial signatures that allow one to draw the opposite conclusion: that a code is convex. In Section 4.1, we classify all convex codes on $n \leq 4$ neurons, and show that these correspond precisely to max $\cap$-complete codes; in Section 4.2, we prove that codes with non-overlapping maximal codewords are all convex and low-dimensional; and in Section 4.3, we show that any code that is both linear and max $\cap$-complete is necessarily convex. Along the way, we give our remaining proofs for Theorem 1.8 corresponding to rows C-5, C-8, and C-9.

4.1 Classification of convex codes on $n \leq 4$ neurons

For $n = 1$ or $n = 2$, all codes are convex. The first non-convex codes appear for $n = 3$. Using our convention that all codes include the all-zeros codeword, there are a total of 40 permutation-inferior-equivalent codes on 3 neurons [7]. Of these, only 6 are non-convex. These can all be detected and proven to be non-convex via local obstructions.
**Proposition 4.1.** There are 6 non-convex codes on \( n \leq 3 \) neurons, up to permutation equivalence. They are the codes shown in Table 4.

<table>
<thead>
<tr>
<th>label</th>
<th>code ( \mathcal{C} )</th>
<th>( \Delta(\mathcal{C}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>B3</td>
<td>000, 010, 001, 110, 101</td>
<td>L6</td>
</tr>
<tr>
<td>B5</td>
<td>000, 010, 110, 101</td>
<td>L6</td>
</tr>
<tr>
<td>B6</td>
<td>000, 110, 101</td>
<td>L6</td>
</tr>
<tr>
<td>E2</td>
<td>000, 100, 010, 110, 101, 011</td>
<td>L7</td>
</tr>
<tr>
<td>E3</td>
<td>000, 100, 110, 101, 011</td>
<td>L7</td>
</tr>
<tr>
<td>E4</td>
<td>000, 110, 101, 011</td>
<td>L7</td>
</tr>
</tbody>
</table>

Table 4: All non-convex codes on \( n = 3 \) neurons, up to permutation equivalence. Code labels are the same as in [7].

**Proof.** Codes B3, B5, and B6 all have simplicial complex L6, with facets \( \{12\} \) and \( \{13\} \), but are missing the codeword 100, corresponding to \( \sigma = \{1\} \in \Delta(\mathcal{C}) \). For each of these codes, the pair \((\sigma, \tau) = (\{1\}, \{2, 3\})\) is a local obstruction because \( \text{Lk}_1(\Delta) \) is the simplicial complex L2, which is not contractible. Codes E2, E3, and E4 all have simplicial complex L7, with facets \( \{12\} \), \( \{13\} \), and \( \{23\} \), but are missing the codeword 001, corresponding to \( \sigma = \{3\} \in \Delta(\mathcal{C}) \). For each of these codes, the pair \((\sigma, \tau) = (\{3\}, \{1, 2\})\) is a local obstruction because \( \text{Lk}_3(\Delta) \) is the simplicial complex L2, which is not contractible. All remaining codes for \( n = 3 \) neurons were shown to be convex in [7], via explicit convex realizations in two dimensions.

**Lemma 4.2.** If \( n \leq 4 \), then \( \text{Lk}_\rho(\Delta) \) is not contractible for any \( \rho \) that is the intersection of two or more facets of \( \Delta \).

**Proof.** See Figure 5, with intersections of two or more facets highlighted in red and orange. One can easily check that all corresponding links are non-contractible.

**Corollary 4.3.** If \( n \leq 4 \), then \( \mathcal{C} \) is locally good if and only if \( \mathcal{C} \) is max \( \cap \)-complete.

**Proof.** The backward direction is an immediate consequence of Theorem 1.6. For the forward direction, consider the contrapositive and observe that if \( \mathcal{C} \) is not max \( \cap \)-complete, then for any intersection of facets \( \rho \notin \mathcal{C} \), \( \text{Lk}_\rho(\Delta) \) is not contractible by Lemma 4.2, so \( \mathcal{C} \) is not locally good.

In fact, for \( n \leq 4 \) we have exhaustively checked that all codes with no local obstructions are in fact convex. Table 5 classifies convex and non-convex codes according to the corresponding simplicial complexes that can arise as \( \Delta(\mathcal{C}) \). Figure 6 shows explicit convex realizations for most of the simplicial complexes L1-L28. The omitted complexes L1-L5, L9-L10, and L12 all have obvious convex realizations, while L15 and L16 are obvious given the realizations for L7 and L8, respectively. We thus have the following theorem, which is equivalent to C-5 of Theorem 1.8.

**Theorem 4.4.** If \( n \leq 4 \), then \( \mathcal{C} \) is convex if and only if \( \mathcal{C} \) is max \( \cap \)-complete.

As we saw in Example 1.7, for \( n > 5 \) there exist convex codes that are not max \( \cap \)-complete, so this theorem cannot be extended to \( n > 4 \).
Figure 5: Classification of convex codes for $n \leq 4$. L1-L28 are the 28 possible simplicial complexes $\Delta(C)$ that can arise. For each simplicial complex, faces highlighted in color correspond to pairwise (red) and triple (orange) intersections of facets (maximal faces). By Theorem 4.4, any code $C$ with simplicial complex $\Delta(C)$ is convex if and only if $C$ includes all codewords corresponding to the colored faces. All other non-maximal faces are optional codewords, whose inclusion or exclusion does not affect convexity. Simplicial complexes with no optional codewords are labeled $+$, while those with no (non-maximal) mandatory codewords are labeled $\circ$. When both $+$ and $\circ$ labels apply, as in L1, L2, and L4, we simply use $\circ$. The possible minimal embedding dimensions $d = d(C)$ that can arise for convex codes are shown in the lower right corners.
Table 5: Convexity and dimension for codes on $n \leq 4$ neurons. For each $\Delta$, the second column is the number of non-convex codes $\mathcal{C}$ such that $\Delta(\mathcal{C}) = \Delta$, up to permutation equivalence and including the all-zeros codeword, while the sixth column $d(\mathcal{C})$ displays the possible minimal embedding dimensions for convex codes only. Mandatory codewords are non-maximal elements of $\Delta$ that must be included in order for $\mathcal{C}$ to be convex. Optional codewords do not affect whether or not $\mathcal{C}$ is convex, though they may alter the minimal embedding dimension $d(\mathcal{C})$. When ‘all’ non-maximal codewords are mandatory or optional, their total number is given in parentheses. The picture column indicates the groupings used for the convex realizations in Table 5. In the notes column, * indicates that the set of optional codewords in $\mathcal{C}$ can not form a 2-chain. A collection of codewords forms a chain if we can completely order the respective sets by containment – so $\{1111, 1100, 1000\}$ is a chain, but $\{1110, 1000, 1101\}$ is not. A collection of codewords can form a 2-chain if it can be partitioned into two sets (possibly empty) which are both chains. + and o are the same as in Figure 5.

<table>
<thead>
<tr>
<th>$\Delta$</th>
<th># non-convex</th>
<th>maximal codewords (facets)</th>
<th>mandatory codewords</th>
<th>optional codewords of $\Delta$</th>
<th>$d(\mathcal{C})$</th>
<th>picture</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1^*$</td>
<td>0</td>
<td>1</td>
<td>none</td>
<td>none</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_2^*$</td>
<td>0</td>
<td>10, 01</td>
<td>none</td>
<td>none</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_3^*$</td>
<td>0</td>
<td>11</td>
<td>none</td>
<td>all (2)</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_4^*$</td>
<td>0</td>
<td>100, 010, 001</td>
<td>none</td>
<td>none</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_5^*$</td>
<td>0</td>
<td>110, 001</td>
<td>none</td>
<td>all (2)</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_6^*$</td>
<td>3</td>
<td>110, 101</td>
<td>100, 010, 001</td>
<td>1 L6-series</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_7^+$</td>
<td>3</td>
<td>110, 101, 011</td>
<td>all (3)</td>
<td>none</td>
<td>2 L7-series</td>
<td>convex $\Leftrightarrow \mathcal{C} = \Delta(\mathcal{C})$</td>
<td></td>
</tr>
<tr>
<td>$L_8^*$</td>
<td>0</td>
<td>111</td>
<td>none</td>
<td>all (6)</td>
<td>1, 2 L8-series</td>
<td>$d = 2 \Leftrightarrow$ *</td>
<td></td>
</tr>
<tr>
<td>$L_9^*$</td>
<td>0</td>
<td>1000, 0100, 0001</td>
<td>none</td>
<td>none</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{10}^*$</td>
<td>0</td>
<td>1100, 0010, 0001</td>
<td>none</td>
<td>all (2)</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{11}^*$</td>
<td>3</td>
<td>1100, 1010, 0001</td>
<td>1000, 0100, 0010</td>
<td>1 L6-series</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{12}^*$</td>
<td>0</td>
<td>1100, 0011</td>
<td>none</td>
<td>all (4)</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{13}^*$</td>
<td>7</td>
<td>1100, 1010, 0011</td>
<td>1000, 0010, 1000, 0001</td>
<td>1 L6-series</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{14}^*$</td>
<td>4</td>
<td>1100, 1010, 1001</td>
<td>1000, 0100, 0010, 0001</td>
<td>1, 2 L6-series</td>
<td>$d = 2 \Leftrightarrow$ *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{15}^+$</td>
<td>3</td>
<td>1100, 1010, 0110, 0001</td>
<td>all (3)</td>
<td>none</td>
<td>2 like L7</td>
<td>convex $\Leftrightarrow \mathcal{C} = \Delta(\mathcal{C})$</td>
<td></td>
</tr>
<tr>
<td>$L_{16}^*$</td>
<td>0</td>
<td>1110, 0001</td>
<td>none</td>
<td>all (6)</td>
<td>1, 2 like L8</td>
<td>$d = 2 \Leftrightarrow$ *</td>
<td></td>
</tr>
<tr>
<td>$L_{17}^*$</td>
<td>10</td>
<td>1100, 1010, 0110</td>
<td>1000, 0100, 0010, 0001</td>
<td>0001</td>
<td>2 L7-series</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{18}^*$</td>
<td>40</td>
<td>1110, 0101</td>
<td>0100</td>
<td>1000, 0010, 0001, 1100, 1010, 0110</td>
<td>1, 2 L8-series</td>
<td>$d = 2 \Leftrightarrow \mathcal{C}$ contains {1000, 0010}, or {1100, 0110, 1010}</td>
<td></td>
</tr>
<tr>
<td>$L_{19}^*$</td>
<td>5</td>
<td>1100, 1010, 0110, 0011</td>
<td>all (4)</td>
<td>none</td>
<td>2 L6-series</td>
<td>convex $\Leftrightarrow \mathcal{C} = \Delta(\mathcal{C})$</td>
<td></td>
</tr>
<tr>
<td>$L_{20}^*$</td>
<td>8</td>
<td>1100, 1010, 0101, 0011</td>
<td>all (4)</td>
<td>none</td>
<td>2 L7-series</td>
<td>convex $\Leftrightarrow \mathcal{C} = \Delta(\mathcal{C})$</td>
<td></td>
</tr>
<tr>
<td>$L_{21}^*$</td>
<td>68</td>
<td>1110, 0101, 0011</td>
<td>0100, 0010, 0001, 1000, 1100, 1010, 0110</td>
<td>2 L8-series</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{22}^*$</td>
<td>62</td>
<td>1110, 0111</td>
<td>0110</td>
<td>1000, 0100, 0010, 0001, 1100, 1010, 0110</td>
<td>1, 2 L22-series</td>
<td>{0101, 0011} \subset \mathcal{C}, or {1100, 1010} \subset \mathcal{C}</td>
<td></td>
</tr>
<tr>
<td>$L_{23}^+$</td>
<td>4</td>
<td>1100, 1010, 0010, 0110, 0011</td>
<td>all (4)</td>
<td>none</td>
<td>2 L7-series</td>
<td>convex $\Leftrightarrow \mathcal{C} = \Delta(\mathcal{C})$</td>
<td></td>
</tr>
<tr>
<td>$L_{24}^*$</td>
<td>36</td>
<td>1110, 0100, 0101, 0111</td>
<td>1000, 0100, 0010, 0001, 1100, 1010, 0110</td>
<td>2, 3 L8-series</td>
<td>$d = 3$ \Leftrightarrow no optional codewords</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{25}^*$</td>
<td>168</td>
<td>1110, 1011, 0101</td>
<td>0100, 0001, 1010, 1010, 1000, 1100, 0110, 1011, 0111</td>
<td>2 L22-series</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{26}^*$</td>
<td>407</td>
<td>1110, 1011, 0111</td>
<td>0010, 1010, 0010, 0110, 1100, 1000, 0010, 1101, 0101, 0111</td>
<td>2 L22-series</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{27}^+$</td>
<td>85</td>
<td>1110, 1101, 0111</td>
<td>all (10)</td>
<td>none</td>
<td>3 L8-series</td>
<td>$d = 3$ \Leftrightarrow convex $\Leftrightarrow \mathcal{C} = \Delta(\mathcal{C})$</td>
<td></td>
</tr>
<tr>
<td>$L_{28}^*$</td>
<td>0</td>
<td>1111</td>
<td>none</td>
<td>all (14)</td>
<td>1, 2 L22-series</td>
<td>$d = 2 \Leftrightarrow$ *</td>
<td></td>
</tr>
</tbody>
</table>
Figure 6: Convex realizations for codes on $n \leq 4$ neurons.
4.2 Codes with non-overlapping maximal codewords

We begin with an example of a code that contains the all-ones codeword, and illustrate how it can be embedded in $\mathbb{R}^2$.

**Example 4.5.** For the code $C = \{1111, 1011, 1101, 1100, 0011, 0010, 0001, 0000\}$, Figure 7 depicts the construction of a convex realization in $\mathbb{R}^2$, which is described more generally in the proof of Proposition 4.6. Here $C$ has a unique maximal codeword, the all-ones word.

![Figure 7: A convex realization in $\mathbb{R}^2$ of the code $C = \{1111, 1011, 1101, 1100, 0011, 0010, 0001, 0000\}$.](image)

If each nonzero codeword is contained in a unique facet of $\Delta(C)$, the facets provide a partition of the code into non-overlapping parts. The above construction can be repeated in parallel to obtain the same dimension bound, giving us the following proposition which is equivalent to C-8 of Theorem 1.8.

**Proposition 4.6.** Let $C \subseteq \{0,1\}^n$ be a neural code. If the facets of $\Delta(C)$ are all disjoint (i.e., all maximal codewords of $C$ are non-overlapping), then $C$ is convex and $d(C) \leq 2$.

**Proof.** Let $\rho_1, \ldots, \rho_k \subseteq [n]$ be the disjoint facets of $\Delta(C)$, and define $C|_{\rho_i}$ to be the restricted code consisting of the codewords of $C$ whose supports are contained in $\rho_i$, excluding the all-zeros word, $00 \cdots 0$. Note that $C \setminus 00 \cdots 0$ is precisely the disjoint union of these restricted codes $C|_{\rho_1}, \ldots, C|_{\rho_k}$, because every nonzero codeword of $C$ is contained in a unique facet. We can thus construct a realization of $C$ in $\mathbb{R}^2$ by realizing each $C|_{\rho_i}$ with sets $\{U_j\}_{j \in \rho_i}$ separately, and then taking the union of these disjoint realizations. The all-zeros codeword, $00 \cdots 0 \in C$, is assigned to the region of $\mathbb{R}^2$ not covered by these realizations.

To realize $C|_{\rho_i}$, let $m_i = |(C|_{\rho_i})| - 1$, the number of non-maximal codewords in $C|_{\rho_i}$. Inscribe a regular $m_i$-gon $P_i$ in a circle, so that there are $m_i$ sectors surrounding $P_i$ (as in Figure 7). If $m_i < 3$, inscribe a triangle into the circle and leave any unnecessary sectors unlabeled. To each sector assign a distinct non-maximal codeword $c_j \in C|_{\rho_i}$, and to the polygon $P_i$ assign the maximal codeword $\rho_i$. For each $j \in \rho_i$, set $U_j$ to be the union of the polygon $P_i$ and all sectors whose assigned codeword contains $j$ in its support. Each $U_j$ is open and convex, and $C|_{\rho_i} = C(\{U_j\}_{j \in \rho_i})$. Finally, taking $U = \bigcup_{i=1}^{k} \{U_j\}_{j \in \rho_i} \subseteq \mathbb{R}^2$ results in $C = C(U)$, so we can conclude that $d(C) \leq 2$.

As an immediate consequence, we see that if $\Delta(C)$ contains the all-ones word, and thus has a unique facet, then $C$ has a convex realization in $\mathbb{R}^2$.

**Corollary 4.7.** If $11 \cdots 1 \in C$, then $C$ is convex and $d(C) \leq 2$. 

4.3 Linear codes and convexity

We now turn our attention briefly to linear codes, which are the primary codes of interest in classical coding theory. A code is called linear if it forms a subspace over its ground field; restricting to codes over $\mathbb{F}_2$, a code is linear if and only if every sum of codewords is also a codeword.
Linear codes are particularly useful in engineering applications because they have a compact representation that allows for efficient storage and simplified computations [19]. Specifically, a linear code can be described as the nullspace of some (non-unique) matrix, known as a parity check matrix $H$. The code $C(H)$ then consists of all binary vectors $c$ such that $Hc = 0$. It is natural to associate to each row $H_i$ of the parity check matrix a linear polynomial $H_i x \in \mathbb{F}_2[x_1, \ldots, x_n]$, which vanishes when evaluated at codewords of $C$. This allows us to define a parity check ideal $J_H$ for any linear code:

$$J_H \overset{\text{def}}{=} \langle H, x \rangle.$$

Although $J_H$ is generated by linear polynomials, while the neural ideal $J_C$ is generated by pseudomonomials, it is not difficult to show that for linear codes, $J_H = J_C$ (see Appendix).

Given the importance of linear codes in coding theory, it is natural to ask how the properties of linearity and convexity interact. Using Theorem 4.4 (C-5 of Theorem 1.8), we can determine whether or not a code of length $n$, up to permutation-equivalence and having no trivial coordinates, is convex by simply checking if it is max $\cap$-complete. Table 6 shows all linear codes for $n \leq 4$ (up to permutation-equivalence and having no trivial coordinates), organized by their corresponding simplicial complexes. There are both convex and non-convex linear codes, indicating that these properties do not have a straightforward relationship. Note that every convex code in Table 6 contains the all-ones word, and is thus known to satisfy $d(C) \leq 2$ by Corollary 4.7 (C-8 of Theorem 1.8). This observation generalizes.

<table>
<thead>
<tr>
<th>possible $\Delta(C)$</th>
<th># linear codes</th>
<th>linear codes</th>
<th>convex?</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3</td>
<td>2</td>
<td>$C_1 = 00, 11$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_2 = 00, 01, 10, 11$</td>
<td>yes</td>
</tr>
<tr>
<td>L7</td>
<td>1</td>
<td>$C_1 = 000, 110, 011, 101$</td>
<td>no</td>
</tr>
<tr>
<td>L8</td>
<td>3</td>
<td>$C_1 = 000, 111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_2 = 000, 100, 011, 111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_3 = \mathbb{F}_2^4$</td>
<td>yes</td>
</tr>
<tr>
<td>L25</td>
<td>1</td>
<td>$C_1 = 0000, 1110, 1011, 0101$</td>
<td>no</td>
</tr>
<tr>
<td>L26</td>
<td>1</td>
<td>$C_1 = 0000, 1110, 1011, 0111, 0101, 1100, 1001, 0010$</td>
<td>no</td>
</tr>
<tr>
<td>L28</td>
<td>6</td>
<td>$C_1 = 0000, 1111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_2 = 0000, 1000, 0111, 1111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_3 = 0000, 1000, 0100, 1100, 0011, 1011, 0111, 1111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_4 = 0000, 1100, 0011, 1111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_5 = 0000, 1100, 1010, 1001, 0110, 0101, 0011, 1111$</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_6 = \mathbb{F}_2^4$</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 6: All linear codes for $n \leq 4$, up to permutation equivalence and with no trivial coordinates (i.e. no coordinates are 0 in all codewords). For each possible simplicial complex, all corresponding linear codes are listed in the third column. A ‘yes’ or ‘no’ in the final column indicates whether or not a code is convex.

**Lemma 4.8.** If $C$ is a linear max $\cap$-complete code, then $C$ has a unique maximal codeword.

**Proof.** Let $C$ be a linear max $\cap$-complete code, and suppose that $C$ contains at least two distinct maximal codewords: $p_1$ and $p_2$. Since $C$ is max $\cap$-complete, $p_1 \cap p_2 \in C$. Now consider $p_1 \cup p_2 = p_1 + p_2 + p_1 \cap p_2$. Because $C$ is linear, $p_1 \cup p_2 \in C$, but this contradicts the assumption that $p_1$ and $p_2$ are maximal and distinct. We conclude that $C$ has a unique maximal codeword. \qed

As an immediate corollary we obtain the following proposition, which is equivalent to C-9 of Theorem 1.8.
Proposition 4.9. If $C$ is a linear max $\cap$-complete code, then $C$ is convex and $d(C) \leq 2$.

Proof. Lemma 4.8 tells us that $\Delta(C)$ has a unique facet. Applying Proposition 4.6 (C-8 of Theorem 1.8), we conclude that $C$ is convex and $d(C) \leq 2$.

5 Algebraic signatures of convex and non-convex codes

In this section we begin by highlighting the connection between the neural ideal $J_C$ and the receptive field relationships $RF(C)$. Next, in Section 5.2 we present an algebraic characterization of $\cap$-complete codes in terms of the canonical form $CF(J_C)$. Finally, we prove Theorem 1.10 in Section 5.3.

5.1 The canonical form $CF(J_C)$ vs. receptive field relationships $RF(C)$

Here we review some key results from [7], relating the neural ideal $J_C$ (defined in Section 1.3) to receptive field relationships $RF(C)$ (described in Section 2.1). The following lemma is essentially equivalent to [7, Lemma 4.2].

Lemma 5.1. Let $C$ be a neural code on $n$ neurons, with neural ideal $J_C$. For $\sigma, \tau \subset [n]$ with $\sigma \cap \tau = \emptyset$,

$$x_\sigma \prod_{i \in \tau} (1 - x_i) \in J_C \iff U_\sigma \subseteq \bigcup_{i \in \tau} U_i.$$ 

In Section 1.3 we saw that elements of the form $x_\sigma \prod_{i \in \tau} (1 - x_i)$, with $\sigma \cap \tau = \emptyset$, are called pseudo-monomials and naturally break up into three types. Lemma 5.1 indicates that each type corresponds to a different kind of RF relationship:

- Type 1: $x_\sigma \in J_C \iff U_\sigma = \emptyset$.
- Type 2: $x_\sigma \prod_{i \in \tau} (1 - x_i) \in J_C \iff U_\sigma \subseteq \bigcup_{i \in \tau} U_i$.
- Type 3: $\prod_{i \in \tau} (1 - x_i) \in J_C \iff X \subseteq \bigcup_{i \in \tau} U_i$.

While the Type 1 pseudo-monomials generate the Stanley-Reisner ideal $I_{\Delta(C)}$ (see Section 1.3), the Type 2 pseudo-monomials capture information about the code that is not contained in $\Delta(C)$. Specifically, for any cover $U$ such that $C = C(U)$, the corresponding Type 2 relationships encode intersections $U_\sigma$ that are fully covered by other sets, a feature that is not reflected in the nerve $N(U) = \Delta(C)$.

Recall that, by convention, we always assume $00 \cdots 0 \in C$ and thus eliminate the Type 3 pseudo-monomials. The canonical form can then be written as:

$$CF(J_C) = CF^1(J_C) \cup CF^2(J_C),$$

where $CF^1(J_C)$ and $CF^2(J_C)$ are disjoint subsets of minimal Type 1 and Type 2 pseudo-monomials, respectively (Section 1.3).

The following useful fact is a direct consequence of [7, Theorem 4.3], and allows us to interpret the elements of $CF(J_C)$ as minimal RF relationships.

Lemma 5.2. $x_\sigma \prod_{i \in \tau} (1 - x_i) \in CF(J_C) \iff (\sigma, \tau)$ is a minimal RF relationship.

Example 5.3 (Example 1.2 continued). Recall the code $C = C(U)$ from Example 1.2. The canonical form is

$$CF(J_C) = \{x_1x_4, x_2(1 - x_1)(1 - x_3), x_2x_4(1 - x_3)\}.$$ 

Using Lemma 5.2 we can immediately read off the minimal RF relationships: $U_1 \cap U_4 = \emptyset$, $U_2 \subseteq U_1 \cup U_3$, and $U_2 \cap U_4 \subseteq U_3$ (cf. Figure 2 and Example 2.1).
5.2 Algebraic characterization of intersection-complete codes

If a code is a simplicial complex, so that $C = \Delta(C)$, then the only elements of $CF(J_C)$ will be Type 1 pseudo-monomials. The converse is also true, giving us an easy algebraic characterization of codes that are simplicial complexes:

**Lemma 5.4.** $C = \Delta(C)$ if and only if $CF(J_C)$ contains only pseudo-monomials $x_{\sigma} \prod_{i \in \tau} (1 - x_i)$, with $|\tau| = 0$ (i.e., only monomials $x_{\sigma}$ appear in $CF(J_C)$).

Recall that $C$ is an $\cap$-complete code if for any pair of codewords $\omega_1, \omega_2 \in C$, we also have $\omega_1 \cap \omega_2 \in C$. In particular, all codes that are simplicial complexes are $\cap$-complete. Because $\cap$-complete codes are always $max \cap$-complete, they are locally good (by C-4 of Theorem 1.8). Moreover, they have also been shown to be convex (C-6 of Theorem 1.8).

We find that $\cap$-complete codes can be characterized via the canonical form as a natural generalization of the description for simplicial complexes, above.

**Proposition 5.5.** A code $C$ is $\cap$-complete if and only if $CF(J_C)$ only contains pseudo-monomials of the form $x_{\sigma} \prod_{i \in \tau} (1 - x_i)$, with $|\tau| \leq 1$.

In other words, $\cap$-complete codes are precisely the codes for which only terms of the form $x_{\sigma}$ and $x_{\sigma}(1 - x_i)$ appear in $CF(J_C)$. In the proof of Proposition 5.5 we will again use the notation $C|_{\sigma}$ for the code obtained from $C$ by restricting all codewords to the neurons in $\sigma$. We will also use the obvious fact that if $C$ is $\cap$-complete, then $C|_{\sigma}$ is $\cap$-complete for any $\sigma \subseteq [n]$.

**Proof.** ($\Rightarrow$) Suppose $C$ is $\cap$-complete, and consider an element $x_{\sigma} \prod_{i \in \tau} (1 - x_i) \in CF(J_C)$. To obtain a contradiction, assume $|\tau| > 1$. For each $i \in \tau$, there exists a codeword $\omega_i \in C|_{\sigma \cup i}$ such that $\sigma \cup i \subseteq \omega_i$, but $j \notin \omega_i$ for any $j \in \tau \setminus \{i\}$. (If there were a $j \in \tau \setminus \{i\}$ such that $j \in \omega_i$ for all possible $\omega_i$, then $i$ could be removed from $\tau$ and $x_{\sigma} \prod_{k \in \tau \setminus \{i\}} (1 - x_k) \in J_C$, contradicting the minimality of $x_{\sigma} \prod_{k \in \tau} (1 - x_k) \in CF(J_C)$.) Since $C$, and hence $C|_{\sigma \cup i}$, is $\cap$-complete, it follows that $\sigma \in C|_{\sigma \cup i}$. But this contradicts the fact that $\sigma$ can only appear in a codeword together with some $i \in \tau$, as indicated by the fact that $x_{\sigma} \prod_{i \in \tau} (1 - x_i) \in CF(J_C)$. We conclude that $|\tau| \leq 1$.

($\Leftarrow$) Suppose $CF(J_C)$ only contains pseudo-monomials with $|\tau| \leq 1$. Consider a pair of codewords $\omega_1, \omega_2 \in C$ and let $\sigma = \omega_1 \cap \omega_2$. Then $x_{\sigma} \notin J_C$, since $x_{\omega_1}, x_{\omega_2} \notin J_C$. To obtain a contradiction, suppose $\sigma \notin C$. This implies $x_{\sigma} \prod_{i \in \tau} (1 - x_i) \notin J_C$ for some $\tau \neq \emptyset$ (since $x_{\sigma} \notin J_C$). It follows that $x_{\sigma} \prod_{i \in \tau} (1 - x_i)$ must be a multiple of a Type 2 pseudo-monomial in the canonical form, say $x_{\sigma'}(1 - x_i) \in CF(J_C)$, where $i \in \tau$ and $\sigma' \subseteq \sigma$ is nonempty. In particular, $x_{\sigma'}(1 - x_i) \in J_C$, which implies that any codeword containing $\sigma$ also contains $i$. This contradicts the fact that $\sigma = \omega_1 \cap \omega_2$. We conclude that $\sigma \in C$, and hence $C$ is $\cap$-complete.

5.3 Algebraic signatures of local obstructions

In this section we prove Theorem 1.10, and relate algebraic signatures in $CF(J_C)$ to receptive field relationships in $RF(C)$. The first statement of Theorem 1.10 is precisely the following proposition, while the rest of the theorem follows directly from Theorem 5.9 below.

**Proposition 5.6.** If there exists $x_{\sigma} \prod_{i \in \tau} (1 - x_i) \in CF^2(J_C)$ such that $Lk_{\sigma}(\Delta|_{\sigma \cup \tau})$ is not contractible, then $C$ is not a convex code.

**Proof.** $x_{\sigma} \prod_{i \in \tau} (1 - x_i) \in CF^2(J_C)$ implies $(\sigma, \tau) \in RF(C)$, by Lemma 5.2 and $\tau \neq \emptyset$. If $Lk_{\sigma}(\Delta|_{\sigma \cup \tau})$ is not contractible, then $(\sigma, \tau)$ is a local obstruction and hence $C$ is not a convex code.

The converse, unfortunately, is not true. In other words, there exist non-convex codes with local obstructions despite the fact that the links $Lk_{\sigma}(\Delta|_{\sigma \cup \tau})$ are contractible for every pair $(\sigma, \tau)$ such that $x_{\sigma} \prod_{i \in \tau} (1 - x_i) \in CF^2(J_C)$. These local obstructions correspond to RF relationships that are not minimal. The smallest example occurs for $n = 4$ neurons:
Example 5.7. Consider the code \( C = \{0000, 1110, 1101, 1011, 0111\} \). This code has \( \Delta(C) = L^27 \), and canonical form

\[
CF(J_C) = \{x_1x_2x_3x_4\} \cup \{x_i(1-x_j)(1-x_k) \mid i, j, k \in [4] \text{ and distinct}\}.
\]

The RF relationships \( (\sigma, \tau) = (\{i\}, \{j, k\}) \) that are detected by the canonical form all have corresponding links \( L_{k_i}(\Delta|_{\{i,j,k\}}) \) that are equivalent to \( L^3 \), and hence contractible. \( C \) is not, however, a convex code. To see this, note that links such as \( L_{k_12}(\Delta) \) and \( L_{k_1}(\Delta) \), corresponding to missing codewords, are non-contractible. Thus, \( C \) has local obstructions and is not convex.

On the other hand, for \( CF^2(J_C) \) elements having \( |\tau| = 1 \), we can safely say that no higher-order local obstructions arise.

Lemma 5.8. Suppose \( x_{\sigma}(1-x_i) \in CF(J_C) \). Then there are no local obstructions \( (\sigma', \tau) \) for any \( \sigma' \supseteq \sigma \) and any \( \tau \) such that \( i \in \tau \).

Proof. Suppose \( x_{\sigma}(1-x_i) \in CF(J_C) \), and \( (\sigma', \tau) \in RF(C) \) for some \( \sigma' \supseteq \sigma \) and \( \tau \neq \emptyset \) such that \( i \in \tau \). We will show that \( (\sigma', \tau) \) is not a local obstruction by proving that \( L_{k_{\sigma'}}(\Delta|_{\sigma'\cup \tau}) \) is contractible, where \( \Delta = \Delta(C) \). Note that \( x_{\sigma}(1-x_i) \in CF(J_C) \) implies that \( i \notin \sigma \), but for every \( \omega \supseteq \sigma \), \( i \in \omega \). Thus, for any \( \omega \supseteq \sigma' \supseteq \sigma \), we have \( i \in \omega \), and so \( i \in \omega \) for any \( \omega \in L_{k_{\sigma'}}(\Delta|_{\sigma'\cup \tau}) \). The link \( L_{k_{\sigma'}}(\Delta|_{\sigma'\cup \tau}) \) is thus a cone, and hence contractible.

Despite the fact that not all local obstructions can be detected directly from the canonical form \( CF(J_C) \), the consequences of Proposition 5.6 allow us to catalog a rich set of algebraic signatures for detecting whether or not a code is locally good. These are summarized in the following theorem.

Note that signature A-2 uses the notation \( \Gamma_{\sigma} \) for the simple graph on vertex set \( \tau \) with edges \( \{ij\} \in \tau \times \tau \mid x_{\sigma}x_ix_j \notin J_C \), and \( G_{\Gamma}(\sigma, \tau) \) for the simple graph on vertex set \( \tau \) with edge set \( \{ij\} \in \tau \times \tau \mid U_{\sigma} \cap U_i \cap U_j \neq \emptyset \).

Theorem 5.9. If \( C \) has any of the algebraic signatures in rows A-1, A-2, or A-3 of Table 4, then \( C \) is not a convex code. If, alternatively, \( C \) has signature A-4 or A-5, then \( C \) is locally good. Codes with signature A-5 are \( \cap \)-complete, and hence convex. Each algebraic signature has a corresponding RF condition, which also implies the given property of \( C \).

<table>
<thead>
<tr>
<th>Algebraic signature</th>
<th>Receptive field condition</th>
<th>Property of ( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>( \exists x_{\sigma}(1-x_i)(1-x_j) \in CF^2(J_C) ) \text{ s.t. } x_{\sigma}x_ix_j \notin J_C</td>
<td>( (\sigma, {i, j}) \in RF(C) ) and ( U_{\sigma} \cap U_i \cap U_j = \emptyset )</td>
</tr>
<tr>
<td>A-2</td>
<td>( \exists x_{\sigma} \prod_{i \in \tau}(1-x_i) \in CF^2(J_C) ) \text{ s.t. } G_C(\sigma, \tau) \text{ is disconnected}</td>
<td>( (\sigma, \tau) \in RF(C) ) and ( G_{\Gamma}(\sigma, \tau) \text{ is disconnected} )</td>
</tr>
<tr>
<td>A-3</td>
<td>( \exists x_{\sigma} \prod_{i \in \tau}(1-x_i) \in CF^2(J_C) ) \text{ s.t. } x_{\sigma}x_{\tau} \in CF^3(J_C)</td>
<td>( (\sigma, \tau) \in RF(C) ) and ( U_{\sigma} \cap U_{\tau} = \emptyset ), but ( U_{\sigma} \cap U_{\tau} \neq \emptyset ) ( \forall \tau' \subseteq \tau )</td>
</tr>
<tr>
<td>A-4</td>
<td>( \forall x_{\sigma} \prod_{i \in \tau}(1-x_i) \in CF^2(J_C), \quad x_{\sigma}x_{\tau} \notin J_C )</td>
<td>( \forall (\sigma, \tau) \in RF(C), \quad U_{\sigma} \cap U_{\tau} \neq \emptyset )</td>
</tr>
<tr>
<td>A-5</td>
<td>( \forall x_{\sigma} \prod_{i \in \tau}(1-x_i) \in CF^2(J_C), \quad</td>
<td>\tau</td>
</tr>
</tbody>
</table>

Table 7: Algebraic signatures and receptive field conditions for convex, non-convex, and locally good codes

For the proof of Theorem 5.9, recall from Lemma 5.2 that \( x_{\sigma} \prod_{i \in \tau}(1-x_i) \in CF^2(J_C) \) implies \( (\sigma, \tau) \in RF(C) \); and also that \( (\sigma, \tau) \in RF(C) \) indicates \( U_{\sigma} \subseteq \bigcup_{i \in \tau} U_i \). It is also useful to remember that \( \mathcal{N}(U_{\sigma} \cap U_i \{i \in \tau\}) = L_{k_{\sigma}}(\Delta|_{\sigma\cup \tau}) \).
Proof. For each signature, A-1, . . . , A-5, we prove that

algebraic signature $\Rightarrow$ RF condition $\Rightarrow$ property of $C$.

(A-1) Clearly, this signature implies $(\sigma, \{i, j\}) \in RF(C)$, and thus $U_\sigma \subseteq U_i \cup U_j$. By Lemma 5.1.x$_{\sigma}x_{\tau}x_{ij} \in J_C$ implies $U_\sigma \cap U_i \cap U_j = \emptyset$. Applying Lemma 2.2 we see that $C$ is non-convex.

(A-2) It is easy to see that if $C = C(U)$, then $G_C(\sigma, \tau) = G_U(\sigma, \tau)$, which equals the 1-skeleton of $Lk_\sigma(\Delta|_{\sigma \cup \tau})$. Since we assume this is disconnected, it follows that $Lk_\sigma(\Delta|_{\sigma \cup \tau})$ is not contractible, and hence $C$ is non-convex by Proposition 5.6.

(A-3) $x_{\sigma}x_{\tau} \in CF^1(J_C)$ implies that $U_\sigma \cap U_\tau = \emptyset$, and $U_{\sigma} \cap U_{\tau'} \neq \emptyset$ for all $\tau' \subseteq \tau$. This means $Lk_\sigma(\Delta|_{\sigma \cup \tau})$ is a hollow simplex, which is not contractible. By Proposition 5.6 $C$ is non-convex.

(A-4) Note that, by Lemma 5.1 $x_{\sigma}x_{\tau} \notin J_C$ implies $U_\sigma \cap U_\tau \neq \emptyset$. Since we assume this is true for all $(\sigma, \tau) \in RF(C)$, we can apply Lemma 2.5 and conclude that $C$ is locally good.

(A-5) If the only elements of $CF^2(J_C)$ are of the form $x_{\sigma}(1 - x_i)$, then we must have that for any RF relationship, $U_\sigma \subseteq \bigcup_{j \in \tau} U_j$, there is an $i \in \tau$ such that $U_\sigma \subseteq U_i$. The converse is also true, since $CF^2(J_C)$ reflects only minimal RF relationships. Thus, the algebraic signature is equivalent to the corresponding RF condition. It follows from Proposition 5.5 that signature A-5 is equivalent to $C$ being an $\cap$-complete code.

We end this section by noting that the graph $G_C(\sigma, \tau)$, appearing in signature A-2, can be constructed directly from the canonical form. This is because the condition $x_{\sigma}x_{\tau}x_{ij} \in J_C$ is easy to check from knowledge of $CF^1(J_C)$, provided we already know that $x_{\sigma} \prod_{k \in \tau}(1 - x_k) \in CF^2(J_C)$.

Lemma 5.10. Suppose $x_{\sigma} \prod_{k \in \tau}(1 - x_k) \in CF^2(J_C)$. Then for any $i, j \in \tau$,

$$x_{\sigma}x_{\tau}x_{ij} \in J_C \iff x_{\sigma}x_{\tau}x_{ij} \in CF^1(J_C) \text{ for some } \emptyset \subseteq \sigma' \subseteq \sigma.$$

Proof. The backwards direction ($\Leftarrow$) is obvious. To see the forward direction ($\Rightarrow$), suppose $x_{\sigma}x_{\tau}x_{ij} \in J_C$. Then it is a multiple of some monomial $x_\rho \in CF^1(J_C)$. We have four possibilities:

1. $\rho \subseteq \sigma$. Then $x_\rho$ divides $x_{\sigma} \prod_{k \in \tau}(1 - x_k)$, contradicting that $x_{\sigma} \prod_{k \in \tau}(1 - x_k) \in CF(J_C)$.
2. $\rho = \sigma' \cup \{i\}$ for some $\sigma' \subseteq \sigma$. Then $x_{\sigma}x_{\tau} \prod_{k \in \tau \setminus \{i\}}(1 - x_k) \in J_C$, and hence $x_{\sigma}x_{i} \prod_{k \in \tau \setminus \{i\}}(1 - x_k) + x_{\sigma} \prod_{k \in \tau}(1 - x_k) = x_{\sigma} \prod_{k \in \tau \setminus \{i\}}(1 - x_k)$, contradicting that $x_{\sigma} \prod_{k \in \tau}(1 - x_k) \in CF(J_C)$.
3. $\rho = \sigma' \cup \{j\}$. This argument is identical to the previous one, leading to a contradiction. Thus, the only possibility left is:
4. $\rho = \sigma' \cup \{i, j\}$, so there must be some $\emptyset \subseteq \sigma' \subseteq \sigma$ with $x_{\sigma'}x_{\tau}x_{ij} \in CF(J_C)$. \qed

6 Bounds on the minimal embedding dimension of convex codes

We now turn to the problem of determining the minimal embedding dimension $d(C)$ of a convex code $C$, as defined in Section 4.2. There is no general method for computing $d(C)$, though bounds can be obtained from the information present in the simplicial complex $\Delta(C)$. In this section, we review known results on $d$-representability, Helly’s theorem, and the Fractional Helly theorem, and apply them to obtain lower bounds on $d(C)$. These bounds rely solely on features of the code captured by $\Delta(C)$, and do not take into account the finer structure of the code. Nevertheless, as we have seen in Section 4.1 the presence or absence of a single codeword can have a significant effect on $d(C)$, even if the simplicial complex $\Delta = \Delta(C)$ is fixed (e.g., see Table 3 for L8, L14, L16, etc.). It remains an open question how to use this additional information in order to improve the bounds on $d(C)$. 

26
6.1 Embedding dimension and \(d\)-representability

As mentioned above, the problem of determining the values of \(d\) for which a convex code \(C\) has an embedding in \(\mathbb{R}^d\) does not appear to have been addressed in the literature. In contrast, the related problem of determining when a simplicial complex \(\Delta\) can be realized as the nerve \(\mathcal{N}(\mathcal{U})\) of a cover \(\mathcal{U}\) has received considerable attention (see [16, 20] and references therein). A simplicial complex \(\Delta\) is said to be \(d\)-representable if there exists a collection of convex (not necessarily open) sets \(\mathcal{U} = \{U_1, \ldots, U_n\}\), with \(U_i \subset \mathbb{R}^d\), such that \(\Delta = \mathcal{N}(\mathcal{U})\). Note that for such a \(\mathcal{U}\), the corresponding code \(C(\mathcal{U})\) need not be equal to \(\Delta\), though it is always true that \(\Delta(C(\mathcal{U})) = \Delta\).

Observe that if a code \(C\) can be embedded in \(\mathbb{R}^d\) via some collection of convex sets \(\mathcal{U}\), then \(\Delta(C) = \mathcal{N}(\mathcal{U})\), and so \(\Delta(C)\) must be \(d\)-representable. This motivates us to define the nerve dimension \(d_{\mathcal{N}}(C)\) of a code \(C\) to be the minimal \(d\) such that \(\Delta(C)\) is \(d\)-representable. It immediately follows that

\[ d_{\mathcal{N}}(C) \leq d(C). \]

In other words, lower bounds on \(d\)-representability for \(\Delta(C)\) automatically give lower bounds on \(d(C)\). Upper bounds on \(d_{\mathcal{N}}(C)\) do not typically provide any information on \(d(C)\), except of course when \(C = \Delta(C)\). In this case, we can apply the following construction to show that \(d(\Delta) \leq n - 1\).

**Proof of C-7 in Theorem 1.8.** In [16, Section 3.1], Tancer describes a construction that realizes any simplicial complex \(\Delta\) on \([n]\) as the intersection patterns of an arrangement of closed convex sets in \(\mathbb{R}^{n-1}\) (for \(n > 1\)). It is straightforward to check that open neighborhoods of these closed sets form a convex realization for the code \(C = \Delta\). Thus, any code on \(n > 1\) neurons with \(C = \Delta(C)\) is convex, and satisfies \(d(C) \leq n - 1\).

It should be noted that the upper bound \(d(\Delta) \leq n - 1\) is tight. For any \(n\), if \(\Delta\) is the hollow simplex on \(n\) vertices, then we must have \(d(\Delta) \geq n - 1\). It is also known that every \(k\)-dimensional simplicial complex on \(n\) elements is \((2k + 1)\)-representable [20, 21]. Thus, for any code \(C\) on \(n\) neurons such that \(\Delta(C)\) has dimension \(k\), we have

\[ d_{\mathcal{N}}(C) \leq \min\{2k + 1, n - 1\}. \]

This upper bound on \(d\)-representability does not give a bound on \(d(C)\), but indicates that for any code \(C\) there exists another neural code \(\tilde{C}\), with matching simplicial complex \(\Delta(\tilde{C}) = \Delta(C)\), such that \(d(\tilde{C}) \leq \min\{2k + 1, n - 1\}\).

6.2 Bounds from Helly’s theorem

One common tool used for addressing \(d\)-representability of simplicial complexes, and thus for giving lower bounds on \(d(C)\), is Helly’s theorem.

**Theorem 6.1** (Helly’s theorem [22]). Let \(\mathcal{U} = \{U_1, \ldots, U_n\}\) be a collection of convex open sets in \(\mathbb{R}^d\). If for every \(d + 1\) sets in \(\mathcal{U}\), the intersection is non-empty, then the full intersection \(\bigcap_{i=1}^{n} U_i\) is non-empty.

In the language of representability, Helly’s theorem states that if \(\Delta\) is \(d\)-representable and \(\Delta\) contains all possible \(d\)-dimensional faces, then \(\Delta\) must be the full simplex. Conversely, if \(\Delta\) contains all possible \(d\)-dimensional faces but \(\Delta\) is not the full simplex, then \(\Delta\) is not \(d\)-representable. This immediately yields a proof for Proposition 1.12 illustrating that the presence or absence of a single codeword can have a large effect on \(d(C)\).

\(^6A\) simplicial complex is said to be \(k\)-dimensional if \(k\) is the maximum dimension of any face in \(\Delta\); i.e., the largest subset in \(\Delta\) has size \(k + 1\).
Proof of Proposition 6.2. In the first case, where $11 \cdots 1 \in \mathcal{C}$, the fact that $\mathcal{C}$ is convex and $d(C) \leq 2$ follows from C-8 of Theorem 1.8. For the second case, where $11 \cdots 1 \notin \mathcal{C}$, suppose $\mathcal{C}$ is realizable as a convex code in $\mathbb{R}^d$ for some $d \leq k$, so that $\mathcal{C} = \mathcal{C}(U)$ for some collection of convex open sets $U = \{U_1, U_2, \ldots, U_n\}$, with each $U_i \subset \mathbb{R}^d$. Since, by hypothesis, $\Delta(\mathcal{C})$ contains all $k$-dimensional faces, it also contains all $d$-dimensional faces, and so the intersection of every collection of $d+1$ subsets in $U$ is non-empty. Thus, by Helly’s Theorem, the full intersection of all sets in $U$ is non-empty, and so $11 \cdots 1 \notin \mathcal{C}$; hence, we must have $d(C) > k$. 

We can also apply Helly’s theorem to every subcollection $\{U_{1i}, \ldots, U_{im}\} \subset U$, or equivalently to the induced subcomplex on elements $i_1, \ldots, i_m$, to see that if all the $d$-dimensional faces of this subcomplex are present, then the top-dimensional face must also be present in order for $\Delta$ to be $d$-representable. This leads us to the following definitions. A simplicial complex is said to contain an induced $k$-dimensional simplicial hole if it contains $k+1$ vertices such that the induced subcomplex on those vertices is isomorphic to a hollow simplex [16]. We define the Helly dimension $d_H(\mathcal{C})$ of $\mathcal{C}$, denoted $d_H(\mathcal{C})$, to be the dimension of the largest induced simplicial hole of $\Delta(\mathcal{C})$:

$$d_H(\mathcal{C}) \stackrel{\text{def}}{=} \max\{k \mid \Delta(\mathcal{C}) \text{ has a } k\text{-dimensional induced simplicial hole}\}.$$ 

Clearly, $d_H(\mathcal{C}) \leq d(\mathcal{C})$. The Helly dimension is particularly useful because it can be detected from the canonical form $\text{CF}(J_C)$, and is equal to the lower bound on $d(\mathcal{C})$ that was given in [7]:

**Lemma 6.2.** Let $\mathcal{C}$ be a neural code with canonical form $\text{CF}(J_C)$. Then

$$d_H(\mathcal{C}) = \max_{x_\sigma \in \text{CF}(J_C)} |\sigma| - 1.$$ 

**Proof.** Let $\sigma \subseteq [n]$ satisfy $x_\sigma \in \text{CF}(J_C)$. Then, by the minimality of generators in $\text{CF}(J_C)$, we have that $\sigma \notin \Delta(\mathcal{C})$ but all subsets of $\sigma$ are in $\Delta(\mathcal{C})$. Thus, $\Delta(\mathcal{C})$ has an induced simplicial hole of dimension $|\sigma| - 1$, and so $d_H(\mathcal{C}) = \max_{x_\sigma \in \text{CF}(J_C)} |\sigma| - 1$. 

### 6.3 Bounds from the Fractional Helly theorem

Given the immediate bounds on $d(\mathcal{C})$ obtained from Helly’s theorem, it is natural to ask whether extensions of this theorem could be used to improve the bounds. The Fractional Helly theorem is a well-known extension that indeed provides new bounds.

**Theorem 6.3** (Fractional Helly theorem, Theorem 6.7 of [16]). Let $\alpha > 0$, and $U = \{U_1, U_2, \ldots, U_n\}$ be a collection of convex open sets in $\mathbb{R}^d$ such that at least $\alpha \left( \begin{array}{c} n \vspace{1mm} \\ d+1 \end{array} \right)$ of the $(d+1)$-tuples of sets in $U$ have non-empty intersections. Then there exists $\sigma \subseteq [n]$ such that $|\sigma| > \frac{\alpha}{d+1} n$, and $\bigcap_{i \in \sigma} U_i$ is non-empty.

Similar to Helly’s theorem, the Fractional Helly theorem shows that if a code $\mathcal{C}$ can be embedded in $\mathbb{R}^d$, and the simplicial complex $\Delta(\mathcal{C})$ has many $d$-dimensional faces, then $\Delta(\mathcal{C})$ must have some sufficiently high-dimensional face. The following lemma quantifies these observations in our context.

**Lemma 6.4.** Let $\Delta$ be a $k$-dimensional simplicial complex on $n$ elements, and let $f_d(\Delta)$ be the number of $d$-dimensional faces in $\Delta$ for $1 \leq d < n$. If $\Delta$ is $d$-representable, then $k + 1 > f_d(\Delta)/\left( \begin{array}{c} n \vspace{1mm} \\ d \end{array} \right)$.
Proof. By definition of \( d \)-representable, we have \( \Delta = \mathcal{N}(U) \) for some \( U = \{ U_1, U_2, \ldots, U_n \} \), where each \( U_i \subset \mathbb{R}^d \). Since each \( d \)-dimensional face of \( \Delta \) corresponds to an intersection of \( (d+1) \) of the \( U_i \)'s, we have that \( f_d(\Delta(C)) \) of the \( (d+1) \)-tuples have non-empty intersections. By the Fractional Helly theorem, there is some \( \sigma \subseteq [n] \) with \( |\sigma| > \frac{\alpha}{d+1} n \) such that \( \bigcap_{i \in \sigma} U_i \neq \emptyset \), where \( \alpha = \frac{f_d(\Delta(C))}{(d+1)} \). Since \( \Delta(C) \) is \( k \)-dimensional, it follows that \( |\sigma| \leq k+1 \), and so \( k+1 > \frac{\alpha}{d+1} n = f_d(\Delta(C))/(n-1) \). \( \square \)

This leads us to the following definition. Let \( C \) be a code on \( n \) neurons with a \( k \)-dimensional simplicial complex \( \Delta(C) \), and let \( f_d(\Delta(C)) \) be the number of \( d \)-dimensional faces in \( \Delta(C) \) for \( 1 \leq d < n \). The Fractional Helly dimension \( d_{FH}(C) \) of \( C \) is given by:

\[
d_{FH}(C) \overset{\text{def}}{=} 1 + \max \left\{ d \left| f_d(\Delta(C)) \geq (k+1) \cdot \binom{n-1}{d}, \ 1 \leq d < n \right. \right\}.
\]

As with the Helly dimension, \( d_{FH}(C) \) can be computed from the canonical form \( \text{CF}(J_C) \), but with a bit more effort because we must compute the \( f \)-vector \( f_d(\Delta(C)) \). Recall from Section 1.3 that the ideal generated by the Type 1 relations of \( \text{CF}(J_C) \) is the Stanley-Reisner ideal \( I_{\Delta(C)} \subset \mathbb{F}_2[x_1, \ldots, x_n] \); thus, \( \text{CF}(J_C) \) allows for a direct computation of \( I_{\Delta(C)} \). It is well known that \( f_d(\Delta(C)) \) is the \( (d+1) \)th coefficient in the \( \mathbb{Z}_{\geq 0} \)-graded Hilbert polynomial of the Stanley-Reisner ring \( \mathbb{F}_2[x_1, \ldots, x_n]/I_{\Delta(C)} \). This provides a procedure for computing \( f_d(\Delta(C)) \) directly from the Type 1 relations in \( \text{CF}(J_C) \), using standard computational algebra software such as Macaulay2 [8].

### 6.4 Comparison of dimension bounds

We have now seen three dimensions related to the minimal embedding dimension \( d(C) \): the nerve dimension \( d_N(C) \), the Helly dimension \( d_H(C) \), and the Fractional Helly dimension \( d_{FH}(C) \). How do these dimensions compare?

First, note that although a simplicial complex \( \Delta \) cannot be represented in any dimension less than its Helly dimension \( d_H(\Delta) \), Helly’s theorem does not guarantee that \( \Delta \) is \( d_{FH}(\Delta) \)-representable. Thus we have

\[
d_H(C) \leq d_N(C) \leq d(C).
\]

The next example gives a \( \Delta \) such that \( d_H(C) < d_N(C) \) for any code \( C \) such that \( \Delta = \Delta(C) \). This shows that the nerve dimension can provide a stronger lower bound than the Helly dimension.

**Example 6.5.** Consider any code \( C \) such that \( \Delta(C) \) is the simplicial complex in Figure 6a. We obtain \( d_H(C) = 1 \) because the two maximal induced simplicial holes of \( \Delta(C) \) arise from the subsets \{1, 3\} and \{2, 4\}, which both have dimension 1. Although \( \Delta(C) \) is contractible, the induced subcomplex on \{1, 2, 3, 4\} is a 1-cycle, so \( \Delta(C) \) is at best 2-representable and \( d_N(C) \geq 2 \). Thus, \( d_N(C) > d_H(C) \). Figure 6b shows that \( d(C) = 2 \).

Next, we turn to the Fractional Helly dimension. By a similar argument as above, we have

\[
d_{FH}(C) \leq d_N(C) \leq d(C),
\]

but again there may be a gap between \( d_{FH}(C) \) and \( d_N(C) \). Given that both \( d_H(C) \) and \( d_{FH}(C) \) give lower bounds on \( d(C) \), it is natural to ask which of these lower bounds is tighter and how these lower bounds compare. The following example shows that it is possible for \( d_H(C) > d_{FH}(C) \) or \( d_H(C) < d_{FH}(C) \), depending on the code.

**Example 6.6.** (a) Let \( n = 3 \), and consider the code \( C_1 = \{ 000, 110, 101, 011 \} \), whose simplicial complex \( \Delta(C_1) \) is the empty triangle, as shown in Figure 7a. This is the boundary of a 2-simplex, so we have \( d_H(C_1) = 2 \). In contrast, a quick computation yields \( d_{FH}(C_1) = 1 \), and so \( d_H(C_1) > d_{FH}(C_1) \).

(b) Let \( n = 2r \), where \( r \geq 4 \), and suppose \( \Delta(C_2) = K_{r,r} \), the complete bipartite graph on \( 2r \) vertices, as shown in Figure 7b. This graph contains no triangles, so the largest induced simplicial holes result
Figure 8: (a) A simplicial complex $\Delta$, with facets 125, 235, 345, 145. For any $C$ such that $\Delta(C) = \Delta$, the Helly dimension $d_H(C) = 1$, while the nerve dimension $d_N(C) = 2$. (b) A convex realization of $C$ in $\mathbb{R}^2$, showing that $d(C) = 2$.

For any $C$ such that $\Delta(C) = \Delta$, the Helly dimension $d_H(C) = 1$, while the nerve dimension $d_N(C) = 2$. (b) A convex realization of $C$ in $\mathbb{R}^2$, showing that $d(C) = 2$.

from missing edges in $\Delta(C_2)$, which have dimension 1. Thus, $d_H(C_2) = 1$. To compute $d_{FH}(C_2)$, we first find the $f$-vector for $\Delta(C_2)$. Observe that $f_0(\Delta(C_2)) = n$, $f_1(\Delta(C_2)) = r^2$, and $f_i(\Delta(C_2)) = 0$ for $2 \leq i < n$. Note also that $k = 1$, since $\Delta(C_2)$ is 1-dimensional. Since $f_d(\Delta(C_2)) = 0$ for $d \geq 2$, the inequality $f_d(\Delta(C_2)) \geq (k + 1) \cdot \binom{n-1}{d}$ is not satisfied for these values of $d$. However, we have

$$f_1(\Delta(C_2)) = r^2 \geq 2(2r - 1) = 2(n - 1) = (k + 1) \cdot \binom{n-1}{1},$$

with the inequality being valid for all $r \geq 4$. Thus, directly from the definition, we find that $d_{FH}(C_2) = 2$. Hence, $d_{FH}(C_2) > d_H(C_2)$.

Figure 9: (a) The simplicial complex of code $C_1$ from Example 6.6. (b) The simplicial complex of code $C_2$ from Example 6.6 for $r = 4$.

Putting together the two inequalities resulting from the Helly-type theorems, we obtain the following proposition about the minimal embedding dimension of a code.

**Proposition 6.7.** Let $C$ be a convex neural code on $n$ neurons. The minimal embedding dimension satisfies:

$$d(C) \geq \max\{d_H(C), d_{FH}(C)\},$$

where

$$d_H(C) = \max_{x_\sigma \in CF(J_C)} |\sigma| - 1,$$

$$d_{FH}(C) = 1 + \max \left\{ d \left| f_d(\Delta(C)) \geq (k + 1) \cdot \binom{n-1}{d}, 1 \leq d < n \right\}. $$

30
Acknowledgments

This work began at a 2014 AMS Mathematics Research Community, “Algebraic and Geometric Methods in Applied Discrete Mathematics,” which was supported by NSF DMS-1321794. CC was supported by NSF DMS-1225666/DMS-1537228 and an Alfred P. Sloan Research Fellowship; EG was supported by NSF DMS-1304167; and AS was supported by NSF DMS-1004380 and NSF DMS-1312473. We thank Chad Giusti, Vladimir Itskov, Carly Klivans, William Kronholm, Keivan Monfared, and Yan X Zhang for numerous discussions, and Eva Pastalkova for providing the data used to create Figure.

7 Appendix

In this appendix, we show the equivalence of the parity check ideal $J_H$ and the neural ideal $J_C$ for linear codes. Recall that a neural code is linear if it forms a subspace over $\mathbb{F}_2$, as described in Section 1.3. This subspace can then be defined as the nullspace of some (non-unique) matrix, and such a matrix is known as a parity check matrix $H$ for the code.

Definition 7.1. Given a linear code $C$ with parity check matrix $H$, the parity check ideal is given by

\[ J_H \overset{\text{def}}{=} \langle H_i \mathbf{x} \rangle, \]

where $H_i \mathbf{x}$ denotes the polynomial corresponding to the $i^{th}$ row of $H$, which is obtained by multiplying the $i^{th}$ row by the column vector $\mathbf{x}$ of the indeterminates $x_1, \ldots, x_n$.

Example 7.2. Consider the parity check matrix $H = [1 0 1]$. This matrix dictates that every codeword must have the binary sum of its first and third entries equal to 0; the code is then the 2-dimensional subspace of $\mathbb{F}_2^3$ whose vectors satisfy this property, so $C(H) = \{000, 010, 101, 111\}$. Note that $H_1 \mathbf{x} = x_1 + x_3$, and so $J_H = \langle x_1 + x_3 \rangle$.

To eliminate spurious polynomials that evaluate to zero on all of $\mathbb{F}_2^n$, we move to working over the Boolean ring $\mathbb{F}_2[x_1, \ldots, x_n]/\mathcal{B}$, where $\mathcal{B} = \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$. Over this ring, we see that despite the fact that the parity check ideal $J_H$ is defined in terms of linear polynomials, $J_H$ is also generated by pseudo-monomials and precisely agrees with the neural ideal $J_C$, as defined in Section 1.3.

Proposition 7.3. Let $C$ be a binary linear code with associated parity check matrix $H$. Then

\[ J_H = J_C \]

over the Boolean ring $\mathbb{F}_2[x_1, \ldots, x_n]/\mathcal{B}$. In particular,

(i) For each $v \in \mathbb{F}_2^n \setminus C$, $\chi_v = (\chi_v)(H_i \mathbf{x})$ for some row $H_i$ of $H$, and

(ii) $H_i \mathbf{x} = \sum_{v \in \mathcal{V}_i} \chi_v$, for $\mathcal{V}_i \overset{\text{def}}{=} \{ v \in \mathbb{F}_2^n \mid H_i v = 1 \}$,

where $\chi_v$ is the indicator pseudo-monomial for $v$, as defined in Section 1.3.

Proof. We first prove (i) and (ii), as these together will easily imply the main result. To prove (i), first observe that by definition of $H$ as a parity check matrix, for any non-codeword $v$ there exists some row $H_i$ such that $H_i v = 1$. For all $w \in \mathbb{F}_2^n$, $w \neq v$, evaluating both polynomials in (i) at $w$ yields the same result: $\chi_v(w) = 0$ and $(\chi_v)(H_i \mathbf{x})(w) = \chi_v(w)[(H_i \mathbf{x})(w)] = 0 \cdot H_i w = 0$. Additionally, evaluating both at $v$ gives $\chi_v(v) = 1$ and $(\chi_v)(H_i \mathbf{x})(v) = \chi_v(v) H_i v = 1$. Thus, both polynomials agree on all vectors in $\mathbb{F}_2^n$, so their difference is 0 on all vectors, which over the Boolean ring $\mathbb{F}_2[x_1, \ldots, x_n]/\mathcal{B}$ implies their difference must be the zero polynomial. We conclude that $\chi_v = (\chi_v)(H_i \mathbf{x})$ for some row $H_i$ of $H$.

To prove (ii), we again evaluate both polynomials on all vectors $w \in \mathbb{F}_2^n$. Since $(H_i \mathbf{x})(w) = H_i w$, we see $(H_i \mathbf{x})(w) = 1$ precisely when $w \in \mathcal{V}_i$ and otherwise it evaluates to 0. For all $w \in \mathcal{V}_i$, $\sum_{v \in \mathcal{V}_i} \chi_v(w) = 1$.
since $\chi_w$ appears in the sum exactly once, and for all $w \notin \mathcal{V}_i$, $\sum_{v \in \mathcal{V}_i} \chi_v(w) = 0$. Thus the two polynomials agree on all vectors in $\mathbb{F}_2^n$, so by the same argument as above they must be equal.

Item (i) immediately implies that $(\chi_v \mid v \in \mathbb{F}_2^n \setminus \mathcal{C}) \subseteq J_H$, since each generator of the first ideal is a multiple of a generator of $J_H$. The reverse containment follows from item (ii), when we observe that $\mathcal{V}_i \subseteq \{v \mid v \in \mathbb{F}_2^n \setminus \mathcal{C}\}$ since $\mathcal{C} = \{c \in \mathbb{F}_2^n \mid H_ic = 0 \text{ for all } i\}$. Thus, $J_H = (\chi_v \mid v \in \mathbb{F}_2^n \setminus \mathcal{C})$, which is precisely $J_{\mathcal{C}}$. □

References


