The Tangled Web They Weave: Exploring Neural Representations with Directed Simplicial Complexes

Michael Bleher
Institute for Mathematics
Heidelberg University
Heidelberg, Germany
mbleher@mathi.uni-heidelberg.de

Abstract

We propose a method to construct filtered directed simplicial complexes (DSCs) from the internal states of artificial neural networks (ANNs). Our approach treats ANNs as structural causal models and uses counterfactual reasoning, based on internal ablation studies, to infer directed causal relationships between neuron activations within specific input contexts. Our findings suggest that the resulting complexes provide an explicit representation of distributed feature representations, offering insights into their superposition at polysemantic neurons. We apply our method to a multilayer perceptron (MLP) on a synthetic quadrant classification task, illustrating how DSCs capture the network's logic for encoding and combining elementary features. Furthermore, we apply our method to an MLP trained on the MNIST dataset, identifying distinct topological signatures within the DSCs that reflect learned feature compositions for different digit classes. This framework offers a novel approach to characterising neural representations, with potential applications in feature disentanglement and interpretability.

1 Introduction

Understanding how information propagates and organises within artificial neural networks (ANNs) is a central challenge in interpretability research. While ANNs demonstrably encode semantic information through intricate patterns of neuron activations (Alain and Bengio, 2017, Kim et al., 2018), how these arise and interact remains poorly understood. Methods that can identify and charactrise these patterns may offer deep insights into how ANNs represent abstract concepts and perform complex computations.

A central challenge in understanding this *neural code* arises from the efficient encoding of numerous features on a finite number of neurons. Often, features are *distributed* across the neural net, relying on coordinated activity across neuronal ensembles rather than single neurons (Hinton, 1986, Olshausen and Field, 1997, Pouget et al., 2000). Moreover, in many situations distributed features overlap significantly and are found to be in *superposition* (Elhage et al., 2022). As a consequence, neural networks typically exhibit *polysemanticity*, where individual neurons may activate for multiple, distinct features (Olah et al., 2017, 2020).

Algebraic topology provides several tools for quantifying the shapes and properties of neural representations. Since neural networks—both biological and artificial—are fundamentally directed systems, approaches from *directed topology* are particularly promising. Such methods have previously been used to investigate brain microcircuitry using directed simplicial complexes (Reimann et al., 2017) and to quantify information flow in ANNs via path homology on network-derived graphs (Chowdhury and Mémoli, 2018, Chowdhury et al., 2019).

Preprint. Under review.

In this work, we introduce a novel method to construct *filtered directed simplicial complexes* (DSCs) $\mathcal{K}_X$ from the activation profiles of a pre-trained feedforward multilayer perceptron (MLP) for a given set of input data $X = \{x_i\}_{i \in I}$. The internal neurons of the network form the vertices of these complexes. Directed edges and higher-order simplices (e.g., an ordered tuple (v_0, v_1, v_2) representing a directed 2-simplex) are constructed by inferring (higher-order) *causal relationships* between neurons in the context of the sample X, using the counterfactual framework introduced by Pearl and Shafer (1995). More concretely, we utilise a form of coherent counterfactual ablation study on the internal neurons of the network. Intuitively, if an input X activates multiple features, ablating the group of neurons $P = \{v_0, \ldots, v_{k-1}\}$ supporting one feature while enforcing the networks constraints should primarily affect other neurons v_k within that same feature's processing pathway. However, if features are superimposed onto overlapping sets of polysemantic neurons, ablating such neurons, or groups containing them, is expected to influence multiple pathways concurrently. By systematically testing the impact of ablating progressively larger sets of neurons P (singletons, pairs, triples, etc.) on target neurons v_k, we identify directed k-simplices $\sigma = (v_0, \ldots, v_k)$ representing these functional dependencies. The filtration value assigned to σ quantifies the minimal coordinated reduction of activations over P required to induce a significant change in the activation of the target neuron v_k.

We propose that the input-specific DSCs $\mathcal{K}_X$ can be viewed as a concrete instantiation of the distributed features that are present in the input data X. More concretely, we expected that $\mathcal{K}_X$ reveals organisational principles of features in X and their interactions in the network's task, that may be obscured in neuron (co-)activation data on their own. Moreover, we hypothesised that, on the one hand, polysemantic neurons will often manifest as vertices of simplex intersections in $\mathcal{K}_X$, reflecting their role in processing or integrating several of X features into the network's task, and on the other hand, will appear as vertices of distinct simplices across DSCs associated to unrelated samples $\mathcal{K}_X$, $\mathcal{K}_Y$. Indeed, we found that when the data X contains multiple features, then the resulting simplicial complex $\mathcal{K}_X$ shows a high degree of similarity with the superposition of simplicial complexes $\mathcal{K}_i$ associated to various single-feature samples X_i.

In summary, our main contributions are:

- We propose a novel methodology to construct filtered directed simplicial complexes $\mathcal{K}_X$ from the activation data of MLPs with respect to input data X. The construction is grounded in Pearl's framework of causal inference, yielding representations of functional, context-dependent neuronal interactions rather than static architectural connectivity.
- We conduct experiments on handcrafted synthetic classifiers, designed to exhibit controlled monosemantic and polysemantic behaviour, and on an MLP trained for MNIST digit classification. These experiments demonstrate that the complexes $\mathcal{K}_X$ are distinct, class-specific summaries of the networks activation state.
- We analyse the resulting directed topological structures and investigate how they capture feature interactions and reflect the network's representational strategy (e.g., polysemanticity vs. monosemanticity). Our findings suggest that the earliest simplices in $\mathcal{K}_X$ capture functional components of the network, while later simplices may reflect the network's ability to combine and generalise features.

2 Related work

Polysemanticity in ANNs is a well-documented challenge for interpretability, extensively explored in foundational work by Olah et al. (2017, 2020). Hypotheses regarding its origins point to network capacity limitations (Elhage et al., 2022) or incidental overlap during initialisation or training (Lecomte et al., 2024). Efforts to disentangle feature representations include techniques like probing for linearly decodable information (Alain and Bengio, 2017), identifying concept-aligned directions in activation space (Kim et al., 2018), or extracting feature-specific activation patterns through sparse autoencoders (Cunningham et al., 2023).

The observation of distributed representations, polysemanticity, and superposition in ANNs resonates with the phenomenon of 'mixed selectivity' in neuroscience, where individual neurons respond nonlinearly to conjunctions of multiple stimuli (Rainer et al., 1998, Rigotti et al., 2013, Fusi et al., 2016). This conjunctive coding is thought to create high-dimensional, flexible neural codes beneficial for complex tasks, e.g. (Jazayeri and Movshon, 2006, Warden and Miller, 2007, Barak et al., 2013). In

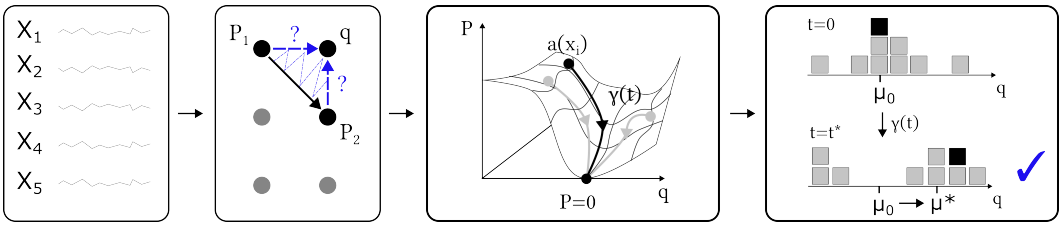

Figure 1: Schematic overview of the methodology for constructing a filtered Directed Simplicial Complex (DSC) $\mathcal{K}_X$ from neural network activations provided by an input sample X. (1st panel) An input sample $X = \{x_i\}_{i\in I}$ is processed by a pre-trained MLP, yielding a set of internal activation profiles $\{a(x_i)\}_{i\in I}$. (2nd panel) The internal neurons provide the vertex set $\{v_0, \ldots, v_N\}$ of the complex. For a candidate simplex σ whose faces are all in the complex, we ask whether the simplex can be extended to $\sigma \cup \{q\}$ by a target neuron q. (3rd panel) For a given set of neurons $P = \{v_0, \ldots, v_{k-1}\}$, we generate coherent counterfactual trajectories $\gamma_{a(x_i),P}(t)$ for each $x_i \in X$ by progressively ablating the activations of neurons in P. (4th panel) We track the activations of the target neuron q along these trajectories and compare them to their baseline values (at $t = 0$). A statistically significant change in q's activation across the sample X indicates a directed causal link $P \to q$. The candidate simplex $\sigma \cup \{q\}$ is added to the complex and assigned a filtration value t^*, corresponding to the minimal extent of P's ablation (parametrised by t) required to induce the significant change in q. Lower f_σ values indicate more immediate or stronger functional interactions within the context of X.

recent years, topological Data Analysis (TDA), and in particular persistent homology, has been used to investigate complex coding schemes in biology by characterising population responses (Singh et al., 2008), functional brain networks (Petri et al., 2014), intrinsic geometric encodings (e.g., tori (Gardner et al., 2022, Giusti et al., 2015)), and neural state spaces (Kang et al., 2021). Simplicial complexes were used to model combinatorial aspects of neural codes and distributed information processing in (Curto, 2017), as well as higher-order interactions in neural data (Chung et al., 2025, Santoro et al., 2024). Similar analyses have revealed directed simplicial structures in brain microcircuits (Reimann et al., 2017) .

Building on this, TDA has also been applied to the study of ANNs in various ways, see Ballester et al. (2024) for a recent review. Applications include analysing weight space evolution (Gabella, 2021), comparing architectures using simplicial complexes (Pérez-Fernández et al., 2021, Watanabe and Yamana, 2022), using topological features of decision boundaries to relate data complexity to network capacity and generalisation (Guss and Salakhutdinov, 2018), and investigating how the topology of data changes as it progresses through the network (Naitzat et al., 2020). Moreover, persistent homology has revealed structural effects of adversarial inputs (Gebhart et al., 2019), characterised CNN internal representations (Carlsson and Gabrielsson, 2018), tracked topological evolution during training, correlating it with generalisation (Gutiérrez-Fandiño et al., 2021), and was shown to predict the generalization gap when calculated for a graph of neuron activation correlations (Ballester et al., 2023). Finally, directed simplicial complexes have previously been constructed from sequential activation paths in ANNs, where it was shown that their path homology distinguishes network states and correlates with network performance (Chowdhury et al., 2019).

3 Background

3.1 Neural activations and feature encodings

Consider a feedforward multilayer perceptron (MLP) with L layers, indexed $l = 0, \ldots, L$. Layer l comprises n_l neurons. The input layer is $l = 0$, with activation vector $a^{(0)} = x \in \mathbb{R}^{n_0}$. For hidden layers ($0 < l < L$) and the output layer ($l = L$), the activation vectors $a^{(l)} \in \mathbb{R}^{n_l}$ are then computed recursively via

$$a^{(l)} = \sigma(W^{(l)} a^{(l-1)} + b^{(l)}), \quad \text{for } 0 < l < L \tag{1}$$

$$a^{(L)} = \sigma_{\text{out}}(z^{(L)}), \tag{2}$$

where $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ are the weights, $b^{(l)} \in \mathbb{R}^{n_l}$ are the biases, and σ is a non-linear, element-wise activation function (e.g., ReLU). The output activation function σ_{out} is typically distinct from internal ones (e.g., softmax for classification, or identity). The overall network computes the function $F(x) = a^{(L)}$.

Our analysis focuses on the internal state of the network when evaluated on a given input x, represented by the concatenated activation vector of all hidden neurons:

$$a(x) = (a^{(1)T}, a^{(2)T}, \ldots, a^{(L-1)T})^T \in \mathbb{R}^N,\ l = 1, \ldots, L-1, \tag{3}$$

where $N = \sum_{l=1}^{L-1} n_l$ is the total number of hidden neurons. For a fixed network, the set of attainable internal activation vectors a forms a subset $\mathcal{M} \subset \mathbb{R}^N$ often refered to as the *neural manifold* (Stopfer et al., 2003).

The vector $a(x) \in M$ embodies the network's internal representation of the input x and encodes the features and their interactions that are relevant to the network's task. Representations of isolated features are usually *polytopic*[1], i.e. distributed across a wide range of (typically polysemantic) neurons (Hinton, 1986). It is generally hypothesised that the activation vector $a(x)$ of a complex input is the result of a superposition of multiple, interacting feature representations (Elhage et al., 2022). Our main goal is to investigate this hypothesis, by constructing an algebraic model of polytopic features that reflects the causal links between neuron activations in a given context of inputs $X = \{x_i\}_{i \in I}$.

3.2 Structural causal models and causal inference

To reason about the influence of one neuron on another within the network's operational constraints, we rely on Pearl's framework for causal inference Pearl and Shafer (1995). This framework utilizes Structural Causal Models (SCMs), which consist of a Directed Acyclic Graph (DAG) where nodes correspond to (random) variables X and edges denote direct causal relationships, together with structural equations of the form $X = f_X(\text{Parents}(X))$ specifying how the value of X depends on the value of its parent nodes and exogenous noise. MLPs map naturally onto (deterministic) SCMs: the architecture is a DAG, neuron activations $a_i^{(l)}$ are endogenous variables, the functions f_X are determined by $W^{(l)}, b^{(l)}, \sigma$, and inputs x are the exogenous variables U_X.

To investigate the causal effects between variables, Pearl's framework introduces two key tools for causal reasoning: *interventions* and *counterfactuals*. An *intervention* assigns a fixed value to a chosen variable while removing the influence of any upstream variables that would otherwise determine X. This simulates the effect of controlled manipulations in the real world to test a causal model; if an intervention on X results in a change in some downstream variable Y, one can infer a structural causal effect of X on Y. For example, in an ANN, interventions on individual neurons would reveal a causal link to all downstream neurons in the network, thus reproducing the network's underlying causal structure.

In this article we instead rely on *counterfactuals*, which explore hypothetical outcomes of individual observations under altered conditions (antecedents): "Given the observed values of endogenous variables, what would have happened to the value of Y if X had been x'?". Counterfactual reasoning proceeds in three steps: *Abduction* identifies the most likely values of the exogeneous variables that produced the observed outcome, the *Action* step modifies the value of specified endogenous variables, and *Prediction* propagates these altered values through the SCM. The resulting counterfactual state then serves as the basis for subsequent reasoning – for example, identifying covarying neurons that collectively encode a common polytopic feature.

3.3 Filtered directed simplicial complexes

To represent the inferred causal relationships between groups of neurons, potentially reflecting polytopic features, we utilise *filtered directed simplicial complexes* (DSCs).

Let V be a set, typically called the set of vertices or nodes. A *directed k-simplex* is an ordered $(k+1)$-tuple $\sigma = (v_0, \ldots, v_k)$ of distinct vertices V. A *face* of $\sigma = (v_0, \ldots, v_k)$ is a directed

[1]Derived from the Greek *poly* (many) and *topos* (place). We sometimes prefer this term over 'distributed representation' as it better highlights the duality between the neuron-centric *polysemanticity* and the feature-centric *polytopicity*.

j-simplex $\tau = (v_{i_0}, \ldots, v_{i_j})$ with $j < k$ and indices $0 \leq i_0 < \cdots < i_j \leq k$ that form a proper subset of $\{0, \ldots, k\}$. A *directed simplicial complex* $\mathcal{K}$ on V is a collection of directed simplices closed under taking faces (i.e., if $\sigma \in K$, any ordered subsequence $(v_{i_0}, \ldots, v_{i_j})$ with $i_0 < \cdots < i_j$ is also in $\mathcal{K}$). Vertices (v_i) are 0-simplices, and directed edges $v_i \to v_j$ correspond to 1-simplices (v_i, v_j).

A *filtration* on $\mathcal{K}$ assigns a real number f_σ to each simplex $\sigma \in K$, satisfying $f_\tau \leq f_\sigma$ whenever τ is a face of σ. This induces a family of nested subcomplexes $\mathcal{K}_t = \{\sigma \in K \mid f_\sigma \leq t\}$ indexed by the filtration value $t \in \mathbb{R}$.

4 Methods

We propose a method to construct, for a given set of inputs $X = \{x_i\}_{i \in I}$, a filtered directed simplicial complex $\mathcal{K}_X$ (subsection 3.3) that models the distributed representation of X's features in a trained MLP. Conceptually, our construction probes collective activations patterns in $a(X)$ by studying the effect of ablating a chosen set of neurons on the rest of the network.

A key issue in probing superimposed polytopic features is to respect the intrinsic operational constraints of the network, i.e. to remain on the neural manifold $\mathcal{M} \subset \mathbb{R}^N$ (subsection 3.1). Standard causal probes, like interventions or counterfactual manipulations that modify the activation of neurons (e.g., setting $a_j(x) = 0$), disregard these constraints (subsection 3.2). This may strongly distort the feature representation in $a(x)$. For instance, altering a single neuron within a polytopic feature might deform its coordinated activity beyond the networks recognition, while ablating a polysemantic neuron could inadvertently affect all features and their interaction. For this reason, we rely on *coherent counterfactuals* – hypothetical activation states on or near $\mathcal{M}$.

The construction of the filtered DSC $\mathcal{K}_X$ involves three main stages, detailed in the upcoming sections:

1. **Generate coherent counterfactual trajectories (4.1):** For each observed activation $a(x)$, $x \in X$, and a given set of predictor neurons P, generate an on-manifold trajectory $\gamma_{a(x),P}(t)$ starting at $a(x)$, along which the activations in P progressively decrease. This acts as a coherent, on-manifold ablation study, quantifying the effect of gradually removing the contribution of P while remaining consistent with the network manifold $\mathcal{M}$.
2. **Infer directed causal links and filtration values (4.2):** Analyse a target neuron q's activation along the ablation trajectories $\{\gamma_{a(x),P}(t)\}_{x \in X}$. Infer a directed link $P \to q$ if there is a statistically significant directional change of q's activations across the whole input sample, denoting the time-to-significance by t^*.
3. **Construct the filtered DSC (4.3):** Assemble $\mathcal{K}_X$ by including simplices $\sigma = (v_0, \ldots, v_k)$ based on the link $P = \{v_0, \ldots, v_{k-1}\} \to v_k$ and with filtration value t^*, while enforcing the filtration property $f_\tau \leq f_\sigma$ for faces τ.

The resulting complex $\mathcal{K}_X$ captures directed causal relationships between neuron activations, which reflect the network's feature representations and their interaction in the context of X.

4.1 Generating counterfactual activation trajectories

Our procedure for generating on-manifold trajectories $\gamma_{a,P}(t) \subset M$ corresponds to gradient-descent of the neuron activations in P pulled back to the input space.

More concretely, for a given activation profile $a(x)$, we iteratively compute a sequence of inputs $x(t_k)$ at discrete time steps $t_k = k\Delta t$ for which the activations of neurons in P progressively decrease. At each step t_k, we first identify the subset $P'_k \subseteq P$ of predictor neurons that are still active and write $m = |P'_k|$ for the number of active neurons. If $m = 0$, the process terminates. Otherwise, we demand that in the next step the activations in P'_k decrease according to the unit vector $u_k := \frac{da_{P'_k}}{dt} = -\frac{1}{\sqrt{m}}\mathbf{1}_m$. The velocity $v(t_k) = \frac{dx(t_k)}{dt}$ that implements this decrease is given implicitely by the chain rule $\frac{da_{P'_k}}{dt} = J(t_k)v(t_k)$, where $J(t_k) = \frac{da_{P'_k}}{dx}$ denotes the Jacobian of the neural net. Based on this, we calculate the input velocity $v(t_k)$ as the minimum-norm solution of $u_k = J(t_k)v(t_k)$, which ensures the change has the smallest possible magnitude, keeping the

trajectory of inputs as close to the original x and minimizing perturbations of unrelated features. Finally, the input is updated to $x(t_{i+1}) = x(t_k) + v(t_k) \cdot \Delta t$.

This process generates a sequence of activations $\{a(t_k)\}$ that approximates the desired trajectory $\gamma_{a,P}(t)$, continuing until all activations in P vanish.

4.2 Inferring directed causal links and filtration values

We use the counterfactual trajectories $\{\gamma_{a(x),P}(t) \mid x \in X\}$ (subsection 4.1) to infer directed causal links $P \to q$ and assign an associated filtration value t^*.

Concretely, for a given predictor set P, target neuron q, and discrete time step $t > 0$, we perform a two-sided paired Wilcoxon signed-rank test comparing q's observed activation $a_q = \gamma_{a,P}(0)|_q$ with its activation in the counterfactual state $\gamma_{a,P}(t)$. The test is performed on the collection of activation differences $\Delta_q(t) = \{\gamma_{a(x),P}(0)|_q - \gamma_{a(x),P}(t)|_q\}_{x \in X}$ calculated across the batch X and compares the null hypothesis H_0 that the median of these differences is zero against the alternative H_1 that it is non-zero. Let $p(t)$ be the p-value obtained from this test. A causal link $P \to q$ is inferred if the null hypothesis is rejected ($p(t) < \alpha$, where α is a chosen significance level, typically $\alpha = .05$).

The filtration value assigned to $P \to q$ is the time-to-significance $t^*(P \to q) = \min\{t \mid p(t) < \alpha\}$. If the test never reaches significance, we set $t^*(P \to q) = \infty$. This procedure is applied for all potential candidate links $P \to q$ that arise in the construction of the directed simplicial complex.

Rejection of the null hypothesis suggests that, in response to the perturbation of P implemented in $\{\gamma_{a(x),P}(t) \mid x \in X\}$, the activations of q show a coherent directional shift across the whole batch. The filtration value t^* quantifies the 'latency' or 'immediacy' of this link: a smaller t^* indicates a more immediate influence (detectable after only a small perturbation of P), whereas a larger t^* suggests a more delayed influence. A small t^* suggests that the collection of neurons $P \cup q$ acts mostly in unison within the context provided by the batch X, reflecting their joint participation in representing features or executing computations relevant to the processed inputs.

4.3 Construction of the filtered directed simplicial complex

We construct a filtered directed simplicial complex $\mathcal{K} = \mathcal{K}_X$ based on directed causal links between neuron activations and their associated times to significance t^* in the context of an input sample X (Section 4.2). The 0-simplices $v \in \mathcal{K}$ are given by the internal neurons of the neural net, with filtration value $f_v = 0$. The full complex $\mathcal{K}$ is constructed inductively: A directed k-simplex $\sigma = (v_0, \ldots, v_k)$, $k \geq 1$, is included in the complex $\mathcal{K}_t$ if and only if:

(a) All of its proper faces $\tau \in \partial\sigma$ are elements of $\mathcal{K}_t$.

(b) The directed causal link $P = \{v_0, \ldots, v_{k-1}\} \to v_k$ has $t^*(P \to v_k) \leq t$.

Note that, as a result of these two conditions, the filtration value of σ satisfies $f_\sigma = \max(\{f_\tau \mid \tau \in \partial\sigma\},\ t^*(P \to v_k))$, such that $\mathcal{K}_t$ is a filtered simplicial complex.

The filtration value t^* assigned to each simplex $\sigma = (v_0, \ldots, v_k)$ in a DSC $\mathcal{K}_X$ quantifies the minimal coordinated perturbation of its face $P = \{v_0, \ldots, v_{k-1}\}$ required to significantly affect the final vertex v_k. Lower t^* values indicate a higher *immediacy* of the functional link. We dub the subcomplex $\mathcal{K}_{\Delta t}$ the *immediate subcomplex*, where Δt is the step size in the trajectory generation, corresponding to the smallest perturbation considered in constructing $\mathcal{K}$.

5 Results

We investigated the capacity of filtered directed simplicial complexes (DSCs) to model feature representations and their interactions within pre-trained neural networks, focusing on synthetic classifiers and an MLP trained on MNIST. To quantitatively compare overall DSC structures, we employed the Jaccard index $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, applied to their sets of simplices.

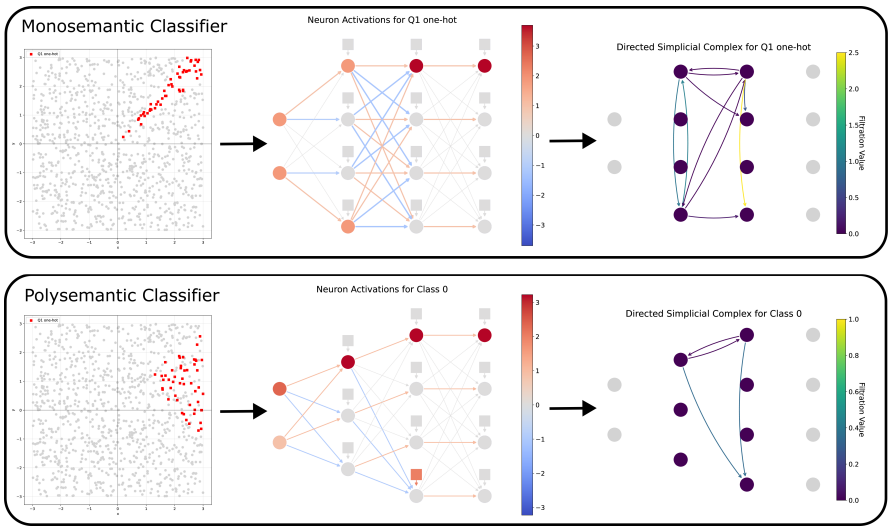

Figure 2: Illustrative results from synthetic quadrant classifiers. Illustrative example of one-hot encoded data points, average neuron activation, and 1-skeleton of $\mathcal{K}_{Q1}$ in the monosemantic classifier (Top) and polysemantic classifier (Bottom). The monosemantic classifier shows an immediate subcomplex that contains 1-simplices and 2-simplices (not-shown) involving the vertices $\{N0, N3, N4\}$, which are related to the networks conjunctive links, while $N5$ and $N6$ are added at a later filtration value. The polysemantic classifier consists of a simpler complex, with 1-simplices involving the vertices $\{N0, N3, N7\}$.

5.1 DSCs reveal encoding strategies and feature superposition

We experimented with two handcrafted MLPs for 2D quadrant classification: a 'monosemantic' [2-4-4-4] MLP where first hidden layer (HL1) neurons (N0,N1) encode the sign x $(+, -)$ and (N2,N3) the sign of y, and a 'polysemantic' [2-3-4-4] MLP with a bottlenecked HL1 (N0-N2) (subsection A.1). We constructed DSCs for single-quadrant, two-quadrant (adjacent and diagonal), and random input samples.

Monosemantic Classifier For single-quadrant contexts like $\mathcal{K}_{Q1}$, the immediate complex (4.3) contained functionally relevant 1-simplices, e.g., $(N0, N4)$ and $(N3, N4)$, linking HL1 feature neurons ($x > 0, y > 0$) to the Q1-output N4 ($x > 0 \wedge y > 0$) (Figure 2). The 2-simplices for the quadrant's core logic (e.g., on $\{N0, N3, N4\}$) formed shortly thereafter at low t values (e.g., the first involving these vertices at $t = 0.09$ in $\mathcal{K}_{Q1}$). Inactive HL1 neurons often remained isolated or formed simplices only at much higher t. For inputs combining features from two *adjacent* quadrants (e.g., $\mathcal{K}_{Q3+Q4}$), $\mathcal{K}_{\Delta t}$ already included numerous 1- and 2-simplices (e.g., $(N1, N2, N6)$) for individual quadrant processing, and much later ($t = 3.54$) an encompassing 4-simplex was created. Similarly, for *diagonally opposite* quadrants (e.g., $\mathcal{K}_{Q1+Q3}$), $\mathcal{K}_{\Delta t}$ contained 1- and 2-simplices for individual features (e.g., $(N0, N4)$; $(N0, N1, N4)$), while maximal 3-simplices completed at intermediate t values ($t \approx 1.17 - 1.94$). Random input DSCs ($\mathcal{K}_{\text{rand}}$) displayed a dense $\mathcal{K}_{\Delta t}$ (e.g., 2-simplex $(N0, N3, N4)$), but their 3-simplices generally lacked the specific low-t high-dimensional structures of defined multi-feature contexts. Jaccard indices comparing subcomplexes at various t values further quantify these distinct assembly dynamics (Appendix A.1.4), confirming that primary designed pathways are captured as high-immediacy interactions.

Polysemantic Classifier The polysemantic classifier's DSCs showed more varied and complex structures (Figure 2). For single-quadrant contexts, the immediate subcomplex varied significantly: $\mathcal{K}_{Q1}$ and $\mathcal{K}_{Q4}$ were sparse, with only 1-simplices appearing immediately and a maximal 2-simplices only completing much later ($t \approx 0.5$). In contrast, $\mathcal{K}_{Q2}$ and $\mathcal{K}_{Q3}$ exhibited a much denser immediate subcomplex, containing multiple 1- and 2-simplices (e.g., $(N0, N1, N3)$, $(N0, N3, N4)$), with their 3-simplices forming at subsequent low t values ($t \approx 0.32 - 0.41$). For inputs combining features from two *adjacent* quadrants, high-dimensional structures representing superposition often formed rapidly. For instance, in $\mathcal{K}_{Q2+Q3}$, the entire 4-simplex on $\{N1, N2, N4, N5\}$ was present

at $t = 0.01$. Similarly, for $\mathcal{K}_{Q1+Q2}$, a 3-simplex on $\{N0, N1, N3, N4\}$ was in $\mathcal{K}_{\Delta t}$, with the full 4-simplex involving N6 completing later ($t \approx 0.54 - 0.76$). Contexts for *diagonally opposite* quadrants, like $\mathcal{K}_{Q1+Q3}$, had an immediate subcomplex containing 1- and 2-simplices (e.g., $(N0, N2, N5)$), with further 2-simplices appearing by $t \approx 0.68$. The random input DSC ($\mathcal{K}_{\text{rand}}$) consisted immediately of 1-, 2-, 3-, and even two distinct 4-simplices (on $\{N0, N1, N2, N3, N4\}$ and $\{N0, N1, N3, N4, N5\}$), indicating widespread, immediate coordination. Jaccard indices highlighted these complex, context-dependent assembly patterns (Appendix A.1.4). Notably, the frequent participation of HL1 neurons (N0-N2) in distinct simplices for different contexts (e.g., $\{N0, N1, N3, N4\}$ in $\mathcal{K}_{Q2}$ vs. $\{N0, N5, N6\}$ in $\mathcal{K}_{Q4}$) supports our hypothesis that polysemantic neurons are in the intersection of superimposed polytopic features.

5.2 Class-specific topological signatures in MNIST

To assess our method's applicability to a standard machine learning benchmark, we analysed an MLP trained on the MNIST handwritten digit classification task. The network architecture was [784-128-64-32-16-10] ($N = 240$ internal neurons). For each of the 10 digit classes ($c = 0, \ldots, 9$), we selected 20 representative input images $\mathcal{X}_c$ from the training set that elicited strong, unambiguous classifications. Using these, we constructed class-specific DSCs, $\mathcal{K}_{\mathcal{X}_c}$ up to a maximum simplex dimension of 2. Additionally, we constructed DSCs for contexts combining pairs of visually similar or dissimilar digits (e.g., $\mathcal{K}_{\mathcal{X}_{1+5}}$, $\mathcal{K}_{\mathcal{X}_{1+8}}$) to investigate feature superposition. Further details are provided in subsection A.2.

The resulting DSCs revealed complex, high-dimensional structures that were nonetheless sparse subsets of all theoretically possible interactions. For a typical single-digit context (e.g., $\mathcal{K}_{\mathcal{X}_1}$), we observed approximately 7,700 1-simplices ($\approx 10\%$) and over 360,000 2-simplices ($\approx 1\%$) among the set of directed simplices on $N = 240$ internal neurons. These numbers, while substantial, suggest a certain degree of specificity of the learned functional organisation. The vast majority of simplices (approximately 97%) appeared at the lowest filtration value (here $t = 0.1$), forming a dense 'background' of immediate functional links, while higher filtration values corresponded to progressively sparser and more specific interactions.

Structurally, analysis of the 1-skeleton in typical DSCs (e.g., for digit '0') revealed a core-periphery organisation. A large neuronal periphery rarely initiated edges, while a smaller core of neurons sourced most 1-simplices. Within this core, out-degree variability was moderate (a factor of approximately 3 between minimum and maximum), lacking a single dominant 'super-hub'. Edges predominantly connected core neurons internally or extended from core to periphery neurons. Core neurons displayed varied roles based on filtration values: some participated broadly in low-filtration simplices (suggesting immediate functional response), while others were prominent in more selective, high-filtration interactions (possibly indicating specialised processing). We observed a high ratio of 2-simplices to 1-simplices (e.g., $\sim$48 triangles per edge in $\mathcal{K}_{\mathcal{X}_0}$), which seems to be mostly due to the tendency of causal links to point in both directions.

The Jaccard similarity between DSCs associated to individual digits were typically below 0.1. The Jaccard index between '4' and '9', two easily confused digits, was found to have the maximum value of 0.17, while visually disimilar digits like '1' and '0' showed the smallest value of 0.02. While this trend is not generally consistent, it suggests that the network may have learned to identify digits using overlapping sets of features that are not shared by disimilar digits. For approximately two-hot inputs there were often a mix of high and low similarities (e.g., $J(\mathcal{K}_{\mathcal{X}_{1+5}}, \mathcal{K}_{\mathcal{X}_1}) = 0.07$ and $J(\mathcal{K}_{\mathcal{X}_{1+5}}, \mathcal{K}_{\mathcal{X}_5})0.26$).

6 Discussion

We introduced a method to construct filtered directed simplicial complexes (DSCs) from neural network activations using coherent counterfactuals derived from Pearl's causal inference framework. The simplices in the DSCs represent on-manifold functional dependencies, where filtration values highlight the immediacy of these multi-neuronal links, offering a novel view of information organisation in ANNs.

Experiments on synthetic classifiers (subsection 5.1) demonstrated that DSCs mirror distinct encoding strategies (monosemantic vs. polysemantic) and provide topological evidence of feature superposition,

as quantified by simplex structures and Jaccard indices. Analysis of an MNIST MLP (subsection 5.2) showed that DSCs form input-specific functional architectures, with variations in simplex counts and Jaccard similarities suggesting differential feature sharing and discrimination (e.g., between visually similar vs. dissimilar digits).

Our framework facilitates a shift from neuron-centric analyses to characterising distributed features and their interactions as algebraic or combinatoric objects. The causally-informed directed topological analysis offers a new, interpretable window into the functional organisation of neural networks, complementing existing methods by focusing on emergent, higher-order interaction patterns.

Limitations

Our method has several limitations, regarding both construction and interpretation. A primary limitation is the computational expense: constructing k-dimensional DSCs over N neurons scales roughly as $\mathcal{O}(N^{k+1})$, making it challenging for large networks or high k. Methodologically, DSC quality depends on representative input samples X (requiring reliable classification or labelling) and is sensitive to hyperparameters (e.g., significance threshold α, trajectory step size Δt). Furthermore, the construction of counterfactuals relies on the assumption that the suggested gradient-descent based ablation adequately explores relevant on-manifold states. Finally, the current causal inference lacks explicit false-discovery rate control, potentially admitting spurious relationships.

The empirical validation is confined to MLPs on synthetic data and MNIST, while contemporary interpretability often focuses on large language models (transformer architecture), complicating direct comparison with state-of-the-art approaches. Most critically, interpreting the often complex, high-dimensional DSCs remains a challenge, though topological summaries provide a promising starting point for future quantitative work.

Acknowledgments and Disclosure of Funding

Acknowledgements This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster) and finalized with the support of the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (project PEPS, no. 101071786). The author acknowledges the use of compute ressources from de.NBI Cloud at the Zentrum für Datenverarbeitung of the University of Tübingen and the German Federal Ministry of Education and Research (BMBF) through grant no 031 A535A. I thank Freya Jensen for helpful discussions.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. 2017.

Rubén Ballester, Xavier Arnal Clemente, Carles Casacuberta, Meysam Madadi, Ciprian A. Corneanu, and Sergio Escalera. Predicting the generalization gap in neural networks using topological data analysis. 2023.

Rubén Ballester, Carles Casacuberta, and Sergio Escalera. Topological Data Analysis for Neural Network Analysis: A Comprehensive Survey. 2024.

Omri Barak, Mattia Rigotti, and Stefano Fusi. The Sparseness of Mixed Selectivity Neurons Controls the Generalization–Discrimination Trade-Off. *J. Neurosci.*, 33, 2013.

Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological Approaches to Deep Learning. 2018.

Samir Chowdhury and Facundo Mémoli. Persistent Path Homology of Directed Networks. In *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1152–1169. Society for Industrial and Applied Mathematics, 2018.

Samir Chowdhury, Thomas Gebhart, Steve Huntsman, and Matvey Yutin. Path homologies of deep feedforward networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1077–1082, 2019.

Moo K. Chung, Anass B. El-Yaagoubi, Anqi Qiu, and Hernando Ombao. From Density to Void: Why Brain Networks Fail to Reveal Complex Higher-Order Structures. 2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. 2023.

Carina Curto. What can topology tell us about the neural code? *Bull. Amer. Math. Soc.*, 54, 2017.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy Models of Superposition. 2022.

Stefano Fusi, Earl K Miller, and Mattia Rigotti. Why neurons mix: High dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37, 2016.

Maxime Gabella. Topology of Learning in Feedforward Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32, 2021.

Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. Toroidal topology of population activity in grid cells. *Nature*, 602, 2022.

Thomas Gebhart, Paul Schrater, and Alan Hylton. Characterizing the Shape of Activation Space in Deep Neural Networks. 2019.

Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov. Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences*, 112, 2015.

William H. Guss and Ruslan Salakhutdinov. On Characterizing the Capacity of Neural Networks using Algebraic Topology. 2018.

Asier Gutiérrez-Fandiño, David Pérez-Fernández, Jordi Armengol-Estapé, and Marta Villegas. Persistent Homology Captures the Generalization of Neural Networks Without A Validation Set. 2021.

Geoffrey E. Hinton. Learning Distributed Representations of Concepts. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 8, 1986.

Mehrdad Jazayeri and J. Anthony Movshon. Optimal representation of sensory information by neural populations. *Nat Neurosci*, 9, 2006.

Louis Kang, Boyan Xu, and Dmitriy Morozov. Evaluating State Space Discovery by Persistent Cohomology in the Spatial Representation System. *Front. Comput. Neurosci.*, 15, 2021.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR, 2018. URL `https://proceedings.mlr.press/v80/kim18d.html`.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014.

Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. What Causes Polysemanticity? An Alternative Origin Story of Mixed Selectivity from Incidental Causes. 2024.

Yann LeCun, Corinna Cortes, and given-i=CJ family=Burges, given=CJ. MNIST handwritten digit database. *ATT Labs [Online].*, 2, 2010.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2018. URL `http://arxiv.org/abs/1802.03426`.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. 2020.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2, 2017.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. *Distill*, 5, 2020.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37, 1997.

Judea Pearl and Glenn Shafer. Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Synthese-Dordrecht*, 104, 1995.

G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, and F. Vaccarino. Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface*, 11, 2014.

A. Pouget, P. Dayan, and R. Zemel. Information processing with population codes. *Nat Rev Neurosci*, 1, 2000.

David Pérez-Fernández, Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, and Marta Villegas. Characterizing and Measuring the Similarity of Neural Networks with Persistent Homology. 2021.

G. Rainer, W. F. Asaad, and E. K. Miller. Selective representation of relevant information by neurons in the primate prefrontal cortex. *Nature*, 393, 1998.

Michael W. Reimann, Max Nolte, Martina Scolamiero, Katharine Turner, Rodrigo Perin, Giuseppe Chindemi, Paweł Dłotko, Ran Levi, Kathryn Hess, and Henry Markram. Cliques of Neurons Bound into Cavities Provide a Missing Link between Structure and Function. *Front. Comput. Neurosci.*, 11, 2017.

Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497, 2013.

Andrea Santoro, Federico Battiston, Maxime Lucas, Giovanni Petri, and Enrico Amico. Higher-order connectomics of human brain function reveals local topological signatures of task decoding, individual identification, and behavior. *Nat Commun*, 15, 2024.

G. Singh, F. Memoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D. L. Ringach. Topological analysis of population activity in visual cortex. *Journal of Vision*, 8, 2008.

Mark Stopfer, Vivek Jayaraman, and Gilles Laurent. Intensity versus Identity Coding in an Olfactory System. *Neuron*, 39, 2003.

Melissa R. Warden and Earl K. Miller. The representation of multiple objects in prefrontal neuronal delay activity. *Cereb Cortex*, 17 Suppl 1, 2007.

Satoru Watanabe and Hayato Yamana. Topological measurement of deep neural networks using persistent homology. *Ann Math Artif Intell*, 90, 2022.

A Technical Appendices and Supplementary Material

A.1 Detailed Configuration of Synthetic Quadrant Classifiers

This section provides the detailed configurations for the handcrafted Multilayer Perceptrons (MLPs) used in the synthetic quadrant classification experiments, as summarised in Section 5.1 of the main paper. The experiments were conducted using PyTorch for model definitions. All computations for Directed Simplicial Complex (DSC) construction were performed using CPU-based parallel processing. The MNIST calculation was performed on the de.NBI Cloud at the Zentrum für Datenverarbeitung of the University of Tübingen.

A.1.1 Experimental Setup

Input Data Generation For both classifiers, input data X consisted of 1000 2D points sampled uniformly from the square $[-3, 3] \times [-3, 3]$. True class labels y corresponded to the four quadrants: Q1 ($x \geq 0, y \geq 0$), Q2 ($x < 0, y \geq 0$), Q3 ($x < 0, y < 0$), and Q4 ($x \geq 0, y < 0$). The networks used ReLU activation functions for all hidden layers. The output layer activations (logits) were taken directly after the final linear transformation for analysis and sample selection.

Selection of Representative Input Samples for DSC Construction For constructing DSCs, subsets of 50 input points were selected for various contexts based on the network's output logits $F(x)$. The selection was based on ranking by cosine similarity to target (one-hot or two-hot) vectors to ensure that the points represent a learned feature of the network.

- **Single-Quadrant (One-Hot) Contexts:** For each quadrant q_i, $i \in \{0, 1, 2, 3\}$, a one-hot vector h_i was defined to have coefficient i equal to one and otherwise zero. The 50 input points whose network output logits $F(x)$ had the highest L^2-normalised cosine similarity to the L^2-normalised h_q were selected to form $\mathcal{X}_{q_i}$.
- **Two-Quadrant (Two-Hot) Contexts:** For each pair of distinct quadrants (q_i, q_j), a two-hot vector h_{ij} (with entries for i and j active) was defined. The 50 input points whose network output logits $F(x)$ had the highest L^2-normalised cosine similarity to the L^2-normalised h_{ij} were selected to form $\mathcal{X}_{q_i+q_j}$. The specific pairs considered were all 6 combinations: (Q1,Q2), (Q1,Q3), (Q1,Q4), (Q2,Q3), (Q2,Q4), (Q3,Q4).
- **Random Context:** 50 input points were selected uniformly at random from the 1000 generated data points to form $\mathcal{X}_{\text{rand}}$.

Directed Simplicial Complex Construction Parameters and Computational Resources For each set of selected samples, a DSC was constructed with the following parameters:

- maximal dimension: 8 (theoretical maximum on $N = 8$ neurons).
- step size $\Delta t = 0.01$ (trajectory generation, see 4.1).
- maximal number of iterations: 1000 (trajectory generation, see 4.1).
- Statistical significance α: 0.01 (Wilcoxon signed-rank test, see 4.2).

Due to the small size of these networks ($N = 8$ internal neurons for monosemantic, $N = 7$ for polysemantic) and the limited number of probe data points (50), the construction of each DSC, even up to dimension 8, was computationally inexpensive, typically completing within a few minutes on a standard four-core CPU Laptop. Memory usage was negligible.

A.1.2 Monosemantic Quadrant Classifier Configuration

The monosemantic quadrant classifier was a [2-4-4-4] MLP. It has 2 input neurons, 4 neurons in the first hidden layer (HL1), 4 neurons in the second hidden layer (HL2), and 4 output neurons. All biases were set to zero.

The weights for the first hidden layer ($W^{(1)} \in \mathbb{R}^{4\times 2}$) were:

$$W^{(1)} = \begin{pmatrix} 1.0 & 0.0 \\ -1.0 & 0.0 \\ 0.0 & -1.0 \\ 0.0 & 1.0 \end{pmatrix}. \tag{4}$$

These weights, with ReLU, allow HL1 neurons N0, N1, N2, N3 to monosemantically activate for the featurse $x > 0$, $x < 0$ (via $-x > 0$), $y < 0$ (via $-y > 0$), and $y > 0$, respectively.

The weights for the second hidden layer ($W^{(2)} \in \mathbb{R}^{4\times 4}$) were:

$$W^{(2)} = \begin{pmatrix} 1.0 & -1.25 & -1.25 & 1.0 \\ -1.25 & 1.0 & -1.25 & 1.0 \\ -1.25 & 1.0 & 1.0 & -1.25 \\ 1.0 & -1.25 & 1.0 & -1.25 \end{pmatrix}. \tag{5}$$

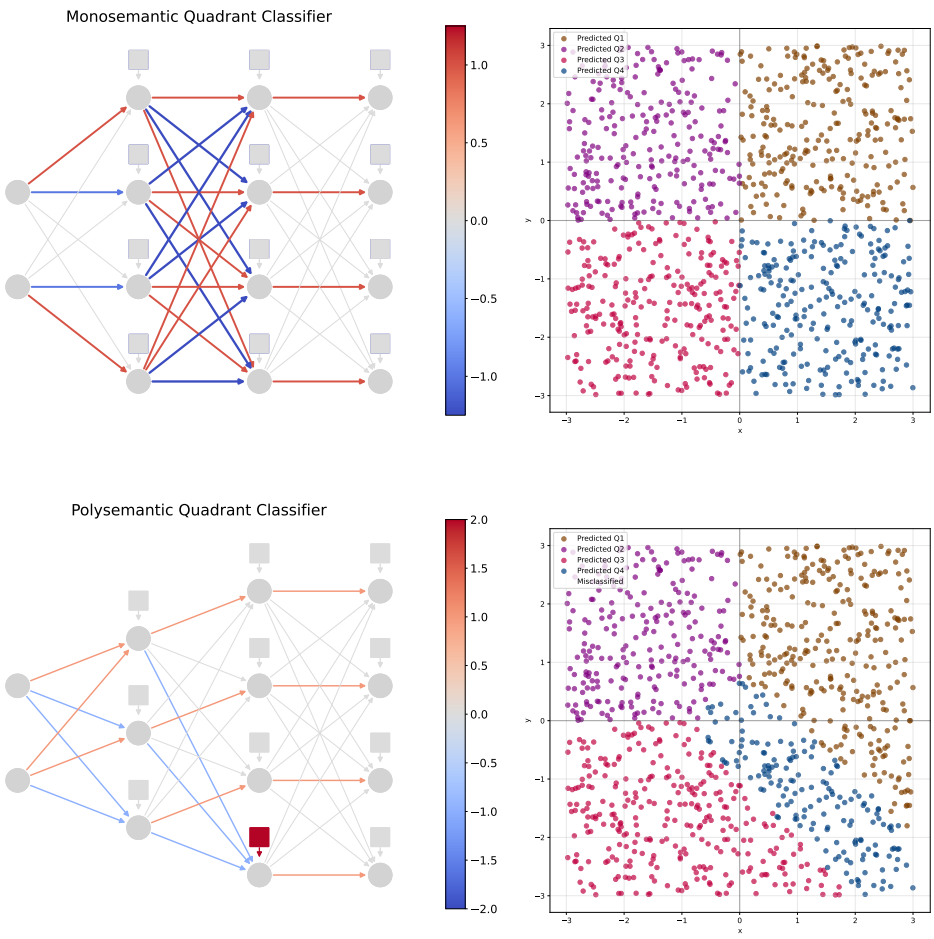

Figure 3: Network architectures and classification performance for synthetic quadrant classifiers. (Top Left) Monosemantic [2-4-4-4] MLP architecture. (Top Right) Classification of input space by the monosemantic MLP. (Bottom Left) Polysemantic [2-3-4-4] MLP architecture. (Bottom Right) Classification of input space by the polysemantic MLP.

This layer provides a (piecewise) linear implementation of conjunctions of HL1 features: e.g., HL2 neuron N4 (first row) activates for Q1 if N0 ($x > 0$) and N3 ($y > 0$) are active. Negative weights inhibit contributions from other quadrant-defining features. The output layer weights ($W^{(\text{out})} \in \mathbb{R}^{4\times4}$) were an identity matrix $\mathbf{I}_4$. The network architecture and its classification of the input space are shown in Figure 3 (top row).

A.1.3 Polysemantic Quadrant Classifier Configuration

The polysemantic quadrant classifier was a [2-3-4-4] MLP. It has 2 input neurons, 3 neurons in HL1, 4 neurons in HL2, and 4 output neurons. The reduced HL1 width promotes polysemantic representations.

The weights for the first hidden layer ($W^{(1)} \in \mathbb{R}^{3\times2}$) were:

$$W^{(1)} = \begin{pmatrix} 1.0 & 1.0 \\ -1.0 & 1.0 \\ -1.0 & -1.0 \end{pmatrix}. \tag{6}$$

Biases for HL1 were zero. These weights, with ReLU, map the corners of the unit square centered at $(0, 0)$ to the corners of a tetrahedron in 3d, necessitating the coactivation of two neurons to represent three of the four edges, as well as a coordinated death of all neurons for the fourth edge, thus implementing a polysemantic representation of the input space.

The weights ($W^{(2)} \in \mathbb{R}^{4\times 3}$) and biases ($b^{(2)} \in \mathbb{R}^4$) for HL2 were:

$$W^{(2)} = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ -1.0 & -1.0 & -1.0 \end{pmatrix}, \quad b^{(2)} = \begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 2.0 \end{pmatrix}. \tag{7}$$

HL2 attempts to disentangle HL1's polysemantic activations. N3, N4, N5 aim to isolate features related to Q1, Q2, Q3 respectively. N6, with its positive bias, activates for Q4 when the sum of activations from HL1 (passed through $W^{(2)}$) is low. The output layer weights ($W^{(\text{out})} \in \mathbb{R}^{4\times 4}$) were an identity matrix $\mathbf{I}_4$, with zero biases. The network architecture and its classification of the input space are shown in Figure 3 (bottom row).

A.1.4 Jaccard Index Analysis

Jaccard indices were computed between the simplex sets of all pairs of constructed DSCs (one-hot vs one-hot, two-hot vs one-hot, random vs one-hot, and a full matrix of all contexts).

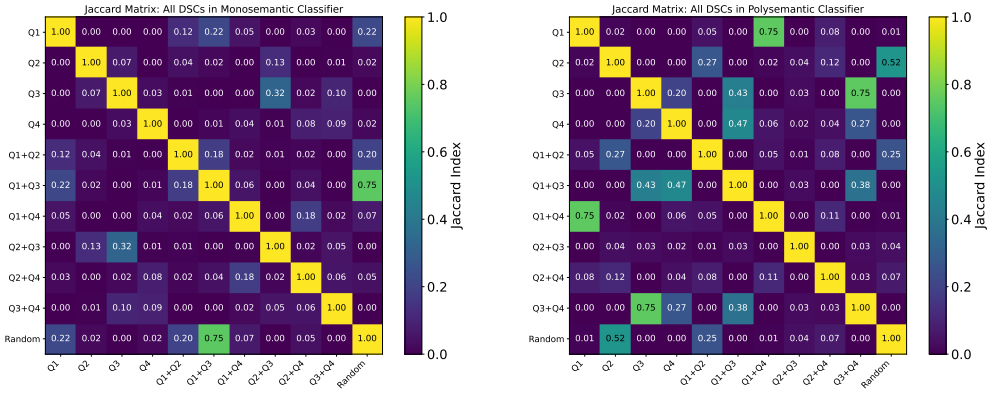

Figure 4: Jaccard similarity matrices for all DSCs built for the monosemantic (left) and polysemantic (right) quadrant classifiers. Values indicate the overlap of simplices between DSCs constructed from different input contexts (single-quadrant, two-quadrant combinations, and random samples).

A.2 MNIST Classifier: Experimental Details

This section provides a detailed description of the experimental setup for the MNIST digit classification task, as summarised in Section 5.2 of the main paper. The experiments were conducted using Python with PyTorch for model definition and training, and UMAP for dimensionality reduction. All computations for Directed Simplicial Complex (DSC) construction were performed primarily using CPU-based parallel processing.

A.2.1 Dataset and Preprocessing

We utilised the standard MNIST dataset of handwritten digits (LeCun et al., 2010), consisting of 60,000 training images and 10,000 testing images, each of size 28×28 pixels. Preprocessing involved converting images to PyTorch tensors, normalising pixel values using the MNIST dataset's mean (0.1307) and standard deviation (0.3081), and subsequently flattening each 28×28 image into a 784-dimensional vector to serve as input to the MLP.

A.2.2 Model Architecture and Training

The multilayer perceptron (MLP) employed had the following architecture:

- Input Layer: 784 neurons (corresponding to the flattened image pixels).
- Hidden Layer 1: 128 neurons.

- Hidden Layer 2: 64 neurons.
- Hidden Layer 3: 32 neurons.
- Hidden Layer 4: 16 neurons.
- Output Layer: 10 neurons (corresponding to the 10 digit classes), followed by a softmax activation for classification.

All hidden layers used the ReLU activation function. The model parameters were initialised using PyTorch's default initialisation for linear layers.

Training was performed on the full MNIST training dataset (60,000 samples). The optimiser used was Adam (Kingma and Ba, 2014) with a learning rate of 0.001, the loss function was standard cross-entropy loss, and the model was trained for 5 epochs. Since the primary goal was to obtain a reasonably performing model for interpretability analysis rather than state-of-the-art classification, and since we planned to use training data with interpretable learned features for our analysis, an explicit test set evaluation was not central to this phase; for our purposes it was sufficient that the model achieved an accuracy of about 98% on the training set.

A.2.3 Selection of Representative Input Samples for DSC Construction

To construct directed simplicial complexes (DSCs), we selected subsets of representative input samples from the MNIST training dataset for various contexts.

Single-Digit (One-Hot) Contexts: For each of the 10 digit classes ($c \in \{0, \ldots, 9\}$), we aimed to identify 20 input images $\mathcal{X}_c = \{x_1, \ldots, x_{20}\}$ in the dataset that strongly and unambiguously activated the network's representation for that class. The selection process was as follows:

1. The trained MLP was applied to all images in the MNIST training set to obtain their corresponding 10-dimensional post-softmax probability vectors.
2. For each true class c, we considered only those input images x whose true label $y(x)$ was c.
3. For these correctly labelled images, we computed the cosine similarity between their network output probability vector $F(x)$ and the one-hot vector h_c for class c. Both $F(x)$ and h_c were L^2-normalised before computing the dot product to ensure a proper cosine similarity measure robust to activation magnitudes.
4. For each class c, the 20 images yielding the highest cosine similarity with h_c were selected to form the set $\mathcal{X}_c$.

This procedure ensures that the DSC for each single-digit class is constructed based on inputs that the model confidently and correctly associates with that class. Figure 5 shows a UMAP (McInnes et al., 2018) projection of the MNIST training data, with these selected representative samples for each class highlighted.

Two-Digit (Two-Hot) Contexts: To investigate feature superposition, DSCs were also constructed for contexts representing pairs of digits. The specific pairs chosen were (2,7) and (5,6), which share features in many handwritings, as well as (1,5) and (1,8) that usually don't share (human perceptible) features. For each pair (a, b), 20 input samples were selected from the entire MNIST training set by:

1. Defining a two-hot vector $h_{a,b}$ with entries for a and b set to 1 and others to 0.
2. Computing the cosine similarity between the L^2-normalised network output probability vector for each training image and the L^2-normalised two-hot vector $h_{a,b}$.
3. Selecting the 20 images yielding the highest cosine similarity to $h_{a,b}$ to form the input set $\mathcal{X}_{a+b}$.

A.2.4 Directed Simplicial Complex Construction

For each set of selected class samples $\mathcal{X}_c$ (one-hot contexts) and $\mathcal{X}_{a+b}$ (two-hot contexts), a directed simplicial complex ($\mathcal{K}_{\mathcal{X}_c}$ or $\mathcal{K}_{\mathcal{X}_{a+b}}$) was constructed using the methodology detailed in section 4. For each set of selected samples, a DSC was constructed with the following parameters:

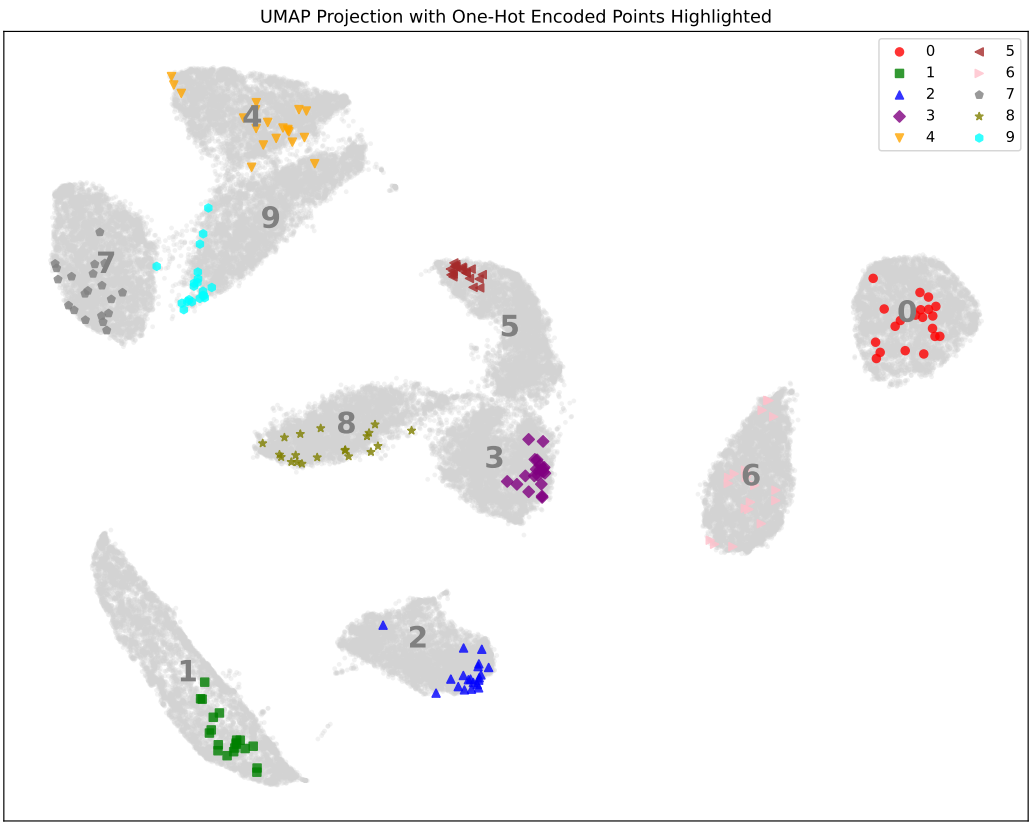

Figure 5: UMAP embedding of the MNIST dataset. Points selected for single-digit DSC construction (one-hot contexts) are highlighted with distinct colours and markers. Class labels (0-9) are annotated near the approximate centroid of each class cluster in the UMAP embedding.

- maximal dimension: 2.
- step size Δt: 0.1 (trajectory generation, see 4.1).
- maximal number of iterations: 10 (trajectory generation, see 4.1).
- Statistical significance α: 0.01 (Wilcoxon signed-rank test, see 4.2).

The calculations were run on the de.NBI Cloud at the Zentrum für Datenverarbeitung of the University of Tübingen using 38 parallel processes in a CPU-based parallelisation. The compute time for constructing each individual DSC (for a context of 20 input samples and max dimension 2 on the MLP with $N = 240$ internal neurons) was approximately 5–10 minutes on a multi-core CPU system. Memory usage per process was modest and did not exceed 1GB. The total compute time using CPU parallelisation for all reported MNIST DSCs (10 one-hot, 4 two-hot) was therefore in the order of 1.5–2.5 hours. Preliminary experiments involving a higher maximal dimension of 3 before final script run incurred significantly longer runtimes of up to 2 hours per DSC and were deemed infeasible to run for all DSCs.

A.2.5 Jaccard Index Analysis

Jaccard indices were computed between the simplex sets of all pairs of constructed DSCs (one-hot vs one-hot, two-hot vs one-hot, random vs one-hot, and a full matrix of all contexts), see Figure 6.

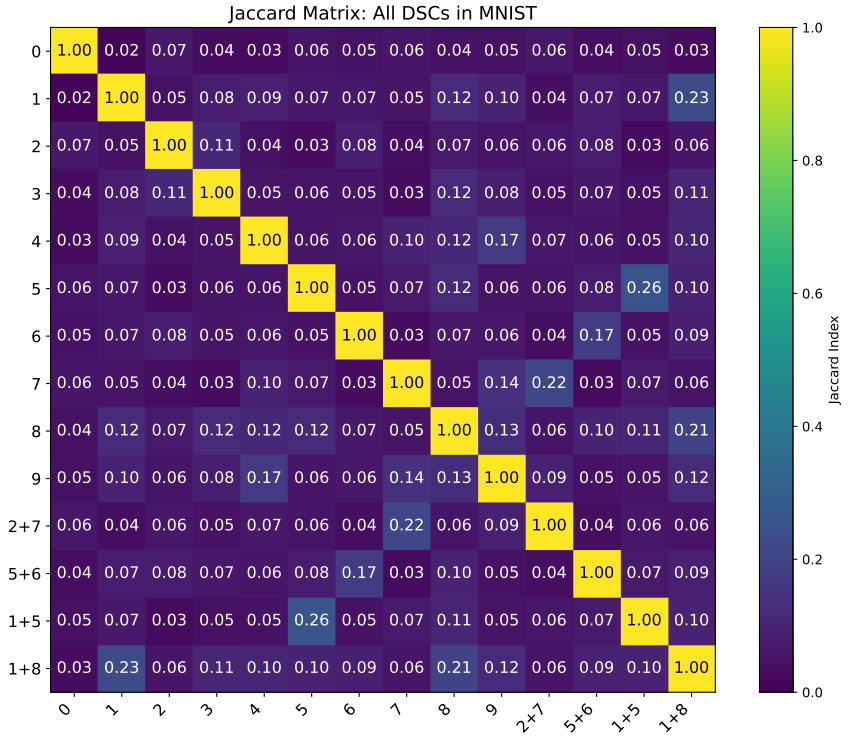

Figure 6: Jaccard similarity matrix for all DSCs built for the MNIST classifier. Values indicate the overlap of simplices between DSCs constructed from different input contexts (single-digit, two-digit combinations, and random samples).

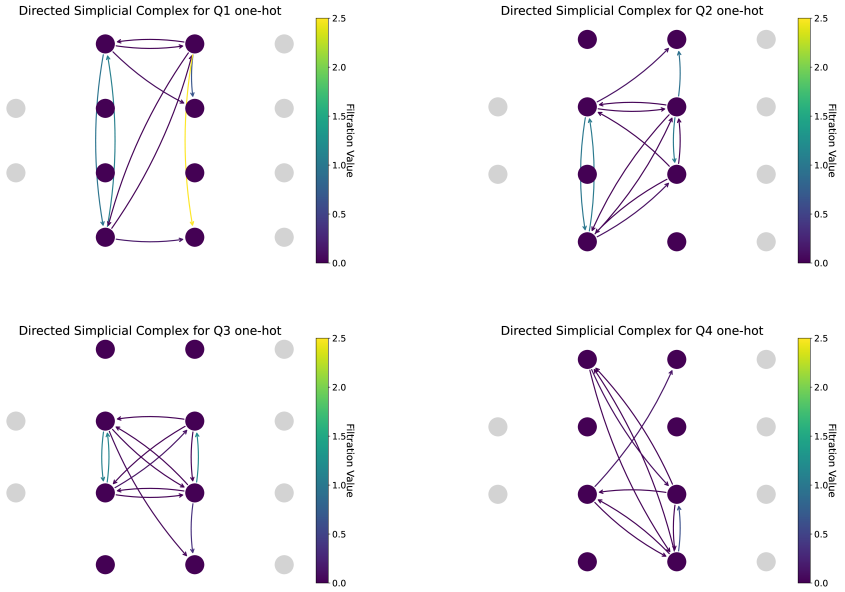

Figure 7: 1-Skeleton of DSCs associated to one-hot encoded input data for the monosemantic quadrant classifiers.

A.3 Supplementary Figures

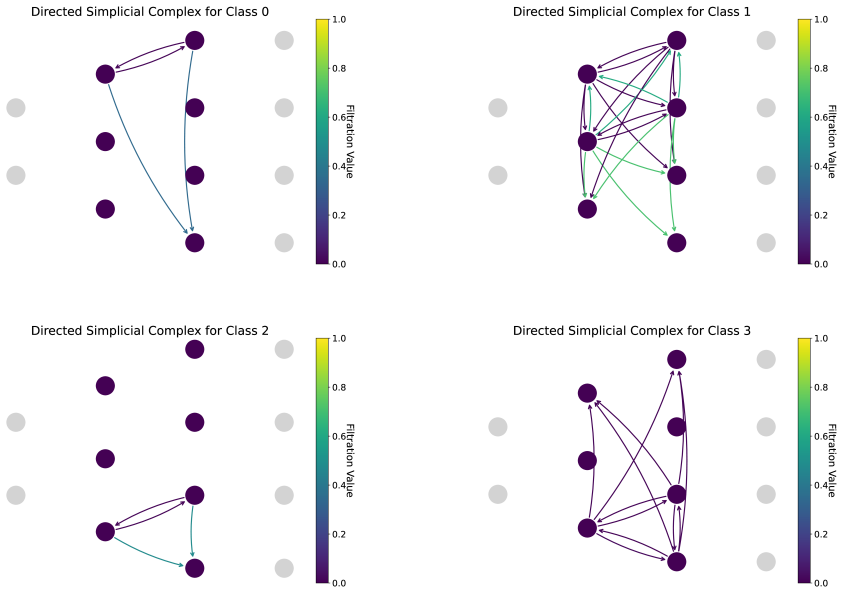

Figure 8: 1-Skeleton of DSCs associated to one-hot encoded input data for the polysemantic quadrant classifiers.