Exploiting Space Folding by Neural Networks

Michal Lewandowski[1], **Raphael Pisoni**[1], **Bernhard Heinzl**[1], **Bernhard A. Moser**[1,2]

[1] Software Competence Center Hagenberg (SCCH)
[2] Johannes Kepler University of Linz (JKU)
michal.lewandowski@scch.at, raphael.pisoni@scch.at, bernhard.heinzl@scch.at, bernhard.moser@scch.at

Abstract

Recent findings suggest that consecutive layers of neural networks with the ReLU activation function *fold* the input space during the learning process. While many works hint at this phenomenon, an approach to quantify the folding was only recently proposed by means of a space folding measure based on the Hamming distance in the discrete activation space. We generalize the space folding measure to a wider class of monotonic activation functions through the introduction of equivalence classes of input data. We then analyze its mathematical and computational properties. Lastly, we link the folding to geometry of adversarial attacks. We underpin our claims with an experimental evaluation.

Introduction

Biological sensory systems, as well as artificial neural networks, transform the input signal into internal representations that efficiently and effectively capture information needed for current and future tasks. For example, in the human eye, the retina removes redundant spatiotemporal structure from incoming light so that it may be efficiently transmitted through the optic nerve. This representation is then transformed in the cortical area by extracting frequently occurring features in support of efficient coding and discrimination of natural images (Barlow 1961; Atick and Redlich 1990; van Hateren 1992; Meister, Lagnado, and Baylor 1995; Balasubramanian and Berry 2002; Puchalla et al. 2005; Doi et al. 2012). Similarly to biological structures, artificial neural networks also transform its input signal, allowing for its use for downstream tasks. This transformation can be analytically studied with tools developed for signal processing Mallat (1989, 2008, 2012).

Motivated by this, we aim to study artificial neural networks (ANNs) through the lens of how they transform the input space. Recent works indicate that ANNs *fold* the input space during the training process, meaning that distant input samples can become close in the activation space (Montúfar et al. 2014; Keup and Helias 2022). Building on these ideas, Lewandowski et al. (2025) proposed a range-based measure in the discrete activation space of ReLU neural networks to quantify *how much* a network folds its input space as it learns. Their analysis focuses on deviations from convexity when mapping a straight-line path in the Euclidean input space to the Hamming activation space. Though simple in nature, analyzing paths in the activation space prove to be insightful as it can be applied to arbitrary paths, and thus statistics derived from these probes might capture the global nature of folds while remaining computationally tractable. Hence, what appears to be a restrictive slice through the input space, instead functions as a stethoscope of sorts that lets us closer examine how (and how much) the network convolutes the input space during learning.

In Lewandowski et al. (2025), the authors rely on the intuition that, if the data manifold learned by a neural network is flat, then the (Euclidean) distance increases monotonically with respect to the initial point when walking along a straight line connecting these points. Contrary, when the data manifold is folded, then the (Hamming) distance computed between respective activation patterns in the Hamming activation space changes non-monotonically – at some point the network "refolds" such that the Hamming distance decreases, which indicates that two previously distant (in the input space) data points have come closer (cf. Fig. 1 right and Fig. 2). Originally developed for ReLU networks, this approach leverages the fact that the ReLU activation function partitions the input space into disjoint linear regions (Makhoul, Schwartz, and El-Jaroudi 1989; Montúfar et al. 2014). In our paper, we firstly show that these regions correspond to equivalence classes defined by the pre-images of either $\{0\}$ or the strictly positive interval $(0, \infty)$. Extending $\{0\}$ to $(-\infty, 0]$ provides a straightforward generalization to a broader class of monotonic activation functions. Secondly, we derive several properties of the space folding measure χ, which do also hold in the general case. Our contributions are as follows.

- **Generalized Folding Measure:** We generalize the space folding measure beyond the ReLU activation function. Our approach relies on the fact that the pre-image of the partition $\{(-\infty, 0), [0, \infty)\}$ divides the domain into two connected sets, $f^{(-1)}((-\infty, 0])$ and $f^{(-1)}((0, \infty))$, for any monotonic and continuous f.
- **Theoretical Analysis:** We state and prove several properties of the folding measure: (*i*) its stability under traversing different activation regions, (*ii*) the sufficient

and necessary (i.e., characterizing) condition for flatness, (*iii*) its sensitivity to the direction of the path, (*iv*) invariance of flatness to direction of path, (*v*) non-additivity.

- **Experimental Evaluation:** We perform small scale experiments to underpin our claims: (*i*) We empirically verify that the sparsity (zero-folding between two inputs) decreases with increasing number of neurons, (*ii*) we show that, during the training process, the folding values steadily increase with the validation accuracy, (*iii*) we establish a link between folding and adversarial geometry.

Related Work

Folding. The idea of folding the (input) space has been investigated, among others, in computational geometry (Demaine, Demaine, and Lubiw 2000). In the context of neural networks, Montúfar et al. (2014) argued that each hidden layer in a ReLU neural network acts as a folding operator, recursively collapsing input-space regions. In Phuong and Lampert (2020), the authors defined the folds by ReLU networks, but left the exploration quite early on. Lewandowski et al. (2025) proposed the first measure to quantify the folding by ReLU neural networks, but stopped short of providing a deeper theoretical analysis. We generalize the measure beyond ReLU, and supplement some theoretical results.

We remark that folding can be seen as a process generating symmetry: When a neural network folds the input space, it effectively identifies different inputs (e.g., an image and its mirror) by mapping them to the same activation pattern - a form of learned invariance. Somewhat implicitly, symmetries have been at the core of some of the most successful deep neural network architectures, e.g., CNNs (Fukushima 1980; LeCun et al. 1989) are equivariant to translation invariance characteristic of image classification tasks, while GNNs (Battaglia et al. 2018) are equivariant to the full group of permutations (see Higgins, Racanière, and Rezende (2022) for a detailed overview). Our work is inspired by symmetries (reflection groups) that arise by space folding and their impact on the generalization capacity of the model.

Path Analysis. Fawzi et al. (2018) used path analysis between input data to explore whether there exists a continuous path that remains in the decision region between any two points of the same label. Hénaff, Goris, and Simoncelli proposed that the visual system transforms inputs to follow straighter temporal trajectories, and developed a methodology for estimating the curvature of an internal trajectory from human perceptual judgments. In Hosseini and Fedorenko (2023), the authors developed a curvature metric that relies on the neural trajectory of words (tokens) in a sentence and found a quantitative behavior of the metric in trained models. Goujon, Etemadi, and Unser (2024) showed that along one-dimensional paths, nonlinearity points scale linearly with depth, width, and activation complexity, while Gamba et al. (2022) proposed a direction-based method to recover all the linear regions along a path. Similarly to these works, we focus on path analysis and its descriptive statistics, however, in addition we leverage the underlying geometry of data. In this way, we capture the transformation of the space by neural networks.

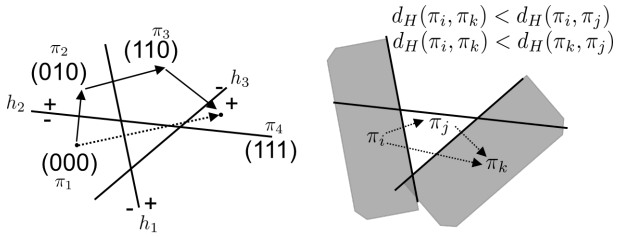

Figure 1: **Left:** Straight-path walk in Euclidean input space vs. Hamming activation space. Dotted: Euclidean shortest path. Arrows: one shortest Hamming path between regions with activation patterns π_1 and π_4 (not unique). Note that adjacent regions can differ by Hamming distance > 1. **Right:** Symmetry in the activation space: gray regions are closer to each other in the Hamming distance than to the region π_j that lies between them.

Preliminaries

We define a *ReLU neural network* $\mathcal{N} : \mathcal{X} \to \mathcal{Y}$ with the total number of N neurons as an alternating composition of the ReLU function $\sigma(x) := \max(x, 0)$ applied element-wise on the input x, and affine functions with weights W_k and biases b_k at layer k. An input $\mathbf{x} \in \mathcal{X}$ propagated through $\mathcal{N}$ generates non-negative activation values on each neuron. A *binarization* is a mapping $\pi : \mathbb{R}^N \to \{0,1\}^N$ applied to a vector $v = (v_1, \ldots, v_N) \in \mathbb{R}^N$, resulting in a binary vector by clipping strictly positive entries of v to 1, and non-positive entries to 0, that is $\pi(v_i) = 1$ if $v_i > 0$, and $\pi(v_i) = 0$ otherwise. In our case, the vector v is the concatenation of all neurons of all hidden layers and its binarization, called an *activation pattern*, represents an element in a binary hypercube $\mathcal{H}^N := \{0,1\}^N$ where the dimensionality is equal to the number N of (hidden) neurons in network $\mathcal{N}$. A *linear region* is an element of a partition covering the input domain where the network behaves as an affine function (Fig. 1, left). The Hamming distance, $d_H(u, v) := |\{u_i \neq v_i \text{ for } i = 1, \ldots, N\}|$, measures the difference between $u, v \in \mathcal{H}^N$, and for binary vectors is equivalent to the L_1 norm between those vectors. Lastly, as we will deal with paths of activation patterns, we denote the operation of joining those paths with the operator $\oplus : \mathcal{H}^{k \cdot N} \times \mathcal{H}^{(n-k+1) \cdot N} \to \mathcal{H}^{n \cdot N}$ such that $\{\pi_1, \ldots, \pi_k\} \oplus \{\pi_k, \ldots, \pi_n\} = \{\pi_1, \ldots, \pi_k, \ldots, \pi_n\}$. The operation $\oplus$ is defined for connected paths, where the last activation pattern of one path matches the first activation pattern of the other.

Space Folding Measure: Construction and Properties

Construction

Consider a straight line connecting two input points $\mathbf{x}_1, \mathbf{x}_2$ in the Euclidean input space. The intermediate points are realized by varying the parameter t in a convex combination $(1 - t)\mathbf{x}_1 + t\mathbf{x}_2$. For a practical implementation, Lewandowski et al. (2025) spaced the parameter t equidistantly on $[0, 1]$, creating n segments. Equal spacing,

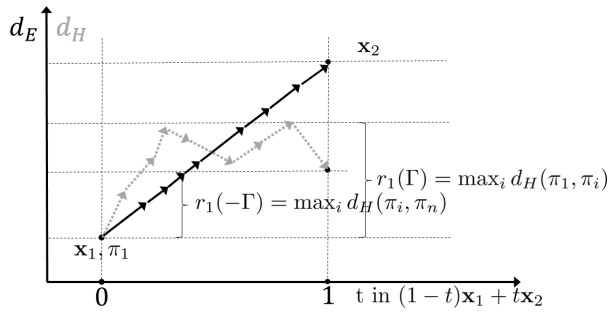

Figure 2: 1D straight walk from $\mathbf{x}_1$ to $\mathbf{x}_2$ in the Euclidean space (black full arrows) and the Hamming activation space (gray dotted arrows). Observe that in the Hamming activation space it might happen that $d_H(\pi_1, \pi_n) < \max_i d_H(\pi_1, \pi_i)$, which indicates space folding. The steps are optimized to visit each equivalence class exactly once (not equidistant).

though easy and fast to implement, frequently results in suboptimal choice of the intermediate points. To obtain a walk through activation patterns, we map the straight line $[\mathbf{x}_1, \mathbf{x}_2]$ through a neural network $\mathcal{N}$ to a *path* $\Gamma := \{\pi_1, \ldots, \pi_n\} \in \mathcal{H}^{n \cdot N}$ in the Hamming activation space, where the intermediate activation patterns belong to a binary hypercube, $\pi_i \in \mathcal{H}^N$ for all $i \in \{1, \ldots, n\}$ (see Fig. 2). We consider a change in the Hamming distance with respect to the initial activation pattern π_1 at each step i, $\Delta_i := d_H(\pi_{i+1}, \pi_1) - d_H(\pi_i, \pi_1)$, and then look at the maximum of the cumulative change $\max_k \sum_{i=1}^{k} \Delta_i$ along the path Γ,

$$r_1(\Gamma) = \max_i \sum_{j=1}^{i} \Delta_j = \max_i d_H(\pi_i, \pi_1). \tag{1}$$

We further keep track of the total distance traveled on the hypercube when following the path,

$$r_2(\Gamma) = \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}). \tag{2}$$

The *space flatness* is defined as the ratio $r_1(\Gamma)/r_2(\Gamma)$; equivalently, this yields the *space folding* measure as

$$\chi(\Gamma) := 1 - \max_i d_H(\pi_i, \pi_1) \Big/ \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}). \tag{3}$$

The folding measure is lower and upper bounded, $\chi \in [0, 1]$ (Lewandowski et al. 2025). Prior to generalizing the folding measure to any activation function, we now formally define folding of the space.

Definition 1 (**Space Folding**). We say that the input space is *folded* between inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ with activation patterns π_1 and π_2, respectively, if $\chi(\Gamma) > 0$ for a path Γ spanned between π_1 and π_2.

Lastly, we motivate the thresholding at 0 used to create activation patterns. While it is a common choice (see e.g., (Montúfar et al. 2014; Hanin and Rolnick 2019)), it raises a frequent question: What is special about 0, and can we threshold at any other $t \neq 0$? We briefly justify its choice through the lens of the space folding measure.

Proposition 1. *For the ReLU activation function, thresholding at $t = 0$ yields the highest information per activation pattern.*

Proof (Informal). Consider thresholds $t_1, t_2 \in \mathbb{R}$, $|t_1| < |t_2|$ in the range of an activation function f. Note that the number of t_1-induced regions is greater or equal the number of t_2-induced regions, hence t_1-induced folding measure is greater or equal t_2-induced folding measure; e.g., for ReLU $t_1 = 0$ is more informative than any $t_2 \neq 0$. □

Beyond ReLU

Before stating several properties of the folding measure χ, we interpret a walk through activation regions in ReLU-based MLP as a walk traversing distinct equivalence classes, and then show how this extends to *any* activation function. This makes our study directly applicable to vast range of activation functions. We start by defining the input equivalence relationship for ReLU neural networks, and will use $\pi(\mathbf{x})$ to denote an activation pattern of an input $\mathbf{x}$.

Definition 2. We say that two inputs $\mathbf{x}_1, \mathbf{x}_2$ are in an equivalence relationship with respect to a neural network $\mathcal{N}$ if their activation patterns $\pi(\mathbf{x}_1), \pi(\mathbf{x}_2)$ are the same, i.e.,

$$\mathbf{x}_1 \sim_{\mathcal{N}} \mathbf{x}_2 \iff d_H\left(\pi(\mathbf{x}_1), \pi(\mathbf{x}_2)\right) = 0.$$

For ReLU neural networks the equivalence class $[\mathbf{x}_1]_{\mathcal{N}} := \{\mathbf{z} \in \mathbb{R}^m \mid \mathbf{z} \sim_{\mathcal{N}} \mathbf{x}_1\}$ corresponds to a linear region which contains point $\mathbf{x}_1$. We now show that the relation in Def. 2 is that of equivalence. Indeed, *reflexivity* holds as $\mathbf{x} \sim \mathbf{x} \Rightarrow \pi(\mathbf{x}) = \pi(\mathbf{x}) \Rightarrow d_H(\pi(\mathbf{x}), \pi(\mathbf{x})) = 0$, and vice-versa, $d_H(\pi(\mathbf{z}), \pi(\mathbf{x})) = 0$ holds for all $\mathbf{z}$ such that $\mathbf{z} \in [\mathbf{x}]_{\mathcal{N}}$, which also contains $\mathbf{x}$. *Symmetry* is straightforward to check, and *transitivity* holds as $\mathbf{x} \sim \mathbf{y}$ and $\mathbf{y} \sim \mathbf{z}$ implies that $d_H(\pi(\mathbf{x}), \pi(\mathbf{y})) = 0$ and $d_H(\pi(\mathbf{y}), \pi(\mathbf{z})) = 0$, thus also $d_H(\pi(\mathbf{x}), \pi(\mathbf{z})) = 0$, and inversely, zero Hamming distances between $\pi(\mathbf{x})$ and $\pi(\mathbf{y})$ as well as $\pi(\mathbf{y})$ and $\pi(\mathbf{z})$ imply that $\mathbf{z} \in [\mathbf{x}]_{\mathcal{N}}$.

Definition 2 paves a way to extending the folding analysis to a richer class of activation functions. However, we lose the geometrical interpretation of equivalence classes as "linear regions". Henceforth for the computation of χ, we consider a walk through input equivalence classes, not linear regions. In order to obtain binary activation vectors, we clip the values on the hidden layers (after applying the activation function) in a similar way as with the ReLU function, i.e., for a vector of activation values $\mathbf{a} \in \mathbb{R}^n$ we create the corresponding activation pattern by considering strictly positive vs. non-positive activation values, and denoting them with 1 and 0, respectively. We note that for non-monotonic activation functions the equivalence classes defined above may be topologically disconnected but the construction applies as is.

Example 1. *For a monotonic activation like sigmoid, we similarly binarize activations at 0 to obtain patterns; two inputs yielding the same pattern lie in an equivalence class.*

Unlike ReLU's linear regions, these classes aren't polyhedral but still partition the input space into two connected regions per neuron (pre-image of negative vs positive outputs). The folding measure χ is computed by traversing these equivalence classes.

Properties

In this section, we look at several properties of the folding measure. They will serve as the base for further analysis.

Lemma 1. *The folding measure χ has the following properties:*

1. ***(Stability.)** Multiple steps in the same activation region do not influence χ.*
2. ***(Flatness Condition.)** $\chi(\Gamma) = 0$ implies that $d_H(\pi_1, \pi_i)$ is increasing for $i = 1, \ldots, n$ along Γ.*
3. ***(Asymmetry.)** The folding measure is sensitive to the direction of traversal, i.e., $\chi(\Gamma) \neq \chi(-\Gamma)$.*
4. ***(Flatness Invariance.)** $\chi(\Gamma) = 0$ if and only if $\chi(-\Gamma) = 0$ for a path $\Gamma = \{\pi_1, \ldots, \pi_n\}$.*
5. ***(Non-additivity.)** The folding measure χ in general is neither sub-additive nor super-additive over concatenated path, i.e., it neither holds that $\chi(\Gamma_1 \oplus \Gamma_2) > \chi(\Gamma_1) + \chi(\Gamma_2)$ nor $\chi(\Gamma_1 \oplus \Gamma_2) < \chi(\Gamma_1) + \chi(\Gamma_2)$, where the operator $\oplus$ is as defined in the preliminaries.*

Proof. We now prove properties listed in Lemma 1.

1. (**Stability.**) The proof is straightforward as staying in the same activation region does not change either Hamming distance.
2. (**Flatness Condition.**) Assume, without loss of generality, that each π_i is distinct from π_{i-1}. By triangle inequality, $d_H(\pi_1, \pi_n) \leq d_H(\pi_1, \pi_i) + d_H(\pi_i, \pi_n)$. If $d_H(\pi_1, \pi_i) \leq d_H(\pi_1, \pi_{i-1})$ were to hold for any i, then by applying triangle inequality again, we would find that $d_H(\pi_1, \pi_n) < \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1})$. However, $\chi(\Gamma) = 0$ implies that $d_H(\pi_1, \pi_n) = \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1})$, leading to a contradiction.
3. (**Asymmetry.**) By counterexample: consider $\Gamma = \{\pi_1, \pi_2, \pi_3\}$, where $\pi_1 = (000), \pi_2 = (111), \pi_3 = (001)$, and its reverse $-\Gamma = \{\pi_3, \pi_2, \pi_1\}$. Then, $r_2(\Gamma) = r_2(-\Gamma)$ but $r_1(\Gamma) = 3$ and $r_1(-\Gamma) = 2$, thus $\chi(\Gamma) \neq \chi(-\Gamma)$.
4. (**Flatness Invariance.**) Observe that it is sufficient to prove it only in way direction as we can re-index the path Γ to obtain its reverse. If $\chi(\Gamma) = 0$, then $d_H(\pi_1, \pi_i)$ never decreases with increasing i. Along the reversed path $-\Gamma$, this translates to $d_H(\pi_n, \pi_{n-i+1})$ never decreasing, so $\chi(-\Gamma) = 0$. Conversely, if $\chi(-\Gamma) = 0$, then similarly $\chi(\Gamma) = 0$.
5. (**Non-additivity.**) For a counter example of the sub-additivity consider paths $\Gamma_1 = \{\pi_1, \pi_2\}$ and $\Gamma_2 = \{\pi_2, \pi_3, \pi_4\}$ with the activation regions defined as

$$\pi_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}, \pi_2 = \begin{pmatrix}0\\0\\1\end{pmatrix}, \pi_3 = \begin{pmatrix}1\\1\\1\end{pmatrix}, \pi_4 = \begin{pmatrix}1\\0\\1\end{pmatrix}. \quad (4)$$

In this case, $\chi(\Gamma_1 \oplus \Gamma_2) = \frac{1}{4}$ and $\chi(\Gamma_1) + \chi(\Gamma_2) = 0 + \frac{1}{3} = \frac{1}{3}$, thus $\chi(\Gamma_1) + \chi(\Gamma_2) \geq \chi(\Gamma_1 \oplus \Gamma_2)$ (for connected paths Γ_1 and Γ_2). To see that we can also construct a counter example for super-additivity, consider paths as previously with the activation patterns defined as

$$\pi_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}, \pi_2 = \begin{pmatrix}1\\1\\1\end{pmatrix}, \pi_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}, \pi_4 = \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad (5)$$

Then, $\chi(\Gamma_1 \oplus \Gamma_2) = \frac{4}{7}$ while $\chi(\Gamma_1) + \chi(\Gamma_2) = 0 + \frac{1}{2} = \frac{1}{2}$, thus $\chi(\Gamma_1) + \chi(\Gamma_2) \leq \chi(\Gamma_1 \oplus \Gamma_2)$.

□

Interpretation

In the remainder of this section, we will discuss and interpret properties listed in Lemma 1.

Stability. The stability property justifies introducing an equivalence relationship between inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ with no folding between, which we formalize as follows.

Definition 3. We say that input points $\mathbf{x}_1$ and $\mathbf{x}_2$ are equivalent under χ, i.e., $\mathbf{x}_1 \sim_\chi \mathbf{x}_2$, if

$$\chi(\Gamma(\mathbf{x}_1, \mathbf{x}_2)) = 0, \quad (6)$$

where $\Gamma(\mathbf{x}_1, \mathbf{x}_2)$ is a path of activation patterns spanned between $\mathbf{x}_1$ and $\mathbf{x}_2$.

We use Def. 3 to introduce a space folding-based metric in the Hamming activation space in the subsequent section.

Flatness Condition. The Flatness condition stated in Lemma 1 implies that folding occurs if r_1 (Eq. (1)) decreases at least once along the path. Flatness means that a straight line mapped through a network is itself a "straight line" in the Hamming space.

Non-additivity. While neither super- nor sub-additivity holds for every path Γ, in our experiments we have only observed sub-additivity of the folding measure. The counterexample for super-additivity (Eq. (5)) seems to be a rare occurrence in trained networks, though it can be observed in specially constructed examples (see CantorNet by (Lewandowski, Eghbalzadeh, and Moser 2024)). The general lack of super- or sub-additivity, but empirical sub-additivity motivates us to introduce the interaction coefficient $\mathcal{I}$ (deviation from additivity) for two paths Γ_1 and Γ_2 as $\mathcal{I} : \mathcal{H}^{n_1 \cdot N} \times \mathcal{H}^{n_2 \cdot N} \to [0, 1]$, where

$$\mathcal{I}(\Gamma_1, \Gamma_2) := |\chi(\Gamma_1 \oplus \Gamma_2) - \chi(\Gamma_1) - \chi(\Gamma_2)|. \quad (7)$$

$\mathcal{I}(\Gamma_1, \Gamma_2) = 0$ for paths Γ_1, Γ_2 if and only if additivity holds for those two subpaths; $\mathcal{I}(\Gamma_1, \Gamma_2) > 0$ means some fold "cancel out" or amplify when combining paths. Later, we study the applicability of the interaction coefficient $\mathcal{I}$ to study the geometry of the decision boundary. Our intuition is as follows: Folding of the input space is related to *deviations* from convexity (see Moser et al. (2022); Lewandowski et al. (2025)) and the increased number of these deviations indicates a decision boundary that is more ragged, and thus more sensitive to small perturbations of the original data. We

hypothesize that $\mathcal{I}$ will exhibit sensitivity when computed between an original sample and its adversarial perturbation, and present preliminary experimental results.

While χ proves to be asymmetric in nature, in our experiments we have observed that, as the number of neurons increases, $\chi(\Gamma)$ and $\chi(-\Gamma)$ seem to converge. We hypothesize that with many neurons, any specific order of folds can be realized in reverse by alternate paths due to the abundance of intermediate regions. We express it as Conjecture 1.

Conjecture 1. *As neural network's total number of neurons N increases, the folding measure becomes asymptotically symmetric, i.e.,* $\lim_{N\to\infty} |\chi(\Gamma) - \chi(-\Gamma)| \to 0$.

We now introduce the notion of *sparsity* of folding values as the ratio of paths that exhibit no folding effects, to the paths along which the folding is positive.

Definition 4 (Sparsity). Let $|A|$ denote the cardinality of a set A. The sparsity $\mathcal{S}_{\mathcal{N}}$ of χ under $\mathcal{N}$ is the ratio

$$\mathcal{S}_{\mathcal{N}} := \frac{|\{\Gamma : \chi(\Gamma) = 0\}|}{|\{\Gamma\}|} \in [0, 1]. \tag{8}$$

We empirically investigate the sparsity as a function of total number of neurons in a later section (Experiments).

Space Folding-based Pseudo-metric

In this section, we introduce a pseudo-metric whose construction is inspired by χ (Eq. (3)). The path Γ is spanned between its edge points, i.e., $\Gamma = \Gamma(\mathbf{x}_1, \mathbf{x}_2)$. Without loss of generality, we assume that every intermediate step visits exactly one activation pattern.

Proposition 2 (Space Folding-based Pseudo-metric)**.** *Let d_χ be a symmetrized space folding measure, i.e.,*

$$\begin{aligned} d_\chi(\mathbf{x}_1, \mathbf{x}_2) &:= \frac{\chi(\Gamma(\mathbf{x}_1, \mathbf{x}_2)) + \chi(\Gamma(\mathbf{x}_2, \mathbf{x}_1))}{2} \\ &= \frac{\max_j d_H(\pi_1, \pi_i) + \max_i d_H(\pi_i, \pi_n)}{2\sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1})}, \end{aligned} \tag{9}$$

where $\Gamma(\mathbf{x}_1, \mathbf{x}_2)$ denotes a path in the activation space between $\mathbf{x}_1$ and $\mathbf{x}_2$. Then, d_χ is a pseudo-metric.

Proof. Positivity follows from bounds: $d_\chi \in [0, 1]$, as proved for the measure χ by (Lewandowski et al. 2025), symmetry $d_\chi(\mathbf{x}_1, \mathbf{x}_2) = d_\chi(\mathbf{x}_2, \mathbf{x}_1)$ follows from the definition (Eq. (9)). To show the triangle inequality, we need to show that $d_\chi(\mathbf{x}_1, \mathbf{x}_2) + d_\chi(\mathbf{x}_2, \mathbf{x}_3) \geq d_\chi(\mathbf{x}_1, \mathbf{x}_3)$ for every $\Gamma(\mathbf{x}_1, \mathbf{x}_2) = \{\pi_1, \ldots, \pi_k\}$ and $\Gamma(\mathbf{x}_2, \mathbf{x}_3) = \{\pi_k, \ldots, \pi_n\}$. It is straightforward to check, and requires using $\max_{j\in\{1,\ldots,k\}} d_H(\pi_1, \pi_j) + \max_{j\in\{k,\ldots,n\}} d_H(\pi_k, \pi_j) \geq \max_{j\in\{1,\ldots,n\}} d_H(\pi_1, \pi_j)$. □

d_χ is a *pseudo*-metric as $\mathbf{x}_1 = \mathbf{x}_2 \Rightarrow d_\chi(\mathbf{x}_1, \mathbf{x}_2) = 0$ but the reverse does not hold, i.e., $d_\chi(\mathbf{x}_1, \mathbf{x}_2) = 0 \not\Rightarrow \mathbf{x}_1 = \mathbf{x}_2$. However, if we restrict $\mathbf{x}_1$ and $\mathbf{x}_2$ such that $\mathbf{x}_1 \not\sim_\chi \mathbf{x}_2$ (Def. 3), then d_χ becomes a metric. Lastly, note that $1 - d_\chi \in [0, 1]$ can be used to measure similarity between input points, and thus may be used in downstream tasks such as clustering or retrieval tasks (see the discussion in Conclusion section).

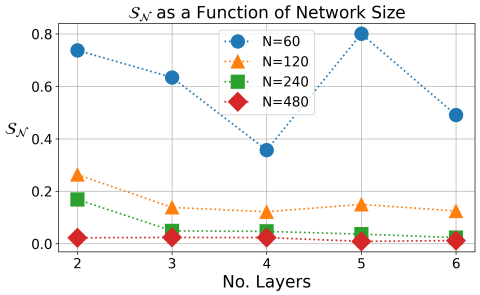

Figure 3: Sparsity $\mathcal{S}_{\mathcal{N}}$ (Def. 4) as a function of the number of layers with varying total number of neurons N (colors).

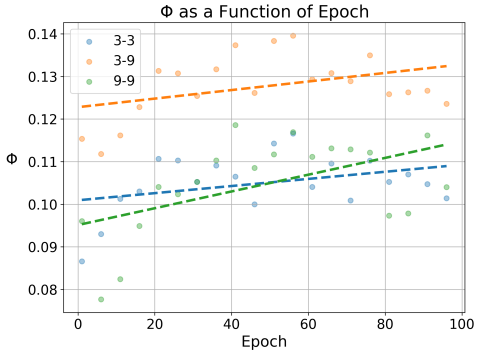

Figure 4: Global folding $\Phi_{\mathcal{N}}$ on CIFAR100 test set over 100 epochs; dashed lines indicate trends. At epoch 0 (random init), folding values are the lowest and steadily increase with training, indicating empirical link between folding and the validation accuracy.

Global Space Folding Measure

In this section, we adapt a global measure of folding as a median of space folding values along paths that exhibit some folding, i.e.,

$$\Phi_{\mathcal{N}} := \underset{\{\Gamma:\chi(\Gamma)>0\}}{\text{median}}\, \chi(\Gamma) \tag{10}$$

Alternatively, we could compute the global folding measure over a simplex consisting of k input points in the input space (not necessarily all), $\sum_{i=1}^{k} \alpha_i \mathbf{x}_i,\ \sum_i \alpha_i = 1,\ \alpha_i \geq 0$, which is then mapped under a neural network $\mathcal{N}$.

Experiments

Sparsity

We evaluate the sparsity according to Def. 4 on fully-connected ReLU networks of varying size trained on MNIST. We find that $\mathcal{S}_{\mathcal{N}} \in (0.4, 0.8)$ for very small networks ($N = 60$ neurons total), and it decreases to 0 with increasing number of total neurons N, i.e., $\lim_{N\to\infty} \mathcal{S}_{\mathcal{N}} = 0$ (Fig. 3). We averaged the results over 4 different architectures for each number of total neurons (using dropout or batch-normalization) and found that the variance is small.

Algorithm 1: $\mathcal{I}$ and Adversarial Attacks

Input: Dataset $\{(\mathbf{x}_i, y_i)\}_{i \in I}$
Output: Mean non-zero $\mathcal{I}$

1: **Step 1:** Adversarially perturb the input $\mathbf{x}$ obtaining $\mathbf{x}^*$.
2: **Step 2:** Assert that $\mathcal{N}(\mathbf{x}) \neq \mathcal{N}(\mathbf{x}^*)$.
3: **Step 3:** Compute $\mathcal{I}$ on a path spanned between $[\mathbf{x}, \mathbf{x}^*]$
4: (a) check the linear combination of $\mathbf{x}$ and $\mathbf{x}^*$ for varying connecting index k.
5: (b) store k s.t. $\mathcal{I}$ increases step-wise
6: **return** $\frac{1}{n-k+1} \sum_{i>k} \mathcal{I}_i$

Evolution of Folding During the Training Process

We investigate the behaviour of $\Phi_{\mathcal{N}}$ (see Eq. (10)) during the training progress (i.e., we study $\Phi_{\mathcal{N}}(\text{epoch})$). We train a ReLU-based MLP (2×256) over 100 epochs to a validation accuracy of around 50% on CIFAR100 - we stopped at 50% as exceeding it requires architectural changes beyond plain MLPs (Neyshabur 2020); the absolute performance is not crucial here. We store the weights and biases of the model after every epoch of training, and compute χ for pre-selected pairs of test points within each class and for pairs from different classes. We found that, with the increasing validation accuracy of the model, the global folding value $\Phi_{\mathcal{N}}$ is monotonically increasing.

We observe that, for networks with lower validation accuracy values, the intra-class folding values are similar (or lower) than inter-class folding values, while the opposite holds for well-trained networks. This might be a consequence of the higher concentration of linear regions close to data for well-trained neural networks, as observed by Zhang and Wu (2020). See Fig. 4 for more details.

Remark 1. In our experiments on simple classification tasks (MNIST, CIFAR100), networks with near-zero training error exhibit higher median folding Φ_N within each class than between classes.

Conjecture 2. *Remark 1 suggests that perfect classification of training data requires folding each class's manifold into a more compact shape (higher intra-class folding on average).*

The intuition behind Conjecture 2 is as follows. A well-trained network learns to transform intra-class variations of an object such that their representations are brought closer together in the activation space. This effectively *folds* the input manifold for each class into a simpler, more compact structure – an idea reminiscent of contrastive learning. We have observed that the folding values does not seem to be influenced by the total number of neurons, which we formalize as Conjecture 3.

Conjecture 3. *For sufficiently large networks, Φ_N approaches a constant that depends only on depth provided low generalization error.*

Impact of Regularization Techniques

For the completeness of our study, in Fig. 5 we have investigated the impact of popular regularization techniques,

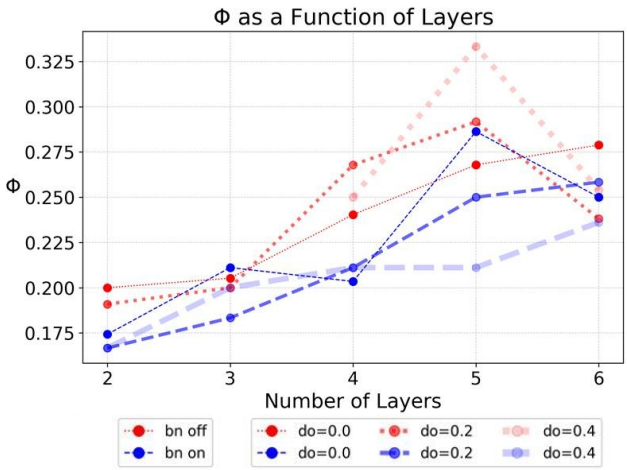

Figure 5: Impact of batch norm and dropout with varying rate on Φ. Though there is a small difference, the behaviour remains similar. The thinnest red line serves as a baseline.

dropout (Srivastava et al. 2014) and batch normalization (Ioffe and Szegedy 2015) on small neural networks with a total number of hidden neurons trained on MNIST. All the networks were trained to a high validation accuracy; the values of $\Phi_{\mathcal{N}}$ increase with the depth of the network as reported by Lewandowski et al. (2025). We found that, although networks without batch norm feature higher values of space folding, the differences are small. We thus conclude that the considered regularization techniques do not influence the space folding values in a significant manner.

Interaction Coefficient

In the next step, we investigate the values of the interaction coefficient $\mathcal{I}$ (Eq. (7)) on the unperturbed and adversarially perturbed images of digits from the MNIST dataset (see Alg. 1). Our intuition here is as follows. Sensitivity of ReLU-based MLPs has been linked to the geometry of their decision boundary (see e.g., (Wong and Kolter 2018)); intuitively, the more *ragged* the decision boundary, the more susceptible are the data samples to being mis-classified by the model upon a small perturbation. Such a raggedness of a decision boundary translates to its non-convexity, and the space folding measure was designed to quantify the *deviations* from convexity (see Lewandowski et al. (2025)), thus studying its non-additivity appears to be a sensible choice to study adversarial geometry.

We proceed as follows. We vary the connecting activation pattern π_k. We observe that $\mathcal{I}$ displays a step-wise behaviour: with $t \to 1^-$, $(1-t)\mathbf{x}_1 + t\mathbf{x}_2 \to \mathbf{x}_2^-$, and the value of $\mathcal{I}$ tends to jump and flatten. The height of the jump depends on the points between which it is computed, see Fig. 6. Across both adversarial attacks considered, the interaction coefficient $\mathcal{I}$ is consistently higher for original (unperturbed) samples (see Fig. 6). Under the projected gradient descent (PGD) attack (Madry et al. 2018), the lowest values of $\mathcal{I}$ occur when computed between two adversarially perturbed samples. In contrast, for the fast gradient sign method

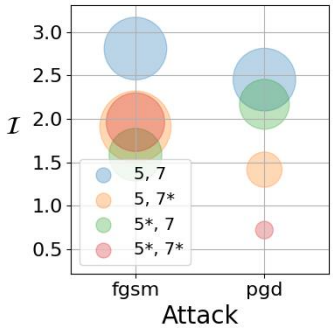

Figure 6: Values of the interaction coefficient (Eq. (7)) were computed using images of digits 5 and 7 from the MNIST test set. The symbol * denotes an adversarially perturbed digit. For visualization, the y-axis has been scaled by a factor of 10^2. Each circle is centered at the mean of the points it contains, with its size reflecting the number of classified observations.

(FGSM) (Goodfellow, Shlens, and Szegedy 2015), the interaction coefficients are similar to those obtained when only one of the two samples is perturbed.

Conclusions and Future Work

In our work, we have generalized the concept of space folding to any monotonic activation function, and we have empirically investigated (*i*) the sparsity of paths as a function of total number of neurons, (*ii*) the evolution of global folding measure Φ during the training process, (*iii*) the behaviour of the interaction coefficient $\mathcal{I}$ (Eq. (7)) on adversarially perturbed samples of MNIST, (*iv*) the impact of dropout (Srivastava et al. 2014) and batch normalization (Ioffe and Szegedy 2015) on the folding values. Our study deepened the mathematical understanding of the space folding measure, lays the groundwork for further experimental work, and highlights key theoretical properties.

We note some parallels between contrastive learning and learning with increased space folding. In short, contrastive learning, accomplished through a contrastive loss, ensures (*i*) alignment (closeness) of features between positive pairs, and (*ii*) uniformity of the induced distribution of the (normalized) features on a manifold (Wang and Isola 2020). Note that by encouraging greater folding during training we bring data points closer in alignment space, and by penalizing folding between samples from different classes we push them farther apart, which parallels contrastive learning.

While our empirical evaluation is limited to smaller-scale datasets (MNIST, CIFAR-100) and simple MLP models due to the heavy computation of χ, the observed trends are consistent and illustrative. Scaling up the folding measure to larger architectures is left for future work.

Acknowledgments

This research was carried out under the Austrian COMET program, which is funded by the Austrian ministries BMIMI, BMWET, and the province of Upper Austria. We thank the anonymous reviewers for their constructive comments, which improved the quality of this work. Additionally, we thank Hamid Eghbalzadeh for many fruitful and inspiring discussions.

References

Atick, J. J.; and Redlich, A. N. 1990. Towards a theory of early visual processing. *Neural Comput.*

Balasubramanian, V.; and Berry, M. J. 2002. A test of metabolically efficient coding in the retina. *Network*.

Barlow, H. B. 1961. Possible principles underlying the transformation of sensory messages. *Sensory Communication*.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Demaine, E. D.; Demaine, M. L.; and Lubiw, A. 2000. Folding and Cutting Paper. *Discrete and Computational Geometry*.

Doi, E.; et al. 2012. Efficient coding of spatial information in the primate retina. *J. Neurosci.*

Fawzi, A.; Moosavi-Dezfooli, S.-M.; Frossard, P.; and Soatto, S. 2018. Empirical Study of the Topology and Geometry of Deep Networks. *CVPR*.

Fukushima, K. 1980. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*.

Gamba, M.; Chmielewski-Anders, A.; Sullivan, J.; Azizpour, H.; and Bjorkman, M. 2022. Are All Linear Regions Created Equal? *AISTATS*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. *ICLR*.

Goujon, A.; Etemadi, A.; and Unser, M. 2024. On the number of regions of piecewise linear neural networks. *Journal of Computational and Applied Mathematics*.

Hanin, B.; and Rolnick, D. 2019. Deep ReLU Networks Have Surprisingly Few Activation Patterns. *NeurIPS*.

Hénaff, O. J.; Goris, R. L. T.; and Simoncelli, E. P. 2019. Perceptual straightening of natural videos. *Nature Neuroscience*.

Higgins, I.; Racanière, S.; and Rezende, D. 2022. Symmetry-Based Representations for Artificial and Biological General Intelligence. *Frontiers in Computational Neuroscience*.

Hosseini, E.; and Fedorenko, E. 2023. Large language models implicitly learn to straighten neural sentence trajectories to construct a predictive representation of natural language. *NeurIPS*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*.

Keup, C.; and Helias, M. 2022. Origami in N dimensions: How feed-forward networks manufacture linear separability. *arXiv preprint arXiv:2203.11355*.

LeCun, Y.; Boser, B. E.; Denker, J. S.; et al. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*.

Lewandowski, M.; Eghbalzadeh, H.; Heinzl, B.; Pisoni, R.; and A.Moser, B. 2025. On Space Folds of ReLU Neural Networks. *TMLR*.

Lewandowski, M.; Eghbalzadeh, H.; and Moser, B. 2024. CantorNet: A Sandbox for Testing Topological and Geometrical Measures. *NeurIPS W*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR*.

Makhoul, J.; Schwartz, R.; and El-Jaroudi, A. 1989. Classification capabilities of two-layer neural nets. *International Conference on Acoustics, Speech, and Signal Processing*.

Mallat, S. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Mallat, S. 2008. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3rd edition.

Mallat, S. 2012. Group invariant scattering. *Communications on Pure and Applied Mathematics*.

Meister, M.; Lagnado, L.; and Baylor, D. A. 1995. Concerted signaling by retinal ganglion cells. *Science*.

Montúfar, G. F.; Pascanu, R.; Cho, K.; and Bengio, Y. 2014. On the Number of Linear Regions of Deep Neural Networks. *NeurIPS*.

Moser, B. A.; Lewandowski, M.; Kargaran, S.; Zellinger, W.; Biggio, B.; and Koutschan, C. 2022. Tessellation-Filtering ReLU Neural Networks. *IJCAI*.

Neyshabur, B. 2020. Towards learning convolutions from scratch. *NeurIPS*.

Phuong, M.; and Lampert, C. H. 2020. Functional vs. Parametric Equivalence of ReLU Networks. *ICLR*.

Puchalla, J. L.; Schneidman, E.; Harris, R. A.; and Berry, M. J. 2005. Redundancy in the population code of the retina. *Neuron*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*.

van Hateren, J. H. 1992. A theory of maximizing sensory information. *Biol. Cybern.*

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *ICML*.

Wong, E.; and Kolter, Z. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. *ICML*.

Zhang, X.; and Wu, D. 2020. Empirical Studies on the Properties of Linear Regions in Deep Neural Networks. *ICLR*.