On Space Folding by Neural Networks

12.05.2026

Michal Lewandowski

Software Competence Center Hagenberg (SCCH)

Presentation Outline

scch { }

- **Introduction**: Motivation & Background
- **Approach**: The "*Space Folding*" Measure χ
  - Construction
  - Properties
  - Connection to Motzkin-Straus graph Lagrangian
- **Interpretation**: Meaning for the ML Community
- Ongoing / Future Work

This research is *not* about LLMs

But it can be used for mechanistic interpretability

Driving Questions:
How neural networks transform the data space during the training process?
Which geometrical structures are preserved?

Co-authors

Bernhard

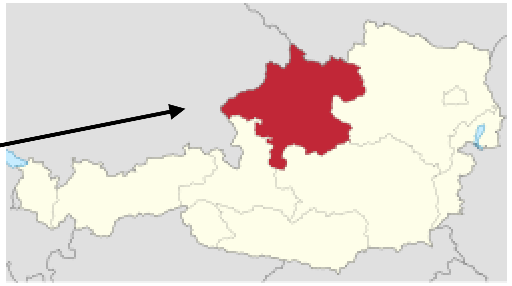
Hamid

Raphael

Roman

Bernhard

Bernhard



SCCH: A Non-University Research Center in Upper Austria

S3AI: A Research Project Focused On Geometry Of Neural Networks (2020-2024)

scch { }
software competence
center hagenberg

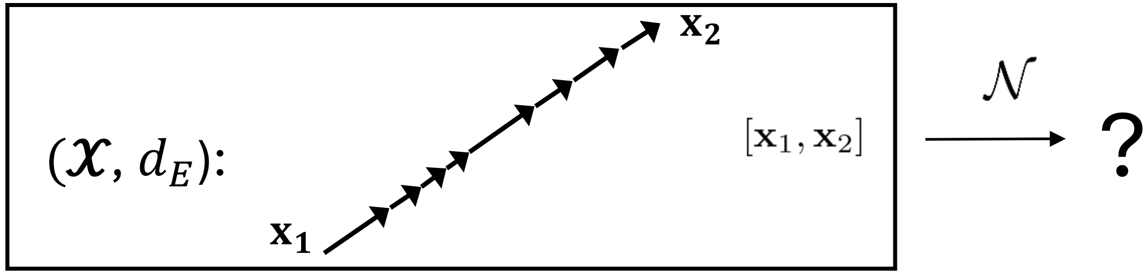
s3AI

Introduction

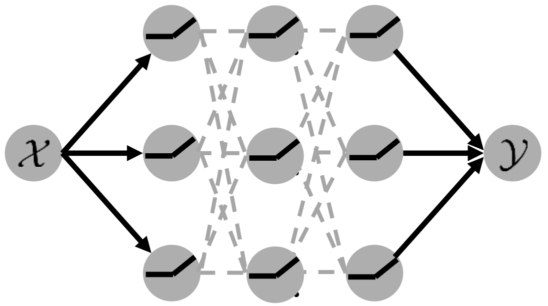
scch { }

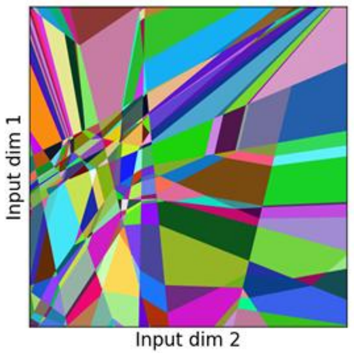
Q: How neural nets transform the data space during training?

How:
by measuring *deviations* from convexity in the activation spaces



- We illustrate our approach on ReLU MLPs
  - ✓ Piece-wise linear functions - inherit a geometrical interpretation
  - ✓ Universal approximation and exponential expressiveness



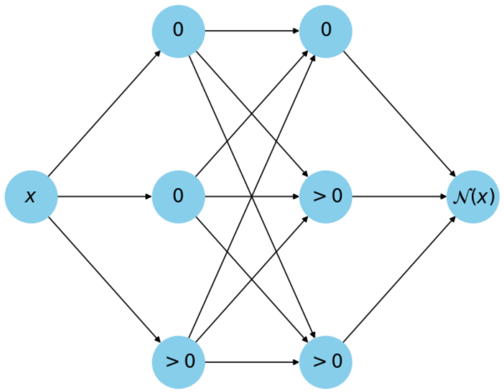


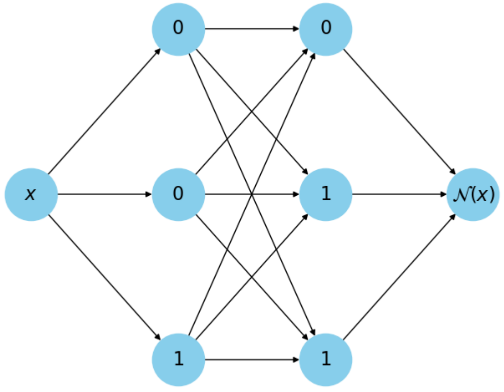
(Reproduced from [1])

[1] H. Boris, and D. Rolnick. (2019). *Deep relu networks have surprisingly few activation patterns.* NeurIPS.

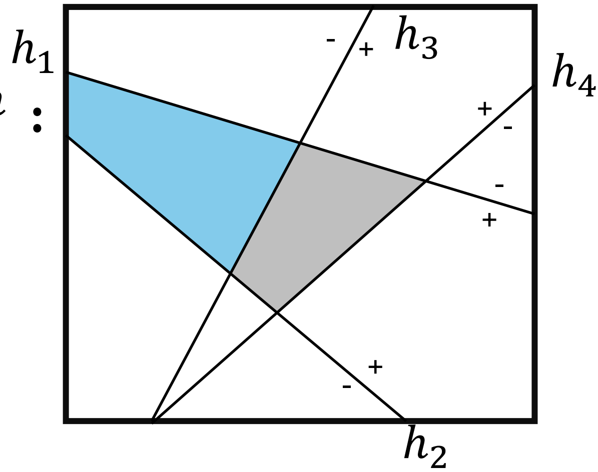
Hamming Activation Space

- A ReLU NN (denoted $\mathcal{N}$) is an alternating composition of the ReLU ($\sigma(x) \coloneqq \max(x, 0)$) and affine functions $g_i(\mathbf{x}) \coloneqq W_i\mathbf{x} + b_i$
- Fix $(W_i, b_i)_{i=1}^{L}$; push an input $\mathbf{x}$ through $\mathcal{N}$; at every layer we obtain a vector of activation values





- Activation pattern: $\pi = (001011)$
- For any $\mathbf{x}_i \in \mathcal{X}$, there is an associated π_i
- Each (realizable) π_i is a solution of affine inequalities, e.g. $\{\mathbf{x}: h_1(\mathbf{x}) \geq 0 \;\&\; \ldots \;\&\; h_4(\mathbf{x}) \geq 0\}$
- In geometry: *a polytope* [1], in ML: *a linear region* [2]



- Hamming distance d_H quantifies difference between $\pi_i, \pi_j \in \{0,1\}^m$:
  $d_H(\pi_i, \pi_j) \coloneqq |\{k: \pi_{i,k} \neq \pi_{j,k}\}|$
- $(\{0,1\}^m, d_H)$ is a metric space *(The Hamming Activation Space)*

[1] Ziegler, G. M. (2000). *Lectures on 0/1-polytopes*. Polytopes—combinatorics and computation.
[2] Montufar et al. (2014). *On the Number of Linear Regions of Deep Neural Networks*. NeurIPS.

Intervals in Euclidean and Hamming Activation Spaces

$[P, Q] \coloneqq \{(1-t)P + tQ : t \in [0,1]\};$ $\longrightarrow$ Interval between P and Q in Euclidean space

$I(\pi_1, \pi_2) \coloneqq \{\pi_a : d_H(\pi_1, \pi_a) + d_H(\pi_a, \pi_2) = d_H(\pi_1, \pi_2)\}$

Interval between π_1 and π_2: set of all $\pi's$ that lie on <u>some shortest Hamming path</u> from π_1 to π_2

Alternatively:

$I(\pi_1, \pi_2) \coloneqq \{\pi_a : \pi_{a,k} = \pi_{1,k} = \pi_{2,k}$ whenever $\pi_{1,k} = \pi_{2,k}\}$

Coordinates where π_1 and π_2 agree are fixed;
Coordinates where they differ are free

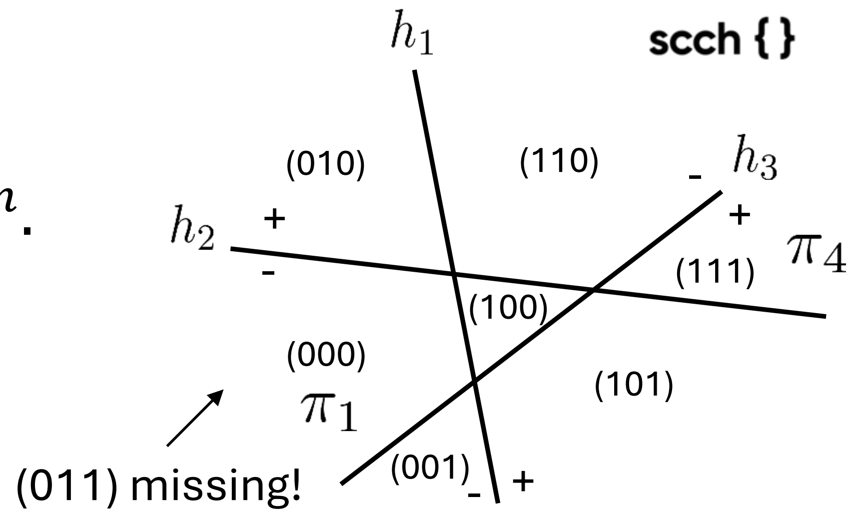
Example:

Let $\pi_1 = 0\mathbf{10}1, \pi_2 = 1\mathbf{10}0$. Every $\pi \in I(\pi_1, \pi_2)$ is of form $x\mathbf{10}y$, thus $I(\pi_1, \pi_2) = \{\pi_1, 0\mathbf{10}0, 1\mathbf{10}1, \pi_2\}$.

Equivalence of Convexity Notions

The Hamming activation space contains every $\pi \in \{0,1\}^m$.
BUT not every π is a pre-image of a linear region!

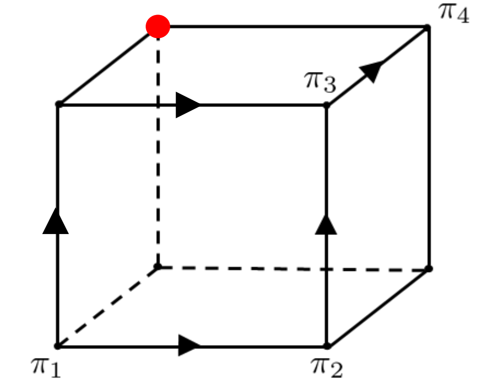
Impacts the definition of convexity in the Hamming space [1, 2]



$$(\mathcal{X}, d_E) \longrightarrow (\{0,1\}^m, d_H)$$

Convexity ✓ | Convexity ?

"line segment between any two points lies in the set"



Definition 2. A subset S of the Hamming cube H^m is convex if, for every pair of $\pi_i, \pi_j \in S$, all <u>realizable</u> π's on <u>every shortest path</u> between π_i, π_j are also in S.

Lemma 1: Let Ω be a set of realizable activation patterns of an affine hyperplane arrangement. For $B \subseteq \Omega$ define

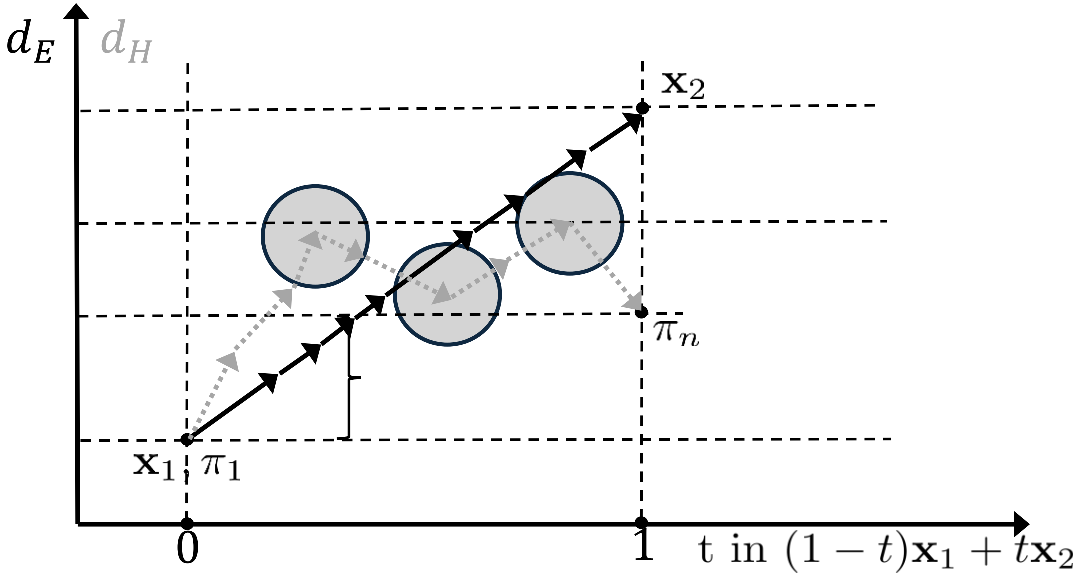
$$R_B = \bigcup_{\pi \in B} R_\pi .$$

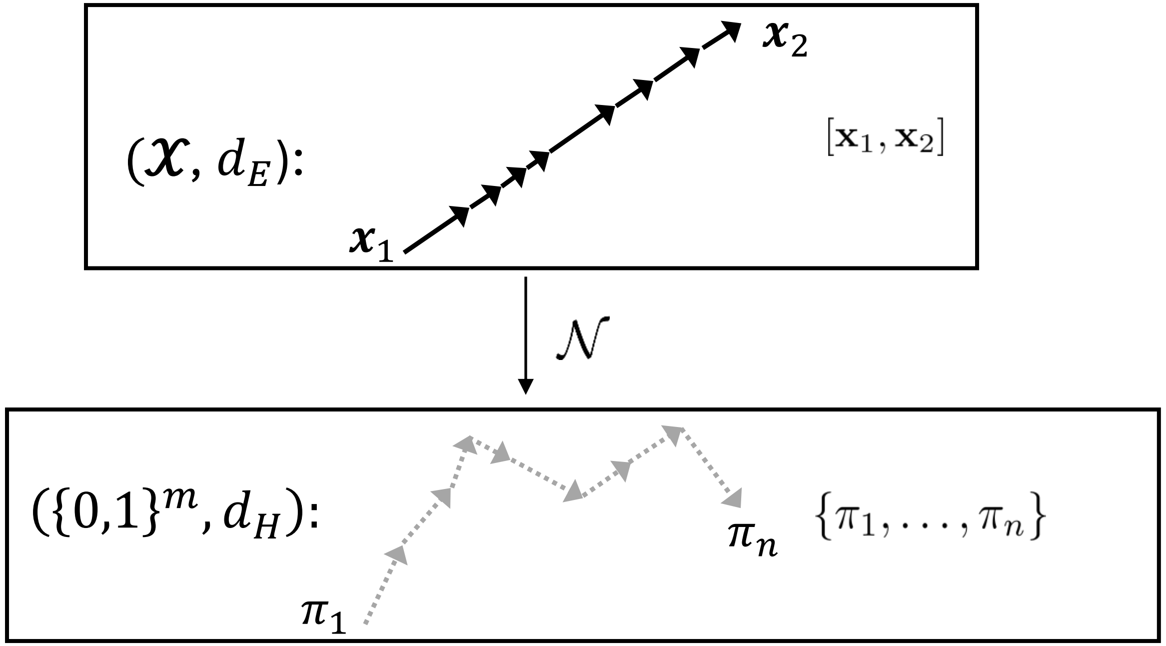
Then, R_B is convex $\Leftrightarrow$ B is Hamming-convex relative to Ω.

[1] B.A.Moser et al. (2022). *Tessellation Filtering ReLU Networks*. IJCAI.
[2] M.Lewandowski et al. (2025). *On Space Folds of ReLU Neural Networks*. TMLR.

Lemma 1 holds for 1-layer ReLU NN, <u>not</u> deeper!

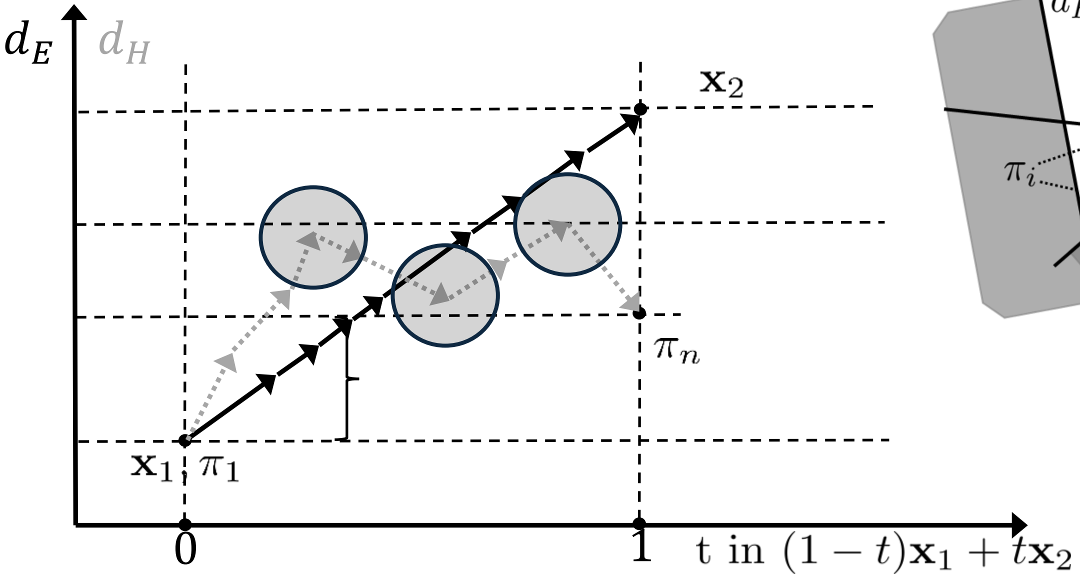
Mappings of Paths: Connection to Convexity

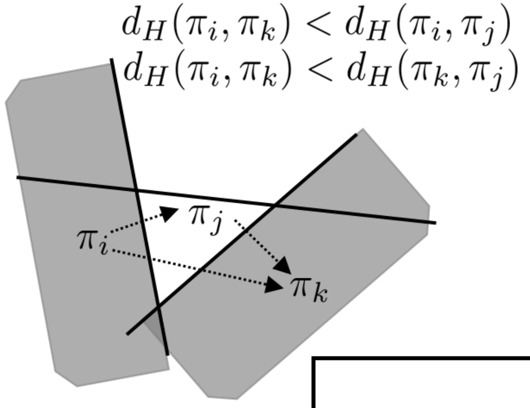


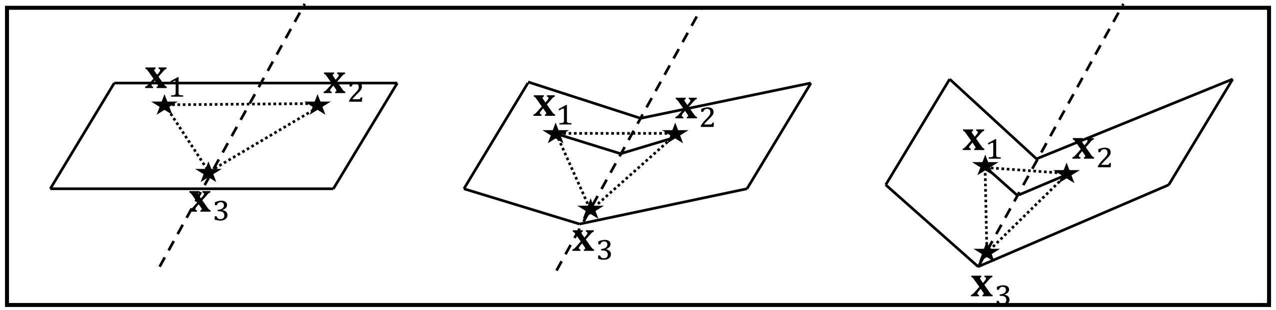


- Sample $y_i := t_i\mathbf{x_1} + (1 - t_i)\mathbf{x_2}$ by varying $t_i \in [0,1]$: the *smallest* convex set containing $\mathbf{x_1}$ and $\mathbf{x_2}$;
- *Map* y_i from $(\mathcal{X}, d_E)$ to $(\{0,1\}^m, d_H)$ under $\mathcal{N}$ at every t_i, obtaining $\Gamma \coloneqq \{\pi_1, \ldots, \pi_n\}$
- Observe that in a general case Γ is not fully contained in $I(\pi_1, \pi_n)$, i.e., Γ is not Hamming-convex
- By studying Γ we can measure *deviations* from convexity between $\mathbf{x}_1$ and $\mathbf{x}_2$

Mappings of Paths: Connection to Folding







- On a path $\Gamma = \{\pi_1, \ldots, \pi_n\}$, we monitor two *range measures:*
  - r_1: Max change in d_H at each step i w.r.t. π_1: $r_1(\Gamma) \coloneqq \max_i d_H(\pi_1, \pi_i)$
  - r_2: Total travelled distance on the hypercube: $r_2(\Gamma) \coloneqq \sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1})$

Local Folding

$$\chi(\Gamma) \coloneqq 1 - \frac{r_1(\Gamma)}{r_2(\Gamma)} \in [0,1]$$

Global Folding

$$\Phi_{\mathcal{N}} \coloneqq median(\{\chi(\Gamma) : \chi(\Gamma) > 0\})$$

Alternatively, map $\sum_{\mathrm{i,j}} \alpha_{\mathrm{j}} \mathbf{x}_{\mathbf{i}}$ through $\mathcal{N}$ for $\forall \mathrm{i}, j$

Space Folding Measure – Properties

scch { }

Association for the Advancement of Artificial Intelligence

Edge cases:

- $\chi(\Gamma) = 0$ if $r_1(\Gamma) = r_2(\Gamma)$
- $\chi(\Gamma) \to 1$ for a looped path: $\max_i d_H(\pi_1, \pi_i) = c$, while $\sum_{i=1}^{n-1} d_H(\pi_i, \pi_{i+1}) \to \infty$.
- $\chi(\Gamma) \geq \frac{1}{2}$ if the path Γ starts and finishes in the same activation pattern π_1

The space folding measure has the following properties:

1. **[Stability]** Multiple steps in the same linear region do not influence its values.
2. **[Asymmetry]** The folding measure is sensitive to the direction of path traversal, i.e., $\chi(\Gamma) \neq \chi(-\Gamma)$, where $-\Gamma = \{\pi_n, \ldots, \pi_1\}$.
3. **[Flatness Invariance]** $\chi(\Gamma) = 0$ if and only if $\chi(-\Gamma) = 0$.
4. **[Non-additivity]** Is neither sub- nor super-additive.

Let $\Gamma = \Gamma(\mathbf{x}_1, \mathbf{x}_2)$ be a path spanned between the edge points $\mathbf{x}_1, \mathbf{x}_2$.
Proposition. Let d_χ be a symmetrized space folding measure,

$$d_\chi(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2}\Big(\chi\big(\Gamma(\mathbf{x}_1, \mathbf{x}_2)\big) + \chi\big(\Gamma(\mathbf{x}_2, \mathbf{x}_1)\big)\Big).$$

Then, d_χ is a pseudo-metric: **(1) Positivity**: from bounds on χ; **(2) Symmetry**: from construction; **(3) Triangle inequality**: from the triangle inequality of d_H

Algorithm & Complexity

Algorithm 1: Computation of the Space Folding Measure

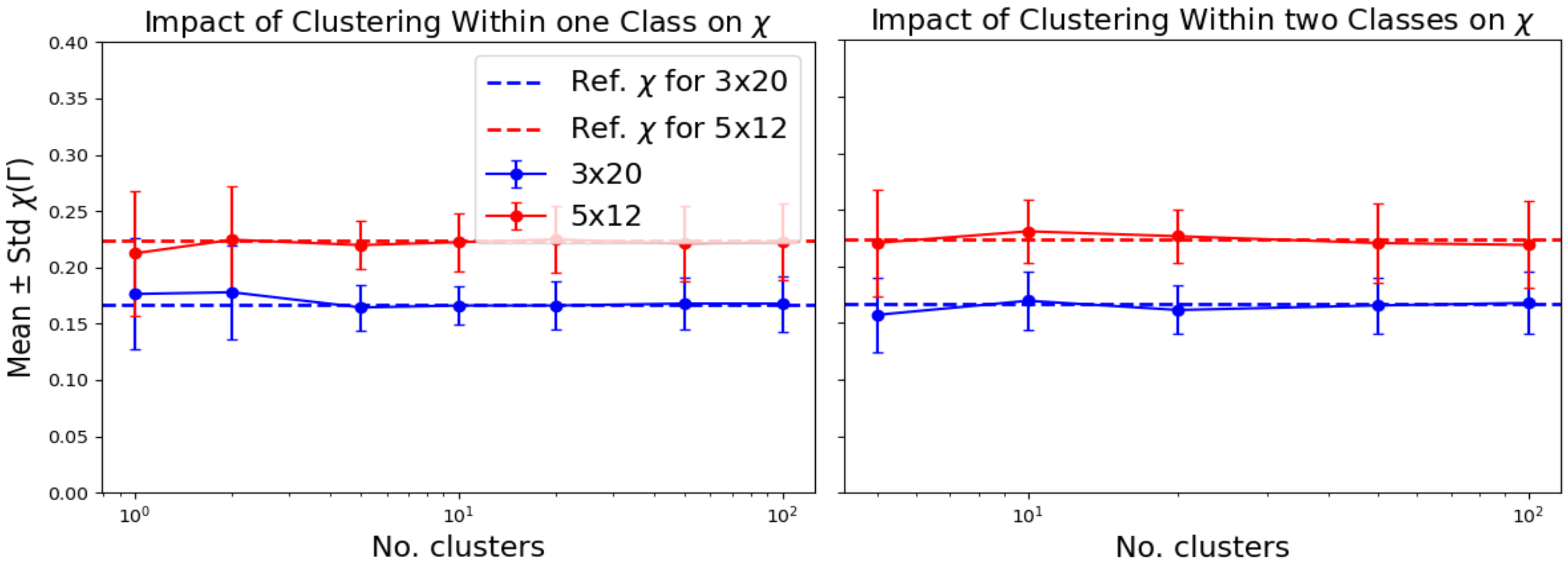
Input: Two inputs x_1, x_2; number of intermediate points n; total number of hidden neurons m; inference cost $\mathcal{O}(C)$.
Output: Space Folding measure $\chi(\Gamma)$.

Step 1: Linearly interpolate x_1 and x_2, sampling n points $z_1, \ldots, z_n$; `// Sampling:` $\mathcal{O}(n)$
Step 2: **for** $k = 1, \ldots, n$ **do**
  Compute activation binarization $\pi_k := b(z_k) \in \{0,1\}^m$; `// Per point:` $\mathcal{O}(C)$
end for `// Total binarization:` $\mathcal{O}(nC)$
Step 3: Compute $r_1(\Gamma)$ and $r_2(\Gamma)$ along $\Gamma := (\pi_1, \ldots, \pi_n)$; `// Range measures:` $\mathcal{O}(nm)$
return $\chi(\Gamma)$; `// Total:` $\mathcal{O}(n(m+C))$

Computational complexity:

$$O(n \cdot (m + C) \cdot |C_1| \cdot |C_2|)$$

- For a dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{k}$
- Cluster $\mathbf{x}$ with the same y_i
- Use clusters' centroids instead $\mathbf{x}_i$



Space Folding - Caveats

scch { }

Association for the Advancement of Artificial Intelligence

- For an activation function f, we considered its pre-image thresholded at 0, i.e.,
$$f^{(-1)}\big((-\infty,0]\big) \mapsto 0 \text{ and } f^{(-1)}\big((0,\infty)\big) \mapsto 1$$
- Why thresholding at 0?
  - consider thresholds $a, b \in (-\infty, \infty)$ in the range of the activation function f such that $|a| < |b|$,
  - $\#a$-induced regions $\geq \#b$-induced regions $\Rightarrow$ a-induced folding measure $\geq$ b-induced folding measure;
  - for ReLU $a = 0$ is "more informative" than any $b > 0$

- Space Folding computation is based on a walk traversing linear regions
- This can be extended to a walk traversing *equivalence classes* [1,2]:
$$\mathbf{x_1} \sim_{\mathcal{N}} \mathbf{x_2} \Leftrightarrow d_H(\pi_1, \pi_2) = 0$$
- For ReLU NN, $[\mathbf{x}]_{\mathcal{N}} \coloneqq \{\mathbf{z}: \mathbf{z} \sim_{\mathcal{N}} \mathbf{x}\}$ correspond to linear regions
- For non-monotonous f equivalence classes may be topologically disconnected, but the construction applies as is

[1] N. Shepeleva, et al. (2020). *Relu code space: a basis for rating network quality besides accuracy*. ICLR Workshop.
[2] M. Lewandowski, et al. (2025). *Exploiting Space Folding by Neural Networks*. AAAI.

Space Folding: Possible Extensions

scch { }

Association for the Advancement of Artificial Intelligence

Beyond fully connected networks

- Consider a **subset of layers** – different underlying tessellation
- **Skip Connections** – applies as is
- **Convolutional layers** – after convolution and pooling

Low sensitivity to batch-norm, dropout

[1] M. Lewandowski, et al. (2025). Exploiting Space Folding by Neural Networks. AAAI.

Interaction Coefficient and Adversarial Examples

χ is neither sub- nor super-additive - define *interaction coefficient*

$$I(k) \coloneqq |\chi(\Gamma_1 \oplus \Gamma_2) - \chi(\Gamma_1) - \chi(\Gamma_2)|,$$

where $k \in \mathrm{N}$ is the connecting index of paths Γ_1 and Γ_2, i.e.,

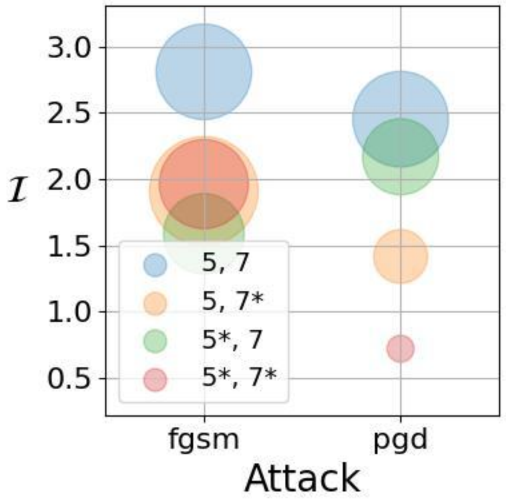
$$\Gamma_1 = \{\pi_1, \dots, \boldsymbol{\pi_k}\}, \Gamma_2 = \{\boldsymbol{\pi_k}, \dots, \pi_n\}, \Gamma_1 \oplus \Gamma_2 = \{\pi_1, \dots, \boldsymbol{\pi_k}, \dots, \pi_n\}.$$

- I computed using digits 5 and 7 from the MNIST test set; $*$ denotes adversarial perturbation
- I is consistently higher for unperturbed samples

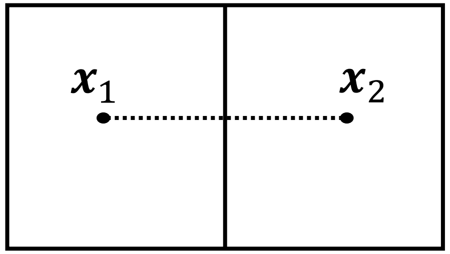
Algorithm 1: $\mathcal{I}$ and Adversarial Attacks

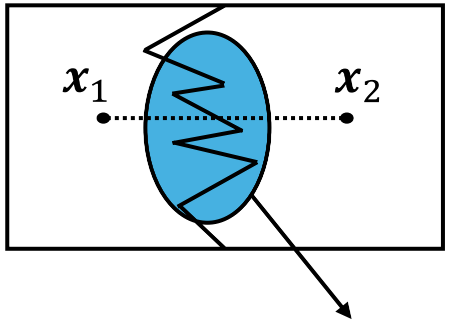
Input: Dataset $\{(\mathbf{x}_i, y_i)\}_{i \in I}$; number of path samples n.
Output: Mean non-zero $\mathcal{I}$.

Step 1: For an input $\mathbf{x}$, compute an adversarial point $\mathbf{x}^*$.
Step 2: If $\mathcal{N}(\mathbf{x}) = \mathcal{N}(\mathbf{x}^*)$, discard the pair.
Step 3: Sample the path $[\mathbf{x}, \mathbf{x}^*]$ using $\mathbf{z}_j = (1 - t_j)\mathbf{x} + t_j\mathbf{x}^*$.
  Compute $\mathcal{I}(j)$ for $j = 1, \dots, n$.
  Let $k_0 = \min\{j : \mathcal{I}_j \neq 0\}$.
return $\overline{\mathcal{I}}_{\mathrm{nz}} = \frac{1}{n - k_0} \sum_{j=k_0+1}^{n} \mathcal{I}_j$.



Intuition: folding and adversarial geometry





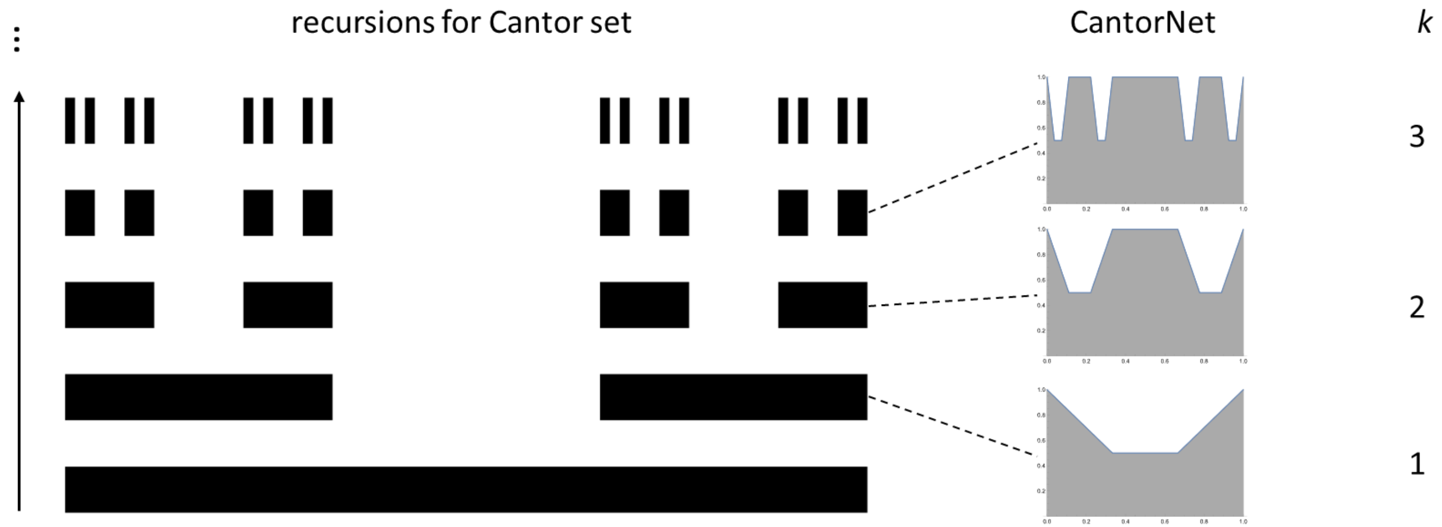
We can measure this!

Space Folding and Generalization – Towards NAS

- Folding values increase with the depth of the network *conditioned on* high validation accuracy
- On CIFAR100, we observed a steady increase in $\Phi_{\mathcal{N}}$ during training

Same output function yet different architectures?

- Studied in [1] based on CantorNet [2];
- higher folding values for less Kolmogorov-complex representation



[1] M.Lewandowski et al. (2025). *On Space Folds of Neural Networks*. TMLR.
[2] M.Lewandowski et al. (2024). *CantorNet: A Sandbox For Testing Topological and Geometrical Complexity Measures*. NeurIPS Workshop.

Summary

scch { }

> The question:
>
> Given a space folding value $\chi(\Gamma) = \tau \in [0, 1]$ for some path Γ, what can we learn?

- χ can be seen as a feature of the network
- χ is upper- and lower-bounded - a reference point across neural networks
- **Interpretation**:
  - Higher χ → indicates a more compact representation
  - Lower χ → indicates potential architecture improvements for the task

[1] M.Lewandowski et al. (2025). *On Space Folds of Neural Networks*. TMLR.

Motzkin-Straus View of Space Folding

scch { }

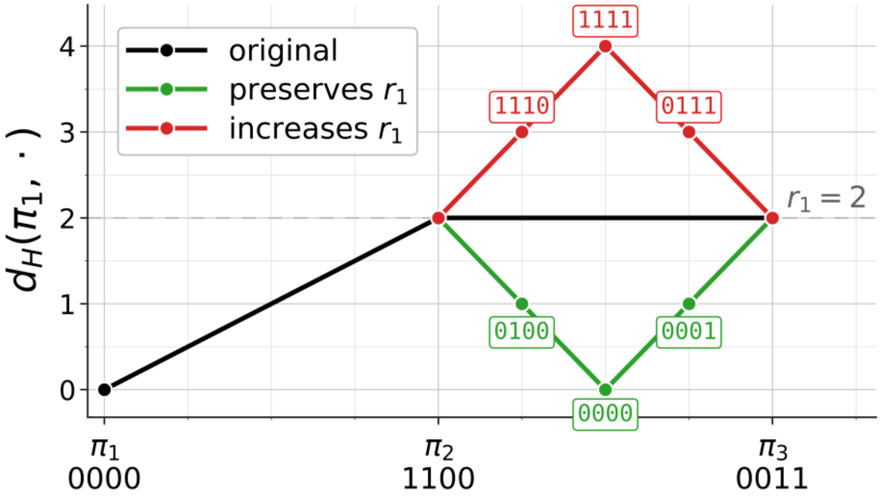
Input segment $[x_1, x_2]$ → **Activation Walk:** $\Gamma = \{\pi_1, \dots, \pi_n\}$ → **Refine Walk to** preserve r_1 and r_2 → **Create graph** G^* from mismatch intervals → Max overlap = clique number $\omega(G^*)$ → **Motzkin-Straus:** $\chi(\Gamma) = f(\Lambda(G^*))$

Lemma (Informal): Refine Γ such that

$$D_i \coloneqq \{j : \pi_i[j] \neq \pi_1[j] \text{ and } \pi_{i+1}[j] = \pi_1[j]\} \text{ (distance decreases)}$$

$$I_i \coloneqq \{j : \pi_i[j] = \pi_1[j] \text{ and } \pi_{i+1}[j] \neq \pi_1[j]\} \text{ (distance increases).}$$

Then r_1, r_2 are preserved.



Consider $\Gamma = (0000, 1100, 0011) \Rightarrow r_1(\Gamma) = 2, r_2(\Gamma) = 6$

$$(1100) \to (1110) \to (1111) \to (0111) \to (0011)$$

$d_H(\pi_1, \cdot)$: $2 \to 3 \to 4 \to 3 \to 2 \Rightarrow r_1(\tilde{\Gamma}) = 4 \neq 2 = r_1(\Gamma)$.

- Flipping decreases before increases yields

$$(1100) \to (0100) \to (0000) \to (0001) \to (0011)$$

with distances $d_H(\pi_1, \cdot)$: $2 \to 1 \to 0 \to 1 \to 2$, preserving r_1

[1] M. Lewandowski, et al. (2026). A Graph-Theoretical View of Space Folding via the Motzkin-Straus Framework. ICLR Workshop.

scch { }

Motzkin-Straus View of Space Folding

Input segment $[x_1, x_2]$ → **Activation Walk:** $\Gamma = \{\pi_1, \dots, \pi_n\}$ → **Refine Walk** to preserve r_1 and r_2 → **Create graph** G^* from mismatch intervals → Max overlap = clique number $\omega(G^*)$ → **Motzkin-Straus:** $\chi(\Gamma) = f(\Lambda(G^*))$

For a refined walk $\tilde{\Gamma}$, for each coordinate of $\pi \in \tilde{\Gamma}$ define *mismatch indicators* as

$$m_j(t) := 1\big(\pi_{t,j} \neq \pi_{1,j}\big).$$

neuron j disagrees with $\pi_{1,j}$

By construction, $d_H(\pi_1, \pi_t) = \sum_j m_j(t)$.

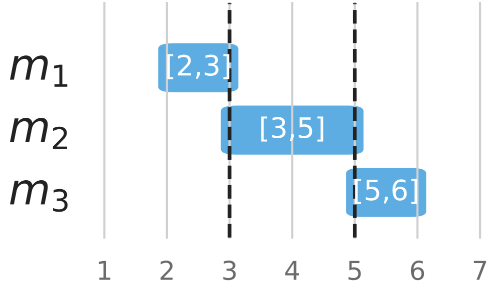
$d_H(\pi_1, \pi_t) = \#$ {coordinates j whose mismatch indicators $m_j(t) = 1$} $= \#\{j : m_j(t) = 1\}$

Example:

π_1	π_2	π_3	π_4	π_5	π_6	π_7
(000)	(100)	(110)	(010)	(011)	(001)	(000)

$m_1 : 0, \mathbf{1}, \mathbf{1}, 0, 0, 0, 0;$
$m_2 : 0, 0, \mathbf{1}, \mathbf{1}, \mathbf{1}, 0, 0;$
$m_3 : 0, 0, 0, 0, \mathbf{1}, \mathbf{1}, 0$

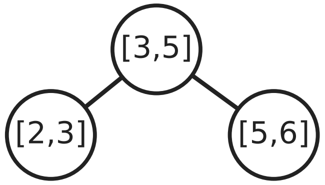
$j \in \{1,2,3\}$
$t \in \{1, \dots, 7\}$



Mismatch intervals

Now <u>build a graph</u>:

- make one vertex for each interval;
- draw an edge between two vertices when the corresponding intervals overlap in t

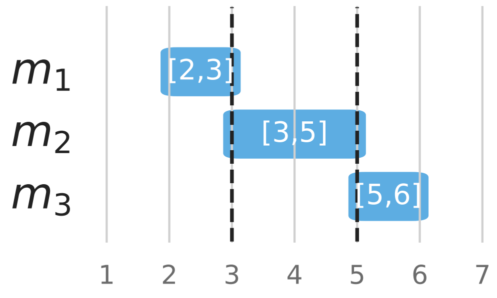


Interval graph

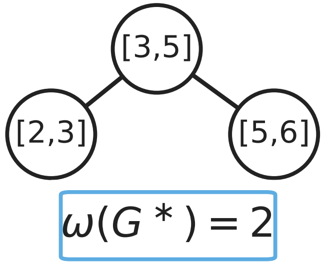
Motzkin-Straus View of Space Folding

Input segment $[x_1, x_2]$ → **Activation Walk:** $\Gamma = \{\pi_1, \dots, \pi_n\}$ → **Refine Walk** to preserve r_1 and r_2 → **Create graph** G^* from mismatch intervals → Max overlap = clique number $\omega(G^*)$ → **Motzkin-Straus:** $\chi(\Gamma) = f(\Lambda(G^*))$

- A *clique* is a set of vertices where every pair is connected
- The *clique number* $\omega(G)$ is the maximum size of a clique $\in G$ (it is a graph invariant)
- Here, a clique means a set of intervals that pairwise overlap



Mismatch intervals



Interval graph

For interval graphs, pairwise overlap ⇒ maximum simultaneous overlap

A clique of size k corresponds to k mismatch intervals being active at the same time. Thus

$$r_1(\widetilde{\Gamma}) = \max_t \#\{j : m_t(j) = 1\} = \text{largest clique size} = \omega(G^*)$$

Motzkin-Straus turns $\omega(G)$ into a quadratic optimization problem over a probability simplex.

T. S. Motzkin and E.G. Straus. *Maxima for graphs and a new proof of a theorem of turan*. Canadian Journal of Mathematics, 1965.

Motzkin-Straus View of Space Folding

Input segment $[x_1, x_2]$ → **Activation Walk:** $\Gamma = \{\pi_1, \dots, \pi_n\}$ → **Refine Walk** to preserve r_1 and r_2 → **Create graph** G^* from mismatch intervals → Max overlap = clique number $\omega(G^*)$ → **Motzkin-Straus:** $\chi(\Gamma) = f(\Lambda(G^*))$

Thm. For the refined walk $\tilde{\Gamma}$ of length $L = r_2(\Gamma)$ there exists a graph G^* s.t.

$$\chi(\Gamma) = 1 - \frac{1}{L(1 - \Lambda(G^*))}$$

where $\Lambda(G^*)$ is the Motzkin-Straus graph Lagrangian.

Consequences: new language to describe space folding

- folding becomes maximum overlap which becomes clique number
- clique number has a classical continuous optimization representation

Path-based folding $\chi(\Gamma)$ → **Overlap / clique view** $\omega(G^*)$ = max overlap → **Quadratic objective** $\Lambda(G^*)$

[1] M. Lewandowski, et al. (2026). *A Graph-Theoretical View of Space Folding via the Motzkin-Straus Framework*. ICLR Workshop.

Applications to Mechanistic Interpretability ?

scch { }

Mechanistic interpretability: an effort to understand how a neural network (e.g., a language model) produces its output

- Open Problems:
  - *Poly-semanticity* [1-4]: single neurons mix unrelated concepts
  - *Locality of concepts* (e.g., [5]): can we avoid generating harmful content?
- Polytopes *often* map to mono-semantic regions [6]
- Use different primitives than neurons, e.g., activation regions, trajectories, locality

[1] Neel Nanda, et al. *Toy Models of Superposition*. Transformer Circuits Thread, 2022.
[2] Trenton Bricken, et al. *Towards Monosemanticity: Decomposing Language Models with Dictionary Learning*. Transformer Circuits Thread, 2023.
[3] R.Huben, et al. *Sparse Autoencoders Find Highly Interpretable Features in Language Models*. ICLR, 2024.
[4] J.Dunefsky, et al. *Transcoders find interpretable LLM feature circuits*. NeurIPS, 2024.
[5] X.Chen, et al. *Learning safety constraints for LLMs*. ICML, 2025.
[6] J.Fernando, G.Guitchounts. *Transformer Dynamics: A neuroscientific approach to interpretability of large language models*. arXiv, 2025.