

Group invariant machine learning on pure maths datasets

Daniel Platt (Imperial College London)

8 Feb 2024

The University of Hong Kong

Abstract: It is a recent trend to use machine learning on pure maths datasets, for example to approximately compute geometric invariants of spaces that are expensive to compute exactly. Often, the map taking some representation of a space to its geometric invariants is invariant under some group action. A common example is that the input space is represented by a matrix and the map is invariant under row and column permutations. I report on some work comparing group invariant and ordinary machine learning models on such datasets. We find that models that are approximately group invariant perform better than fully group invariant models and better than models that are not invariant at all. I will explain one such "approximately group invariant" machine learning model in detail. This is based on two joint works: one published paper with B. Aslan, D. Sheard, and one unpublished work in progress with C. Ewert, S. Magruder, V. Maiboroda, Y. Shen, P Singh.

Motivation

Geometric objects

!

Numerical invariants

- | Topological space
- | Complex manifold
- | Knot

- | Betti numbers
- | Hodge numbers
- | Jones polynomial

- | **String theory**: find complex manifolds with large Hodge number and other prescribed properties [He et al., 2014, p.7]
- | Billions of candidates, single computation can take days [Aggarwal et al., 2023]
- | Idea: machine learning computes numerical **fast but approximately**
y identify most promising candidates
- | (Bonus motivation: machine learning may suggest new theorems/ways to compute invariants, e.g. [Coates et al., 2023, Davies et al., 2021, Dong et al., 2023])

Motivation

Geometric objects

!

Numerical invariants

- | Topological space
- | Complex manifold
- | Knot

- | Betti numbers
- | Hodge numbers
- | Jones polynomial

- | **String theory**: find complex manifolds with large Hodge number and other prescribed properties [He et al., 2014, p.7]
- | Billions of candidates, single computation can take days [Aggarwal et al., 2023]
- | Idea: machine learning computes numerical **fast but approximately**
y identify most promising candidates
- | (Bonus motivation: machine learning may suggest new theorems/ways to compute invariants, e.g. [Coates et al., 2023, Davies et al., 2021, Dong et al., 2023])

Motivation

Geometric objects

!

Numerical invariants

- | Topological space
- | Complex manifold
- | Knot

- | Betti numbers
- | Hodge numbers
- | Jones polynomial

- | **String theory**: nd complex manifolds with large Hodge number and other prescribed properties [He et al., 2014, p.7]
- | Billions of candidates, single computation can take days [Aggarwal et al., 2023]
- | Idea: machine learning computes numerical **fast but approximately** identify most promising candidates
- | (Bonus motivation: machine learning may suggest new theorems/ways to compute invariants, e.g. [Coates et al., 2023, Davies et al., 2021, Dong et al., 2023])

Motivation

Geometric objects

!

Numerical invariants

- | Topological space
- | Complex manifold
- | Knot

- | Betti numbers
- | Hodge numbers
- | Jones polynomial

- | **String theory**: nd complex manifolds with large Hodge number and other prescribed properties [He et al., 2014, p.7]
- | Billions of candidates, single computation can take days [Aggarwal et al., 2023]
- | Idea: machine learning computes numerical **fast but approximately**
y identify most promising candidates
- | (Bonus motivation: machine learning may suggest new theorems/ways to compute invariants, e.g. [Coates et al., 2023, Davies et al., 2021, Dong et al., 2023])

Motivation

Geometric objects

!

Numerical invariants

- | Topological space
- | Complex manifold
- | Knot

- | Betti numbers
- | Hodge numbers
- | Jones polynomial

- | **String theory**: nd complex manifolds with large Hodge number and other prescribed properties [He et al., 2014, p.7]
- | Billions of candidates, single computation can take days [Aggarwal et al., 2023]
- | Idea: machine learning computes numerical **fast but approximately** identify most promising candidates
- | (Bonus motivation: machine learning may suggest new theorems/ways to compute invariants, e.g. [Coates et al., 2023, Davies et al., 2021, Dong et al., 2023])

Motivation

Geometric objects

!

Numerical invariants

- | Topological space
- | Complex manifold
- | Knot

- | Betti numbers
- | Hodge numbers
- | Jones polynomial

- | **String theory**: nd complex manifolds with large Hodge number and other prescribed properties [He et al., 2014, p.7]
- | Billions of candidates, single computation can take days [Aggarwal et al., 2023]
- | Idea: machine learning computes numerical **fast but approximately** identify most promising candidates
- | (Bonus motivation: machine learning may suggest new theorems/ways to compute invariants, e.g. [Coates et al., 2023, Davies et al., 2021, Dong et al., 2023])

Neural networks

- Data $(x_i; y_i) \in \mathbb{R}^k \times \mathbb{R}$ for $i = 1; \dots; N$. Find $f : \mathbb{R}^k \rightarrow \mathbb{R}$ s.t. $f(x_i) = y_i$
- Linear regression: let $W \in \mathbb{R}^{k \times 1}$ (view as $W : \mathbb{R}^k \rightarrow \mathbb{R}^1$) minimise

\mathbb{R}^k

\mathbb{R}

$$\min_{W \in \mathbb{R}^{k \times 1}} \sum_{i=1}^N \|W \cdot x_i - y_i\|^2$$

- Neural network: let $g : \mathbb{R} \rightarrow \mathbb{R}$ be non-linear, e.g. $g(x) = \text{ReLU}(x) := \max(0; x)$. Let $W^j \in \mathbb{R}^{j \times k}$ and $b^j \in \mathbb{R}^{1 \times j}$ minimise

\mathbb{R}^k
 (\cdot)
 0

\mathbb{R}

$$\min_{\substack{W^j \in \mathbb{R}^{j \times k} \\ b^j \in \mathbb{R}^{1 \times j}}} \sum_{i=1}^N \|g(W^j \cdot x_i + b^j) - y_i\|^2$$

Neural networks

- | Data $(x_i; y_i) \in \mathbb{R}^k \times \mathbb{R}$ for $i = 1; \dots; N$. Find $f : \mathbb{R}^k \rightarrow \mathbb{R}$ s.t. $f(x_i) = y_i$
- | Linear regression: let $w \in \mathbb{R}^{k-1}$ (view as $\beta : \mathbb{R}^k \rightarrow \mathbb{R}^1$) minimise

$$\min_{\beta \in \mathbb{R}^{k-1}} \sum_{i=1}^N (\beta \cdot x_i - y_i)^2$$

- | **Neural network:** let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be non-linear, e.g. $\sigma(x) = \text{ReLU}(x) := \max(0; x)$. Let $w^j \in \mathbb{R}^{j-k}$ and $b^j \in \mathbb{R}^1$ minimise

$$\min_{\{w^j, b^j\}} \sum_{i=1}^N \sigma(b^j + w^j \cdot x_i)^2$$

Neural networks

- Data $(x_i; y_i) \in \mathbb{R}^k \times \mathbb{R}$ for $i = 1; \dots; N$. Find $f : \mathbb{R}^k \rightarrow \mathbb{R}$ s.t. $f(x_i) = y_i$
- Linear regression: let $\theta \in \mathbb{R}^{k+1}$ (view as $\theta : \mathbb{R}^k \rightarrow \mathbb{R}^1$) minimise

\mathbb{R}^k

\mathbb{R}

$$\min_{\theta \in \mathbb{R}^{k+1}} \sum_{i=1}^N \theta^T x_i - y_i)^2$$

- Neural network:** let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be non-linear, e.g. $\sigma(x) = \text{ReLU}(x) := \max(0; x)$. Let $\theta \in \mathbb{R}^j$ and $\theta^0 \in \mathbb{R}^1$ minimise

\mathbb{R}^k
 (\cdot)

0

\mathbb{R}

$$\min_{\theta \in \mathbb{R}^j, \theta^0 \in \mathbb{R}^1} \sum_{i=1}^N (\sigma(\theta^0 + \theta^T x_i) - y_i)^2$$

Group actions

- | Example: S_3 = permutation group of 3 elements

$S_3 \curvearrowright \mathbb{R}^3$, e.g. $(1; 2) \cdot (x_1; x_2; x_3) = (x_2; x_1; x_3)$

- | $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ group invariant; $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

- | Example:

$$\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} \max_{g \in S_3} f(x_1; x_2; x_3g)$$

- | Given many pairs $((x_1; x_2; x_3); \max_{g \in S_3} f(x_1; x_2; x_3g))$ can train neural network NN
- | Approximate max, but **need not be group invariant**
- | Q1: how to find **group invariant NNs**?
- | Q2: does this **improve performance** of NNs?

Group actions

- | Example: S_3 = permutation group of 3 elements

$S_3 \curvearrowright \mathbb{R}^3$, e.g. $(1; 2) \cdot (x_1; x_2; x_3) = (x_2; x_1; x_3)$

- | $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ group invariant ; $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

- | Example:

$$\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} \max_{g \in S_3} f(x_1; x_2; x_3g)$$

- | Given many pairs $((x_1; x_2; x_3); \max_{g \in S_3} f(x_1; x_2; x_3g))$ can train neural network NN
- | Approximate max, but **need not be group invariant**
- | Q1: how to find **group invariant NNs**?
- | Q2: does this **improve performance** of NNs?

Group actions

- | Example: S_3 = permutation group of 3 elements

$S_3 \curvearrowright \mathbb{R}^3$, e.g. $(1; 2) \cdot (x_1; x_2; x_3) = (x_2; x_1; x_3)$

- | $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ group invariant ; $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

- | Example:

$$\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$$

- | Given many pairs $((x_1; x_2; x_3); \max_{g \in S_3} f(x_1; x_2; x_3; g))$ can train neural network NN
- | Approximate max, but need not be group invariant
- | Q1: how to find group invariant NNs?
- | Q2: does this improve performance of NNs?

Group actions

- | Example: S_3 = permutation group of 3 elements
 $S_3 \curvearrowright \mathbb{R}^3$, e.g. $(1; 2) \cdot (x_1; x_2; x_3) = (x_2; x_1; x_3)$

- | $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ group invariant ; $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- | Example:

$$\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$$

- | Given many pairs $(x_1; x_2; x_3)$; $\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$ can train neural network NN
- | Approximate max, but need not be group invariant
- | Q1: how to find group invariant NNs?
- | Q2: does this improve performance of NNs?

Group actions

- | Example: S_3 = permutation group of 3 elements

$S_3 \curvearrowright \mathbb{R}^3$, e.g. $(1; 2) \cdot (x_1; x_2; x_3) = (x_2; x_1; x_3)$

- | $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ group invariant ; $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

- | Example:

$$\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$$

- | Given many pairs $(x_1; x_2; x_3)$; $\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$ can train neural network NN

- | Approximate max, but **need not be group invariant**

- | Q1: how to find **group invariant NNs**?

- | Q2: does this **improve performance** of NNs?

Group actions

- | Example: S_3 = permutation group of 3 elements
 S_3 y \mathbb{R}^3 , e.g. $(1; 2) (x_1; x_2; x_3) = (x_2; x_1; x_3)$

- | $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ group invariant ; $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

- | Example:

$$\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$$

- | Given many pairs $(x_1; x_2; x_3)$; $\max_{(x_1; x_2; x_3) \in \mathbb{R}^3} f(x_1; x_2; x_3)$ can train neural network NN
- | Approximate max, but **need not be group invariant**
- | Q1: how to find **group invariant NNs**?
- | Q2: does this **improve performance** of NNs?

Previous approaches

1. **Data augmentation:** Given many pairs $(x_1; x_2; x_3); \max_{x_1; x_2; x_3} f(x_1; x_2; x_3)$, add pairs $(g(x_1; x_2; x_3); \max_{x_1; x_2; x_3} f(x_1; x_2; x_3))$ for all $g \in S_3$ to the training data
2. **Restricting weights** of neural networks [Zaheer et al., 2017] ("Deep Sets")
3. **Averaging techniques:**
Let $NN : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\mathbb{N} : \mathbb{R}^3 \rightarrow \mathbb{R} \\ (x_1; x_2; x_3) \mapsto \frac{1}{|S_3|} \sum_{g \in S_3} NN(g(x_1; x_2; x_3))$$

) \mathbb{N} is group invariant train \mathbb{N} instead of NN

Previous approaches

1. **Data augmentation:** Given many pairs $(x_1; x_2; x_3); \max_{x_1; x_2; x_3} g$, add pairs $(g \circ (x_1; x_2; x_3); \max_{x_1; x_2; x_3} g)$ for all $g \in S_3$ to the training data
2. **Restricting weights** of neural networks [Zaheer et al., 2017] ("Deep Sets")
3. **Averaging techniques:**
Let $NN : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\begin{aligned} \mathbb{N} : \mathbb{R}^3 &\rightarrow \mathbb{R} \\ (x_1; x_2; x_3) &\mapsto \frac{1}{|S_3|} \sum_{g \in S_3} NN(g \circ (x_1; x_2; x_3)) \end{aligned}$$

) \mathbb{N} is group invariant train \mathbb{N} instead of NN

Previous approaches

1. **Data augmentation:** Given many pairs $(x_1; x_2; x_3; \max\{x_1; x_2; x_3\})$, add pairs $(g(x_1; x_2; x_3); \max\{x_1; x_2; x_3\})$ for all $g \in S_3$ to the training data
2. **Restricting weights** of neural networks [Zaheer et al., 2017] ("Deep Sets")
3. **Averaging techniques:**
Let $NN : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\mathbb{N} : \mathbb{R}^3 \rightarrow \mathbb{R} \\ (x_1; x_2; x_3) \mapsto \frac{1}{|S_3|} \sum_{g \in S_3} NN(g(x_1; x_2; x_3))$$

) \mathbb{N} is group invariant train \mathbb{N} instead of NN

Previous approaches

1. **Data augmentation:** Given many pairs $(x; \max_{x_1, x_2, x_3} f(x_1, x_2, x_3))$, add pairs $(g(x_1, x_2, x_3); \max_{x_1, x_2, x_3} f(x_1, x_2, x_3))$ for all $g \in S_3$ to the training data
2. **Restricting weights** of neural networks [Zaheer et al., 2017] ("Deep Sets")
3. **Averaging techniques:**
Let $NN : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\tilde{NN} : \mathbb{R}^3 \rightarrow \mathbb{R} \\ (x_1; x_2; x_3) \mapsto \frac{1}{|S_3|} \sum_{g \in S_3} NN(g(x_1; x_2; x_3))$$

) \tilde{NN} is group invariant train \tilde{NN} instead of NN

Previous approaches

1. **Data augmentation:** Given many pairs $(x_1; x_2; x_3); \max f(x_1; x_2; x_3; g)$, add pairs $(g(x_1; x_2; x_3); \max f(x_1; x_2; x_3; g))$ for all $g \in S_3$ to the training data
2. **Restricting weights** of neural networks [Zaheer et al., 2017] ("Deep Sets")
3. **Averaging techniques:**
Let $NN : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\begin{aligned} \mathcal{NN} : \mathbb{R}^3 &\rightarrow \mathbb{R} \\ (x_1; x_2; x_3) &\mapsto \frac{1}{|S_3|} \sum_{g \in S_3} NN(g(x_1; x_2; x_3)) \end{aligned}$$

) \mathcal{NN} is group invariant train \mathcal{NN} instead of NN

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \cdot \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3$ & \mathbb{R}^3 , $U := \{x_1, x_2, x_3 \in \mathbb{R}^3 : x_1 > x_2 > x_3\}$

$$F : \mathbb{R}^3 \rightarrow \bar{U}$$
$$\begin{cases} \max\{x_1, x_2, x_3\} \\ \text{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \cdot \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3$ & \mathbb{R}^3 , $U := \{ (x_1; x_2; x_3) \in \mathbb{R}^3 : x_1 > x_2 > x_3 \}$

$$F : \mathbb{R}^3 \rightarrow \bar{U} \\ (x_1; x_2; x_3) \mapsto \begin{cases} \max\{x_1; x_2; x_3\} & 1 \\ \text{middle}\{x_1; x_2; x_3\} & A \\ \min\{x_1; x_2; x_3\} & \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

| Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

| Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

| $U \subset \mathbb{R}^N$ fundamental domain for $G \cdot \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

| $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3$ $\gamma \mathbb{R}^3$, $U := \{ (x_1; x_2; x_3) \in \mathbb{R}^3 : x_1 > x_2 > x_3 \}$

$$F : \mathbb{R}^3 \rightarrow \bar{U} \\ (x_1; x_2; x_3) \mapsto \begin{cases} \max\{x_1; x_2; x_3\} & 1 \\ \text{middle}\{x_1; x_2; x_3\} & A \\ \min\{x_1; x_2; x_3\} & \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \curvearrowright \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3 \curvearrowright \mathbb{R}^3$, $U := \{ (x_1; x_2; x_3) \in \mathbb{R}^3 : x_1 > x_2 > x_3 \}$

$$F : \mathbb{R}^3 \rightarrow \bar{U} \\ (x_1; x_2; x_3) \mapsto \begin{cases} \max\{x_1, x_2, x_3\} & 1 \\ \text{middle}\{x_1, x_2, x_3\} & A \\ \min\{x_1, x_2, x_3\} & \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \curvearrowright \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3 \curvearrowright \mathbb{R}^3$, $U := \{ (x_1; x_2; x_3) \in \mathbb{R}^3 : x_1 > x_2 > x_3 \}$

$$F : \mathbb{R}^3 \rightarrow \bar{U} \\ (x_1; x_2; x_3) \mapsto \begin{cases} \max\{x_1; x_2; x_3\} & 1 \\ \text{middle}\{x_1; x_2; x_3\} & 2 \\ \min\{x_1; x_2; x_3\} & 3 \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \cdot \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and \bar{U}

Example: $G = S_3$ $\gamma \mathbb{R}^3$, $U := \{ (x_1; x_2; x_3) \in \mathbb{R}^3 : x_1 > x_2 > x_3 \}$

$$F : \mathbb{R}^3 \rightarrow \bar{U} \\ (x_1; x_2; x_3) \mapsto \begin{cases} \max\{x_1; x_2; x_3\} \\ \text{middle}\{x_1; x_2; x_3\} \\ \min\{x_1; x_2; x_3\} \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \cdot \mathbb{R}^N$;

1. U open and connected

2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}

3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3$ & \mathbb{R}^3 , $U := \{x_1, x_2, x_3 \in \mathbb{R}^3 : x_1 > x_2 > x_3\}$

$$F : \mathbb{R}^3 \rightarrow \bar{U}$$
$$\{x_1, x_2, x_3\} \mapsto \begin{cases} \max\{x_1, x_2, x_3\} & 1 \\ \text{middle}\{x_1, x_2, x_3\} & 2 \\ \min\{x_1, x_2, x_3\} & 3 \end{cases}$$

New approach: group invariant pre-processing [Aslan et al., 2023]

Take $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$

Neural network \mathcal{N} define $\mathcal{N} := \text{NN} \circ F$

$$\mathcal{N}(g \cdot x) = \text{NN}(F(g \cdot x)) = \text{NN}(F(x)) = \mathcal{N}(x)$$

Train \mathcal{N} instead of NN

(Equivalent: train on data $F(x); y$ rather than $(x; y)$)

How to get good F ?

$U \subset \mathbb{R}^N$ fundamental domain for $G \cdot \mathbb{R}^N$;

1. U open and connected
2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects \bar{U}
3. if $G \cdot x$ intersects U , then the intersection is unique

$F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ def by $x \mapsto \text{intersection of } G \cdot x \text{ and } \bar{U}$

Example: $G = S_3$ & \mathbb{R}^3 , $U := \{x_1 > x_2 > x_3\}$

$$F : \mathbb{R}^3 \rightarrow \bar{U}$$

$$\begin{cases} \max\{x_1, x_2, x_3\} \\ \text{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{cases}$$

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

I Approach 1: Combinatorial Fundamental Domain

[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:
fast combinatorial algorithm to compute U and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$

I Approach 2: Dirichlet Fundamental Domain

$G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|x\| = \|g \cdot x\|$
 $x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \|x - x_0\| \leq \|g \cdot x - x_0\| \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$x \mapsto g \cdot x \text{ where } g \in G \text{ s.t. } \|g \cdot x - x_0\| = \min_{g \in G} \|g \cdot x - x_0\|$$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y; x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order $y_1; y_2; y_3$ s.t. biggest coord first

$\bar{U} = \{y \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

I Approach 1: Combinatorial Fundamental Domain

[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:

fast combinatorial algorithm to compute \bar{U} and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$

I Approach 2: Dirichlet Fundamental Domain

$G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|jg \cdot x\| = \|jg \cdot x_j\|$

$x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \langle h x, x_0 \rangle > \langle h g x, x_0 \rangle \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product

$F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$x \mapsto g x$ where $g \in G$ s.t. $\langle h g x, x_0 \rangle = \max_{g \in G} \langle h g x, x_0 \rangle$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle h y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order $y_1; y_2; y_3$ s.t. biggest coord first

$\bar{U} = \{y \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

I Approach 1: **Combinatorial Fundamental Domain**

[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:

fast combinatorial algorithm to compute \bar{U} and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$

I Approach 2: **Dirichlet Fundamental Domain**

$G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|x_j\| = \|g \cdot x_j\|$

$x_0 \in \mathbb{R}^n$ generic, define

$$U := \{x \in \mathbb{R}^n : \|x - x_0\| < \|g \cdot x - x_0\| \text{ for all } g \in G\}$$

where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$x \in U$ where $g \in G$ s.t. $\|g \cdot x - x_0\| = \min_{g \in G} \|g \cdot x - x_0\|$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y; x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order $y_1; y_2; y_3$ s.t. **biggest coord first**

$\bar{U} = \{y \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

- | Approach 1: **Combinatorial Fundamental Domain**
[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:
fast combinatorial algorithm to compute \bar{U} and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$
- | Approach 2: **Dirichlet Fundamental Domain**
 $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|jg \cdot x_j\| = \|jg \cdot x_j\|$
 $x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \|hx; x_0\| > \|hg \cdot x; x_0\| \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$x \in U$ where $g \in G$ s.t. $\|hg \cdot x; x_0\| = \max_{g \in G} \|hg \cdot x; x_0\|$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y; x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order $y_1; y_2; y_3$ s.t. **biggest coord first**

$\bar{U} = \{f(y_1; y_2; y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

| Approach 1: Combinatorial Fundamental Domain

[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:

fast combinatorial algorithm to compute \bar{U} and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$

| Approach 2: Dirichlet Fundamental Domain

$G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|jx\| = \|x\|$

$x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \|x - x_0\| \leq \|g \cdot x - x_0\| \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$x \in U \iff \exists g \in G \text{ s.t. } \|g \cdot x - x_0\| = \min_{g \in G} \|g \cdot x - x_0\|$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first

$\bar{U} = \{y \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

- | Approach 1: **Combinatorial Fundamental Domain**
[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:
fast combinatorial algorithm to compute U and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$
- | Approach 2: **Dirichlet Fundamental Domain**
 $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|jg \cdot x_j\| = \|jg \cdot x_j\|$
 $x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \|x - x_0\| \leq \|x - g \cdot x_0\| \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$x \mapsto g \cdot x \text{ where } g \in G \text{ s.t. } \|g \cdot x - x_0\| = \min_{g \in G} \|g \cdot x - x_0\|$$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. **biggest coord first**

$\bar{U} = \{y = (y_1; y_2; y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

- | Approach 1: **Combinatorial Fundamental Domain**
[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:
fast combinatorial algorithm to compute \bar{U} and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$
- | Approach 2: **Dirichlet Fundamental Domain**
 $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|jg \cdot x\| = \|jg \cdot x_0\|$
 $x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \|hx; x_0\| > \|hg \cdot x; x_0\| \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$x \mapsto g \cdot x \text{ where } g \in G \text{ s.t. } \|hg \cdot x; x_0\| = \max_{g \in G} \|g \cdot x; x_0\|$$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y; x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order $y_1; y_2; y_3$ s.t. **biggest coord first**

$\bar{U} = \{y = (y_1; y_2; y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$! \mathbb{R}^n ?

| Approach 1: Combinatorial Fundamental Domain

[Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup:

fast combinatorial algorithm to compute \bar{U} and F for the action $G \curvearrowright \mathbb{R}^n$,
we extend to case $G \curvearrowright \mathbb{R}^n$

| Approach 2: Dirichlet Fundamental Domain

$G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $\|jg \cdot x\| = \|jg \cdot x_0\|$

$x_0 \in \mathbb{R}^n$ generic, define

$U := \{x \in \mathbb{R}^n : \|x - x_0\| \leq \|g \cdot x - x_0\| \text{ for all } g \in G\}$; where $\langle \cdot, \cdot \rangle$ is dot product
 $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$x \in U \iff \exists g \in G \text{ s.t. } \|g \cdot x - x_0\| = \min_{g \in G} \|g \cdot x - x_0\|$$

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3; 2; 1)$, project $y = (y_1; y_2; y_3)$

to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order $y_1; y_2; y_3$ s.t. **biggest coord first**

$\bar{U} = \{y = (y_1; y_2; y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

For more general groups

- | Groups can be large, e.g. S_{15} y \mathbb{R}^{15} has $|S_{15}| = 15! \approx 10^{12}$
 -) data augmentation and averaging techniques **impossible**
(NN with restricted weights still possible)
- | Ours can be generalised to G y M for M a complete Riemannian manifold

$$U := \{x \in M : d(x; x_0) < d(g \cdot x; x_0) \text{ for all } g \in G\}$$

e.g. $SL(2; \mathbb{Z})$ y \mathbb{H}^2

- | Remark: for Lie groups G y M : choose U to be **slice**

For more general groups

- | Groups can be large, e.g. S_{15} y \mathbb{R}^{15} has $|S_{15}| = 15! \approx 10^{12}$
 -) data augmentation and averaging techniques **impossible**
(NN with restricted weights still possible)
- | Ours can be generalised to G y M for M a complete Riemannian manifold

$$U := \{x \in M : d(x; x_0) < d(g \cdot x; x_0) \text{ for all } g \in G\}$$

e.g. $SL(2; \mathbb{Z})$ y \mathbb{H}^2

- | Remark: for Lie groups G y M : choose U to be **slice**

For more general groups

- | Groups can be large, e.g. S_{15} y \mathbb{R}^{15} has $|S_{15}| = 15! \approx 10^{12}$
 -) data augmentation and averaging techniques **impossible**
(NN with restricted weights still possible)
- | Ours can be generalised to G y M for M a complete Riemannian manifold

$$U := \{x \in M : d(x; x_0) < d(g \cdot x; x_0) \text{ for all } g \in G\}$$

e.g. $SL(2; \mathbb{Z})$ y \mathbb{H}^2

- | Remark: for Lie groups G y M : choose U to be **slice**

For more general groups

- | Groups can be large, e.g. S_{15} y \mathbb{R}^{15} has $|S_{15}| = 15! \approx 10^{12}$
) data augmentation and averaging techniques **impossible**
(NN with restricted weights still possible)
- | Ours can be generalised to G y M for M a complete Riemannian manifold

$$U := \{x \in M : d(x; x_0) < d(g \cdot x; x_0) \text{ for all } g \in G\}$$

e.g. $SL(2; \mathbb{Z})$ y H^2

- | Remark: for Lie groups G y M : choose U to be **slice**

For more general groups

- | Groups can be large, e.g. S_{15} y \mathbb{R}^{15} has $|S_{15}| = 15! \approx 10^{12}$
 -) data augmentation and averaging techniques **impossible**
(NN with restricted weights still possible)
- | Ours can be generalised to G y M for M a complete Riemannian manifold

$$U := \{x \in M : d(x; x_0) < d(g \cdot x; x_0) \text{ for all } g \in G\}$$

e.g. $SL(2; \mathbb{Z})$ y \mathbb{H}^2

- | Remark: for Lie groups G y M : choose U to be **slice**

Example 1: Rotated MNIST

- 28 28 pixel images showing a digit, possibly rotated by $\theta \in [0, 2\pi)$

Learn

$$h : \mathbb{R}^{28 \times 28} \rightarrow \{0, 1, 2, \dots, 9\}$$

x is the digit shown in x

- Have Z_4 by rotation and h is Z_4 -invariant

(note $Z_4 \subset S_{28 \times 28} = S_{784}$)

- Define U (fundamental domain) and F (projection):

(small lie, x_0 not generic)

$$x_0 = \begin{array}{c|c} \begin{array}{ccc} 0 & & 1 \\ \hline 4 & 4 & \dots \\ 4 & 4 & \dots \\ \vdots & \vdots & \vdots \\ 2 & 2 & \dots \\ 2 & 2 & \dots \\ \vdots & \vdots & \vdots \end{array} & \begin{array}{ccc} 3 & 3 & \dots \\ 3 & 3 & \dots \\ \vdots & \vdots & \vdots \\ 1 & 1 & \dots \\ 1 & 1 & \dots \\ \vdots & \vdots & \vdots \end{array} \end{array}; \quad \bar{U} := \{x \in \mathbb{R}^{28 \times 28} : h(x; x_0) = \max_{g \in Z_4} h(gx; x_0)\}$$

$F : \mathbb{R}^{28 \times 28} \rightarrow \mathbb{R}^{28 \times 28}$; x is rotated so that top left quadrant is brightest

Example 1: Rotated MNIST

- 28 28 pixel images showing a digit, possibly rotated by $\theta \in [0, 2\pi)$

Learn

$$h : \mathbb{R}^{28 \times 28} \rightarrow \{0, 1, 2, \dots, 9\}$$

x is the digit shown in x

- Have Z_4 by rotation and h is Z_4 -invariant (note $Z_4 \subset S_{28 \times 28} = S_{784}$)

- Define U (fundamental domain) and F (projection): (small lie, x_0 not generic)

$$x_0 = \begin{array}{c|ccc} 0 & 4 & 4 & \dots & 3 & 3 & \dots & 1 \\ & 4 & 4 & \dots & 3 & 3 & \dots & \\ \hline & \vdots & \vdots & & \vdots & \vdots & & \\ & 2 & 2 & \dots & 1 & 1 & \dots & \\ & 2 & 2 & \dots & 1 & 1 & \dots & \\ & \vdots & \vdots & & \vdots & \vdots & & \end{array}; \quad \bar{U} := \{x \in \mathbb{R}^{28 \times 28} : h(x; x_0) = \max_{g \in S_4} h(gx; x_0)\}$$

$F : \mathbb{R}^{28 \times 28} \rightarrow \mathbb{R}^{28 \times 28}$; x is rotated so that top left quadrant is brightest

Example 1: Rotated MNIST

- 28 28 pixel images showing a digit, possibly rotated by $\theta \in [0, 2\pi)$

Learn

$$h : \mathbb{R}^{28 \times 28} \rightarrow \{0, 1, 2, \dots, 9\}$$

x is the digit shown in x

- Have Z_4 by rotation and h is Z_4 -invariant

(note $Z_4 \subset S_{28 \times 28} = S_{784}$)

- Define U (fundamental domain) and F (projection):

(small lie, x_0 not generic)

$$x_0 = \begin{array}{c|c} \begin{array}{ccc} 0 & & 1 \\ 4 & 4 & \dots \\ 4 & 4 & \dots \\ \vdots & \vdots & \vdots \\ 2 & 2 & \dots \\ 2 & 2 & \dots \\ \vdots & \vdots & \vdots \end{array} & \begin{array}{ccc} 3 & 3 & \dots \\ 3 & 3 & \dots \\ \vdots & \vdots & \vdots \\ 1 & 1 & \dots \\ 1 & 1 & \dots \\ \vdots & \vdots & \vdots \end{array} \end{array};$$

$$\bar{U} := \{x \in \mathbb{R}^{28 \times 28} : h(x; x_0) = \max_{g \in S_4} h(gx; x_0)\}$$

$F : \mathbb{R}^{28 \times 28} \rightarrow \mathbb{R}^{28 \times 28}$; x is rotated so that top left quadrant is brightest

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

I have procedure $M \in \mathbb{R}^{12 \times 15}$ f_1, \dots, f_{15} polynomials such that

$$\text{CY}(M) := \{x \in \mathbb{C}P^{k_1} \mid \text{CY}^{k_2} : f_1(x) = 0; \dots; f_{15}(x) = 0\}$$

is Calabi-Yau manifold

$$\begin{array}{ccccccccc}
 & & & & & & & & & 1 \\
 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & & \\
 & 0 & 0 & 1 & 0 & 0 & 1 & \dots & & \\
 \text{m} & 0 & 0 & 0 & 0 & 1 & 1 & \dots & \text{C} & \\
 & 1 & 0 & 0 & 1 & 0 & 0 & \dots & \text{C} & \\
 & 1 & 0 & 0 & 0 & 0 & 1 & \dots & \text{C} & \\
 \text{C} & 0 & 0 & 1 & 2 & 0 & 0 & \dots & \text{C} & \\
 & 0 & 1 & 0 & 0 & 2 & 0 & \dots & \text{A} & \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & & \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & &
 \end{array}$$

I geometric invariant "second Hodge number" $h^2 : \text{Calabi-Yau mfg} \rightarrow \mathbb{Z}$

I Learn

$$h : \mathbb{R}^{12 \times 15} \rightarrow \mathbb{Z}$$

$$M \mapsto h^2(\text{CY}(M))$$

I Fact: h invariant under action of $S_{12} \times S_{15}$ acting by row/column permutations

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

- have procedure $M \in \mathbb{R}^{12 \times 15}$ f_1, \dots, f_{15} polynomials such that

$$CY(M) := \{x \in \mathbb{C}P^{k_1} \mid \mathbb{C}P^{k_2} : f_1(x) = 0; \dots; f_{15}(x) = 0\}$$

is Calabi-Yau manifold

$$\begin{matrix} & & & & & & & & 1 \\ & 1 & 1 & 0 & 0 & 0 & 0 & \vdots & \\ & 0 & 0 & 1 & 0 & 0 & 1 & \vdots & \\ \text{m} & 0 & 0 & 0 & 0 & 1 & 1 & \vdots & \text{C} \\ & 1 & 0 & 0 & 1 & 0 & 0 & \vdots & \text{C} \\ & 1 & 0 & 0 & 0 & 0 & 1 & \vdots & \text{C} \\ \text{C} & 0 & 0 & 1 & 2 & 0 & 0 & \vdots & \text{C} \\ & 0 & 1 & 0 & 0 & 2 & 0 & \vdots & \text{A} \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{matrix}$$

- geometric invariant "second Hodge number" h^2 : Calabi-Yau manifold Z
- Learn

$$h : \mathbb{R}^{12 \times 15} \rightarrow \mathbb{Z}$$
$$M \mapsto h^2(CY(M))$$

- Fact: h invariant under action of $S_{12} \times S_{15}$ acting by row/column permutations

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

- have procedure $M \in \mathbb{R}^{12 \times 15}$ $f_1; \dots; f_{15}$ polynomials such that

$$CY(M) := \{x \in \mathbb{C}P^{k_1} \mid \mathbb{C}P^{k_2} : f_1(x) = 0; \dots; f_{15}(x) = 0\}$$

is Calabi-Yau manifold

$$\begin{array}{ccccccc}
 1 & 1 & 0 & 0 & 0 & 0 & \dots & 1 \\
 0 & 0 & 1 & 0 & 0 & 1 & \dots & \\
 0 & 0 & 0 & 0 & 1 & 1 & \dots & \\
 1 & 0 & 0 & 1 & 0 & 0 & \dots & \\
 1 & 0 & 0 & 0 & 0 & 1 & \dots & \\
 0 & 0 & 1 & 2 & 0 & 0 & \dots & \\
 0 & 1 & 0 & 0 & 2 & 0 & \dots & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots &
 \end{array}$$

- geometric invariant "second Hodge number" h^2 : Calabi-Yau manifold Z
- Learn

$$h : \mathbb{R}^{12 \times 15} \rightarrow \mathbb{Z}$$

$$M \mapsto h^2(CY(M))$$

- Fact: h invariant under action of $S_{12} \times S_{15}$ acting by row/column permutations

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

Let $x_0 = \begin{pmatrix} 0 \\ 10^{179} \\ \vdots \\ 10^{29} \\ 10^{14} \end{pmatrix} \begin{matrix} 10^{178} \\ \vdots \\ 10^{28} \\ 10^{13} \end{matrix} \begin{matrix} 10^{177} \\ \vdots \\ 10^{27} \\ 10^{12} \end{matrix} \begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix} \begin{matrix} 10^{165} \\ \vdots \\ 10^{15} \\ 10^0 \end{matrix} \begin{matrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}$

$$U := \{ M \in \mathbb{R}^{12 \times 15} : \langle M; x_0 \rangle > \langle g M; x_0 \rangle \text{ for all } g \in S_{12} \times S_{15} \}$$

$$= \{ M \in \mathbb{R}^{12 \times 15} : M \text{ is lexicographically bigger than } g M \text{ for all } g \in S_{12} \times S_{15} \}$$

$F : M \mapsto$ lexicographically biggest row/column permutation of M

E.g. $F \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}$

Compute F ? For $M \in \mathbb{R}^{12 \times 15}$ apply random permutations until get no bigger
(Side note: computing F is slower than solving graph isomorphism problem)

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

| Let $x_0 = \begin{pmatrix} 0 \\ 10^{179} \\ \vdots \\ 10^{29} \\ 10^{14} \end{pmatrix} \begin{matrix} 10^{178} \\ \vdots \\ 10^{28} \\ 10^{13} \end{matrix} \begin{matrix} 10^{177} \\ \vdots \\ 10^{27} \\ 10^{12} \end{matrix} \begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix} \begin{matrix} 10^{165} \\ \vdots \\ 10^{15} \\ 10^0 \end{matrix} \begin{matrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}$

$U := \{ M \in \mathbb{R}^{12 \times 15} : \langle M, x_0 \rangle > \langle g \cdot M, x_0 \rangle \text{ for all } g \in S_{12} \times S_{15} \}$
 $= \{ M \in \mathbb{R}^{12 \times 15} : M \text{ is lexicographically bigger than } g \cdot M \text{ for all } g \in S_{12} \times S_{15} \}$

| $F : M \mapsto$ lexicographically biggest row/column permutation of M

E.g. $F \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}$

| Compute F ? For $M \in \mathbb{R}^{12 \times 15}$ apply random permutations until get no bigger
 (Side note: computing F is slower than solving graph isomorphism problem)

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

| Let $x_0 = \begin{pmatrix} 0 \\ 10^{179} \\ \vdots \\ 10^{29} \\ 10^{14} \end{pmatrix} \begin{matrix} 10^{178} \\ \vdots \\ 10^{28} \\ 10^{13} \end{matrix} \begin{matrix} 10^{177} \\ \vdots \\ 10^{27} \\ 10^{12} \end{matrix} \begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix} \begin{matrix} 10^{165} \\ \vdots \\ 10^{15} \\ 10^0 \end{matrix} \begin{matrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}$

$U := \{ M \in \mathbb{R}^{12 \times 15} : \langle M, x_0 \rangle > \langle g \cdot M, x_0 \rangle \text{ for all } g \in S_{12} \times S_{15} \}$
 $= \{ M \in \mathbb{R}^{12 \times 15} : M \text{ is lexicographically bigger than } g \cdot M \text{ for all } g \in S_{12} \times S_{15} \}$

| $F : M \rightarrow M^T$ lexicographically biggest row/column permutation M

E.g. $F \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}$

| Compute F ? For $M \in \mathbb{R}^{12 \times 15}$ apply random permutations until get no bigger
 (Side note: computing F is slower than solving graph isomorphism problem)

Example 2: Complete Intersection Calabi-Yau (CICY) matrices

Let $x_0 = \begin{pmatrix} 0 \\ 10^{179} \\ \vdots \\ 10^{29} \\ 10^{14} \end{pmatrix} \begin{matrix} 10^{178} \\ \vdots \\ 10^{28} \\ 10^{13} \end{matrix} \begin{matrix} 10^{177} \\ \vdots \\ 10^{27} \\ 10^{12} \end{matrix} \begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix} \begin{matrix} 10^{165} \\ \vdots \\ 10^{15} \\ 10^0 \end{matrix} \begin{matrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}$

$$U := \{ M \in \mathbb{R}^{12 \times 15} : \forall g \in S_{12} \quad \forall h \in S_{15} \quad M; x_{0i} > h(g; x_{0i}) \}$$

M is lexicographically bigger

$$= \{ M \in \mathbb{R}^{12 \times 15} : \forall g \in S_{12} \quad \forall h \in S_{15} \quad M; x_{0i} > h(g; x_{0i}) \}$$

$F : M \rightarrow M$! lexicographically biggest row/column permutation M

E.g. $F \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}$

Compute F ? For $M \in \mathbb{R}^{12 \times 15}$ apply random permutations until get no bigger
 (Side note: computing F is slower than solving graph isomorphism problem)

Inception

[Erbin and Finotello, 2021]

Example 3: Kreuzer-Skarke toric variety list

- | $M \subset \mathbb{R}^4$ 26
- \$ polytope in \mathbb{R}^4 with 26 vertices
- \$ Toric Fano variety

(Suitable degree) Hypersurface is Calabi-Yau manifold $\mathbb{C}P^1(\mathbb{C}P^1)$

- | Learn

$$h : \mathbb{R}^4 \rightarrow \mathbb{Z}$$
$$M \cong \mathbb{C}P^1(\mathbb{C}P^1)$$

- | x_0, U, F as before

cf.
[Berglund et al., 2021]

Thank you for the attention!

References I

Aggarwal, D., He, Y.-H., Heyes, E., Hirst, E., Earp, H. N. S., and Silva, T. S. (2023).

Machine-learning sasakian and $sp(2)$ topology on contact calabi-yau 7-manifolds.
arXiv preprint arXiv:2310.03064

Aslan, B., Platt, D., and Sheard, D. (2023).

Group invariant machine learning by fundamental domain projections.
In NeurIPS Workshop on Symmetry and Geometry in Neural Representations
pages 181{218. PMLR.





Berglund, P., Campbell, B., and Jejjala, V. (2021).



Machine learning kreuzer{skarbe calabi{yau threefolds.
arXiv preprint arXiv:2112.09117

Coates, T., Kasprzyk, A. M., and Veneziale, S. (2023).

Machine learning detects terminal singularities.
In NeurIPS PMLR.

References II

-  Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., et al. (2021). Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74.
-  Dixon, J. D. and Majeed, A. (1988). Coset representatives for permutation groups. *Portugaliae mathematica*, 45:61–68.
-  Dong, B., He, X., Jin, P., Schremmer, F., and Yu, Q. (2023). Machine learning assisted exploration for affine deligne-lusztig varieties.
-  Erbin, H. and Finotello, R. (2021). Machine learning for complete intersection calabi-yau manifolds: a methodological study. *Physical Review D*, 103(12):126014.

-  He, Y.-H., Lee, S.-J., Lukas, A., and Sun, C. (2014).
Heterotic model building: 16 special manifolds.
Journal of High Energy Physics, 2014(6).
-  Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017).
Deep sets.
Advances in neural information processing systems, 30.

- | Polytope image:
https://en.wikipedia.org/wiki/Simple_polytope#/media/File:Associahedron_K5.svg
- | Tessellation of hyperbolic plane:
<https://www.pngwing.com/en/free-png-cmyrj>

This presentation is licensed under Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0).