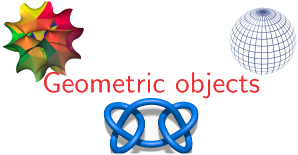
Group invariant machine learning on pure maths datasets

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 This is based on two joint works: one published paper with B. Aslan, D. Sheard, and one unpublished work in progress with C. Ewert, S. Magruder, V. Maiboroda, Y. Shen, P Singh.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



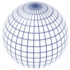
Geometric objects $\longrightarrow$ Numerical invariants

- 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
  $\curvearrowright$ 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

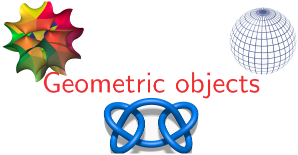
$\longrightarrow$ Numerical invariants

- 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
  $\curvearrowright$ 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



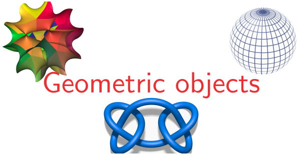
Geometric objects $\longrightarrow$ Numerical invariants

- 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 $\curvearrowright$ 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



$\longrightarrow$

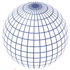
Numerical invariants

- 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
  $\curvearrowright$ 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

$\longrightarrow$

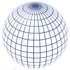
Numerical invariants

- 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
  $\curvearrowright$ 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

$\longrightarrow$

Numerical invariants

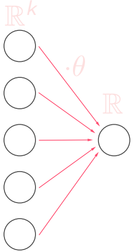
- Topological space
- Complex manifold
- Knot

- 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
  $\curvearrowright$ 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])

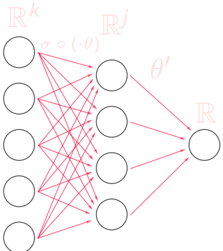
Neural networks

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



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

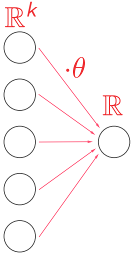
- Neural network: let $\sigma : \mathbb{R} \to \mathbb{R}$ be non-linear, e.g. $\sigma(x) = \mathrm{ReLU}(x) := \max(0, x)$. Let $\theta \in \mathbb{R}^{j\times k}$ and $\theta' \in \mathbb{R}^{1\times j}$ minimise



$$\min_{\substack{\theta\in\mathbb{R}^{j\times k}\\ \theta'\in\mathbb{R}^{1\times j}}} \sum_{i=1}^{N} |\theta'\cdot(\sigma(\theta\cdot x_i)) - y_i|^2$$

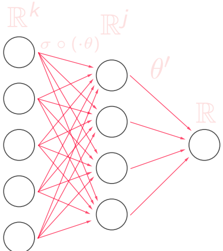
Neural networks

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



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

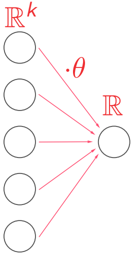
- Neural network: let $\sigma : \mathbb{R} \to \mathbb{R}$ be non-linear, e.g. $\sigma(x) = \mathrm{ReLU}(x) := \max(0, x)$. Let $\theta \in \mathbb{R}^{j\times k}$ and $\theta' \in \mathbb{R}^{1\times j}$ minimise



$$\min_{\substack{\theta\in\mathbb{R}^{j\times k}\\ \theta'\in\mathbb{R}^{1\times j}}} \sum_{i=1}^{N} \left|\theta'\cdot(\sigma(\theta\cdot x_i)) - y_i\right|^2$$

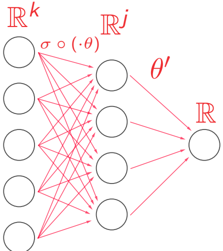
Neural networks

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



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

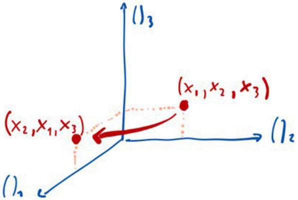
- Neural network: let $\sigma : \mathbb{R} \to \mathbb{R}$ be non-linear, e.g. $\sigma(x) = \mathrm{ReLU}(x) := \max(0, x)$. Let $\theta \in \mathbb{R}^{j \times k}$ and $\theta' \in \mathbb{R}^{1 \times j}$ minimise



$$\min_{\substack{\theta \in \mathbb{R}^{j \times k} \\ \theta' \in \mathbb{R}^{1 \times j}}} \sum_{i=1}^{N} \left|\theta' \cdot (\sigma(\theta \cdot x_i)) - y_i\right|^2$$

Group actions

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



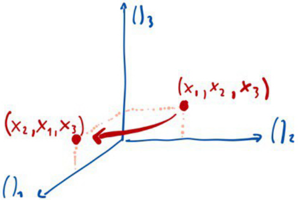
- $f : \mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow f(g\cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

$$\max : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1,x_2,x_3) \mapsto \max\{x_1,x_2,x_3\}$$

- Given many pairs $((x_1,x_2,x_3), \quad \max\{x_1,x_2,x_3\})$ 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. $(1,2)\cdot(x_1,x_2,x_3) = (x_2,x_1,x_3)$



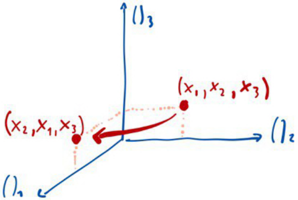
- $f:\mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow f(g\cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

$$\max : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1,x_2,x_3) \mapsto \max\{x_1,x_2,x_3\}$$

- Given many pairs $((x_1,x_2,x_3), \quad \max\{x_1,x_2,x_3\})$ 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. $(1,2)\cdot(x_1,x_2,x_3) = (x_2,x_1,x_3)$



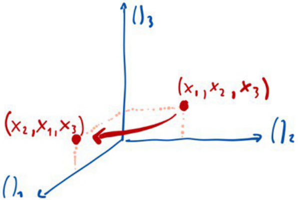
- $f : \mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow f(g\cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

$$\max : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1,x_2,x_3) \mapsto \max\{x_1,x_2,x_3\}$$

- Given many pairs $((x_1,x_2,x_3), \quad \max\{x_1,x_2,x_3\})$ 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. $(1,2) \cdot (x_1, x_2, x_3) = (x_2, x_1, x_3)$



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

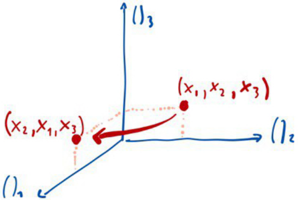
$$\max : \mathbb{R}^3 \to \mathbb{R}$$

$$(x_1, x_2, x_3) \mapsto \max\{x_1, x_2, x_3\}$$

- Given many pairs $((x_1, x_2, x_3), \quad \max\{x_1, x_2, x_3\})$ 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. $(1,2) \cdot (x_1, x_2, x_3) = (x_2, x_1, x_3)$



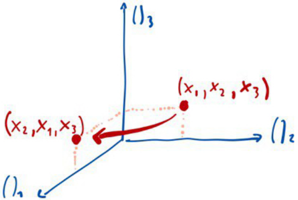
- $f : \mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

$$\max : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto \max\{x_1, x_2, x_3\}$$

- Given many pairs $((x_1, x_2, x_3), \quad \max\{x_1, x_2, x_3\})$ 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. $(1,2)\cdot(x_1,x_2,x_3) = (x_2,x_1,x_3)$



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

$$\max : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1,x_2,x_3) \mapsto \max\{x_1,x_2,x_3\}$$

- Given many pairs $((x_1,x_2,x_3),\quad \max\{x_1,x_2,x_3\})$ 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 $((x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$, add pairs $(g \cdot (x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ 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 $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\widetilde{NN} : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto \sum_{g \in S_3} NN(g \cdot (x_1, x_2, x_3))$$

$\Rightarrow$ $\widetilde{NN}$ is group invariant $\rightsquigarrow$ train $\widetilde{NN}$ 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 $(g \cdot (x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ 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 $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\widetilde{NN} : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto \sum_{g \in S_3} NN(g \cdot (x_1, x_2, x_3))$$

$\Rightarrow$ $\widetilde{NN}$ is group invariant $\rightsquigarrow$ train $\widetilde{NN}$ 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 $(g \cdot (x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ 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 $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\widetilde{NN} : \mathbb{R}^3 \to \mathbb{R}$$

$$(x_1, x_2, x_3) \mapsto \sum_{g \in S_3} NN(g \cdot (x_1, x_2, x_3))$$

$\Rightarrow$ $\widetilde{NN}$ is group invariant $\rightsquigarrow$ train $\widetilde{NN}$ 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 $(g \cdot (x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ 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 $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\widetilde{NN} : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto \sum_{g \in S_3} NN(g \cdot (x_1, x_2, x_3))$$

$\Rightarrow \widetilde{NN}$ is group invariant $\rightsquigarrow$ train $\widetilde{NN}$ 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 $(g \cdot (x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ 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 $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant

$$\widetilde{NN} : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto \sum_{g \in S_3} NN(g \cdot (x_1, x_2, x_3))$$

$\Rightarrow$ $\widetilde{NN}$ is group invariant $\rightsquigarrow$ train $\widetilde{NN}$ instead of NN

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

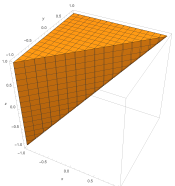
$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN(F(g \cdot x)) = NN(F(x)) = \widetilde{NN}(x)$$

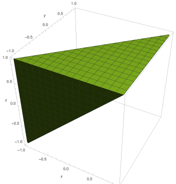
  Train $\widetilde{NN}$ instead of NN
  (Equivalent: train on data $(F(x), y)$ rather than (x, y))

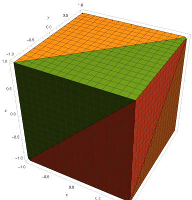
How to get good F?

- $U \subset \mathbb{R}^N$ fundamental domain for $G \curvearrowright \mathbb{R}^N :\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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







$F : \mathbb{R}^3 \to \overline{U}$

$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \text{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN(F(g \cdot x)) = NN(F(x)) = \widetilde{NN}(x)$$

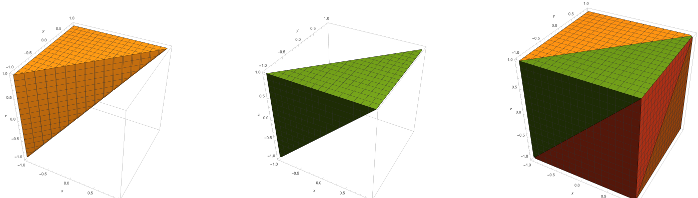
Train $\widetilde{NN}$ instead of NN

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

How to get good F?

- $U \subset \mathbb{R}^N$ fundamental domain for $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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



$$F : \mathbb{R}^3 \to \overline{U}$$

$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \mathrm{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN\big(F(g \cdot x)\big) = NN\big(F(x)\big) = \widetilde{NN}(x)$$

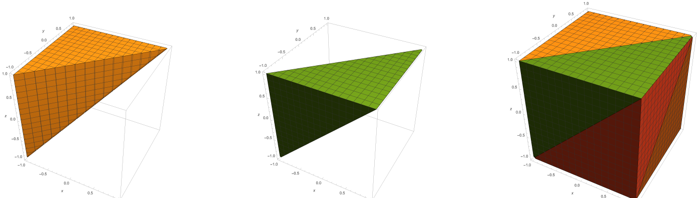
Train $\widetilde{NN}$ instead of NN

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

How to get good F?

- $U \subset \mathbb{R}^N$ fundamental domain for $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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



$$F : \mathbb{R}^3 \to \overline{U}$$
$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \mathrm{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

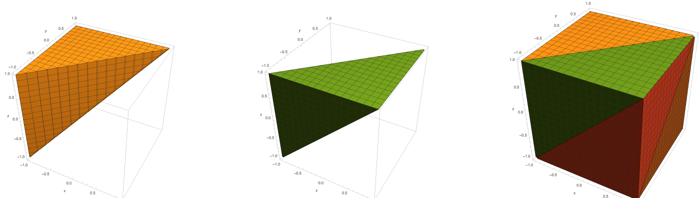
$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN\big(F(g \cdot x)\big) = NN\big(F(x)\big) = \widetilde{NN}(x)$$

Train $\widetilde{NN}$ 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 $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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



$F : \mathbb{R}^3 \to \overline{U}$

$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \mathrm{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

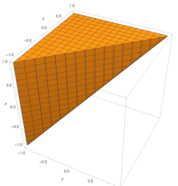
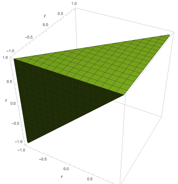
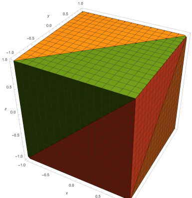
$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN\big(F(g \cdot x)\big) = NN\big(F(x)\big) = \widetilde{NN}(x)$$

Train $\widetilde{NN}$ 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 $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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

$F : \mathbb{R}^3 \to \overline{U}$

$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \mathrm{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

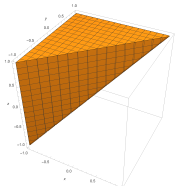
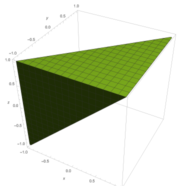
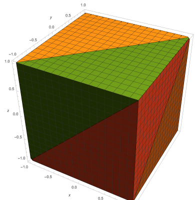
$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN\big(F(g \cdot x)\big) = NN\big(F(x)\big) = \widetilde{NN}(x)$$

  Train $\widetilde{NN}$ 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 $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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

$F : \mathbb{R}^3 \to \overline{U}$
$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \text{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

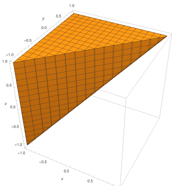
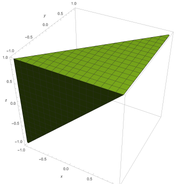
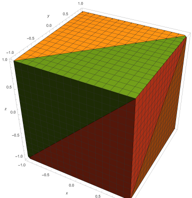
$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN\big(F(g \cdot x)\big) = NN\big(F(x)\big) = \widetilde{NN}(x)$$

  Train $\widetilde{NN}$ 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 $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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

$F : \mathbb{R}^3 \to \overline{U}$

$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \mathrm{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

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

- Take $F : \mathbb{R}^3 \to \mathbb{R}^3$ s.t. $F(g \cdot x) = F(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Neural network $NN \rightsquigarrow$ define $\widetilde{NN} := NN \circ F$

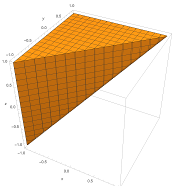
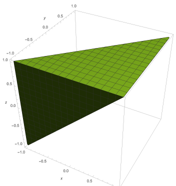
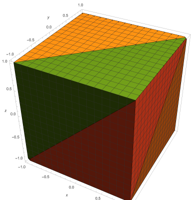
$$\Rightarrow \quad \widetilde{NN}(g \cdot x) = NN\big(F(g \cdot x)\big) = NN\big(F(x)\big) = \widetilde{NN}(x)$$

  Train $\widetilde{NN}$ 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 $G \curvearrowright \mathbb{R}^N$ $:\Leftrightarrow$
  1. U open and connected
  2. for all $x \in X$ the orbit $G \cdot x := \{g \cdot x : g \in G\}$ intersects $\overline{U}$
  3. if $G \cdot x$ intersects U, then the intersection is unique
- $F : \mathbb{R}^N \to \mathbb{R}^N$ def by $x \mapsto$ intersection of $G \cdot x$ and $\overline{U}$

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

$$F : \mathbb{R}^3 \to \overline{U}$$
$$(x_1, x_2, x_3) \mapsto \begin{pmatrix} \max\{x_1, x_2, x_3\} \\ \mathrm{middle}\{x_1, x_2, x_3\} \\ \min\{x_1, x_2, x_3\} \end{pmatrix}$$

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
$\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

  e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
  to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
  $\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
$\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
$\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

  e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
  to maximise $\langle y, x_0\rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
  $\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
$\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
$\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

How to get $F : \mathbb{R}^N \to \mathbb{R}^N$?

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

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

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

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

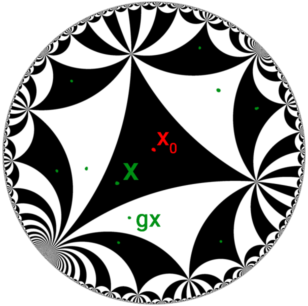
e.g. $S_3 \curvearrowright \mathbb{R}^3$, $x_0 = (3, 2, 1)$, project $y = (y_1, y_2, y_3)$
to maximise $\langle y, x_0 \rangle = 3y_1 + 2y_2 + y_3$ want to order y_1, y_2, y_3 s.t. biggest coord first
$\rightsquigarrow \overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$ same as before!

For more general groups

- Groups can be large, e.g. $S_{15} \curvearrowright \mathbb{R}^{15}$ has $|S_{15}| = 15! \approx 10^{12}$
  $\Rightarrow$ data augmentation and averaging techniques impossible
  (NN with restricted weights still possible)
- Ours can be generalised to $G \curvearrowright M$ 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. $SL(2, \mathbb{Z}) \curvearrowright \mathbb{H}^2$



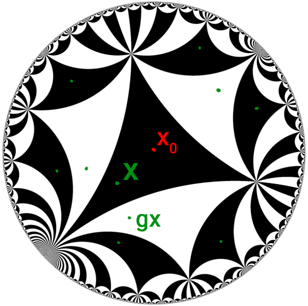
- Remark: for Lie groups $G \curvearrowright M$: choose U to be *slice*

For more general groups

- Groups can be large, e.g. $S_{15} \curvearrowright \mathbb{R}^{15}$ has $|S_{15}| = 15! \approx 10^{12}$
  $\Rightarrow$ data augmentation and averaging techniques impossible
  (NN with restricted weights still possible)
- Ours can be generalised to $G \curvearrowright M$ 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. $SL(2, \mathbb{Z}) \curvearrowright \mathbb{H}^2$



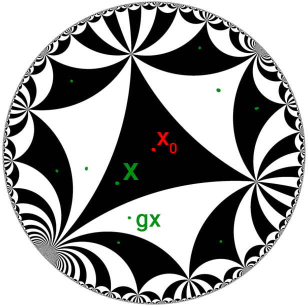
- Remark: for Lie groups $G \curvearrowright M$: choose U to be *slice*

For more general groups

- Groups can be large, e.g. $S_{15} \curvearrowright \mathbb{R}^{15}$ has $|S_{15}| = 15! \approx 10^{12}$
  $\Rightarrow$ data augmentation and averaging techniques impossible
  (NN with restricted weights still possible)
- Ours can be generalised to $G \curvearrowright M$ 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. $SL(2, \mathbb{Z}) \curvearrowright \mathbb{H}^2$



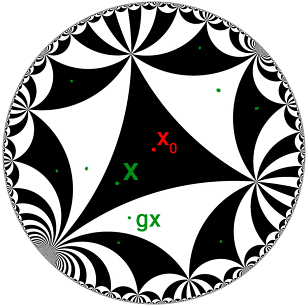
- Remark: for Lie groups $G \curvearrowright M$: choose U to be *slice*

For more general groups

- Groups can be large, e.g. $S_{15} \curvearrowright \mathbb{R}^{15}$ has $|S_{15}| = 15! \approx 10^{12}$
  $\Rightarrow$ data augmentation and averaging techniques impossible
  (NN with restricted weights still possible)
- Ours can be generalised to $G \curvearrowright M$ 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. $SL(2, \mathbb{Z}) \curvearrowright \mathbb{H}^2$



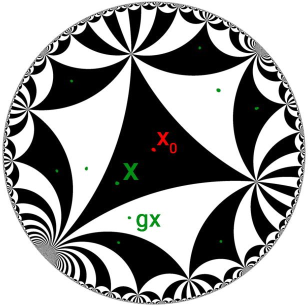
- Remark: for Lie groups $G \curvearrowright M$: choose U to be *slice*

For more general groups

- Groups can be large, e.g. $S_{15} \curvearrowright \mathbb{R}^{15}$ has $|S_{15}| = 15! \approx 10^{12}$
  $\Rightarrow$ data augmentation and averaging techniques impossible
  (NN with restricted weights still possible)
- Ours can be generalised to $G \curvearrowright M$ 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. $SL(2, \mathbb{Z}) \curvearrowright \mathbb{H}^2$



- Remark: for Lie groups $G \curvearrowright M$: choose U to be *slice*

Example 1: Rotated MNIST

- 28×28 pixel images showing a digit, possibly rotated by $90°, 180°, 270°$

- Learn

$$h : \mathbb{R}^{28\times 28} \to \{0, 1, 2, \ldots, 9\}$$
$$x \mapsto \text{the digit shown in } x$$

- Have $\mathbb{Z}_4 \curvearrowright \mathbb{R}^{28\times 28}$ by rotation and h is $\mathbb{Z}_4$-invariant (note $\mathbb{Z}_4 \subset S_{28\cdot 28} = S_{784}$)
- Define U (fundamental domain) and F (projection): (small lie, x_0 not generic)

$$x_0 = \left(\begin{array}{ccc|ccc} 4 & 4 & \cdots & 3 & 3 & \cdots \\ 4 & 4 & \cdots & 3 & 3 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ \hline 2 & 2 & \cdots & 1 & 1 & \cdots \\ 2 & 2 & \cdots & 1 & 1 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \end{array}\right), \quad \overline{U} := \left\{ x \in \mathbb{R}^{28\times 28} : \langle x, x_0 \rangle = \max_{g \in S_4} \langle g \cdot x, x_0 \rangle \right\}$$

$$F : \mathbb{R}^{28\times 28} \to \mathbb{R}^{28\times 28}, \quad x \mapsto x \text{ rotated so that top left quadrant is brightest}$$

Example 1: Rotated MNIST

- 28×28 pixel images showing a digit, possibly rotated by $90^\circ, 180^\circ, 270^\circ$

  - Learn

$$h : \mathbb{R}^{28\times 28} \to \{0, 1, 2, \ldots, 9\}$$
$$x \mapsto \text{the digit shown in } x$$

- Have $\mathbb{Z}_4 \curvearrowright \mathbb{R}^{28\times 28}$ by rotation and h is $\mathbb{Z}_4$-invariant (note $\mathbb{Z}_4 \subset S_{28\cdot 28} = S_{784}$)
- Define U (fundamental domain) and F (projection): (small lie, x_0 not generic)

$$x_0 = \left(\begin{array}{ccc|ccc} 4 & 4 & \cdots & 3 & 3 & \cdots \\ 4 & 4 & \cdots & 3 & 3 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ \hline 2 & 2 & \cdots & 1 & 1 & \cdots \\ 2 & 2 & \cdots & 1 & 1 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \end{array}\right), \qquad \overline{U} := \left\{ x \in \mathbb{R}^{28\times 28} : \langle x, x_0 \rangle = \max_{g \in S_4} \langle g \cdot x, x_0 \rangle \right\}$$

$$F : \mathbb{R}^{28\times 28} \to \mathbb{R}^{28\times 28}, \qquad x \mapsto x \text{ rotated so that top left quadrant is brightest}$$

Example 1: Rotated MNIST

- 28×28 pixel images showing a digit, possibly rotated by $90^\circ, 180^\circ, 270^\circ$

  - Learn

$$h : \mathbb{R}^{28\times 28} \to \{0, 1, 2, \ldots, 9\}$$
$$x \mapsto \text{the digit shown in } x$$

- Have $\mathbb{Z}_4 \curvearrowright \mathbb{R}^{28\times 28}$ by rotation and h is $\mathbb{Z}_4$-invariant
  (note $\mathbb{Z}_4 \subset S_{28\cdot 28} = S_{784}$)
- Define U (fundamental domain) and F (projection):
  (small lie, x_0 not generic)

$$x_0 = \left(\begin{array}{ccc|ccc} 4 & 4 & \ldots & 3 & 3 & \ldots \\ 4 & 4 & \ldots & 3 & 3 & \ldots \\ \vdots & \vdots & & \vdots & \vdots & \\ \hline 2 & 2 & \ldots & 1 & 1 & \ldots \\ 2 & 2 & \ldots & 1 & 1 & \ldots \\ \vdots & \vdots & & \vdots & \vdots & \end{array}\right), \quad \overline{U} := \left\{ x \in \mathbb{R}^{28\times 28} : \langle x, x_0 \rangle = \max_{g \in S_4} \langle g \cdot x, x_0 \rangle \right\}$$

$$F : \mathbb{R}^{28\times 28} \to \mathbb{R}^{28\times 28}, \quad x \mapsto x \text{ rotated so that top left quadrant is brightest}$$

Example 1: Rotated MNIST

- 28×28 pixel images showing a digit, possibly rotated by $90^\circ, 180^\circ, 270^\circ$

	No pre-processing	F
Linear	0.677 ± 0.001	0.784 ± 0.001
MLP	0.939 ± 0.001	0.953 ± 0.003
SimpNet (19)	0.979	0.979

(pre-processing useful for very small models)

  - Learn

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

$$x \mapsto \text{the digit shown in } x$$

- Have $\mathbb{Z}_4 \curvearrowright \mathbb{R}^{28\times 28}$ by rotation and h is $\mathbb{Z}_4$-invariant (note $\mathbb{Z}_4 \subset S_{28\cdot 28} = S_{784}$)
- Define U (fundamental domain) and F (projection): (small lie, x_0 not generic)

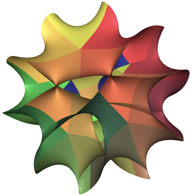
$$x_0 = \left(\begin{array}{ccc|ccc} 4 & 4 & \dots & 3 & 3 & \dots \\ 4 & 4 & \dots & 3 & 3 & \dots \\ \vdots & \vdots & & \vdots & \vdots & \\ \hline 2 & 2 & \dots & 1 & 1 & \dots \\ 2 & 2 & \dots & 1 & 1 & \dots \\ \vdots & \vdots & & \vdots & \vdots & \end{array}\right), \quad \overline{U} := \left\{ x \in \mathbb{R}^{28\times 28} : \langle x, x_0 \rangle = \max_{g \in S_4} \langle g \cdot x, x_0 \rangle \right\}$$

$$F : \mathbb{R}^{28\times 28} \to \mathbb{R}^{28\times 28}, \quad x \mapsto x \text{ rotated so that top left quadrant is brightest}$$

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

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

$$\mathrm{CY}(M) := \{x \in \mathbb{CP}^{k_1} \times \cdots \times \mathbb{CP}^{k_{12}} : f_1(x) = 0, \dots, f_{15}(x) = 0\}$$



is Calabi-Yau manifold

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

- geometric invariant "second Hodge number" $h^2 : \{\text{Calabi-Yau mf}\} \to \mathbb{Z}$
- Learn

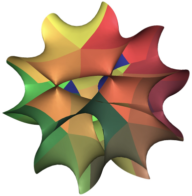
$$h : \mathbb{R}^{12\times 15} \to \mathbb{Z}$$
$$M \mapsto h^2(\mathrm{CY}(M))$$

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

$$\mathrm{CY}(M) := \{x \in \mathbb{CP}^{k_1} \times \cdots \times \mathbb{CP}^{k_{12}} : f_1(x) = 0, \dots, f_{15}(x) = 0\}$$



is Calabi-Yau manifold

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

- geometric invariant "second Hodge number" $h^2 : \{\text{Calabi-Yau mf}\} \to \mathbb{Z}$
- Learn

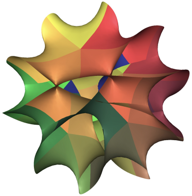
$$h : \mathbb{R}^{12\times 15} \to \mathbb{Z}$$
$$M \mapsto h^2(\mathrm{CY}(M))$$

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

$$\mathrm{CY}(M) := \{x \in \mathbb{CP}^{k_1} \times \cdots \times \mathbb{CP}^{k_{12}} : f_1(x) = 0, \dots, f_{15}(x) = 0\}$$



is Calabi-Yau manifold

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

- geometric invariant "second Hodge number" $h^2 : \{\text{Calabi-Yau mf}\} \to \mathbb{Z}$
- Learn

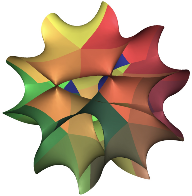
$$h : \mathbb{R}^{12\times 15} \to \mathbb{Z}$$
$$M \mapsto h^2(\mathrm{CY}(M))$$

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

$$\mathrm{CY}(M) := \{x \in \mathbb{CP}^{k_1} \times \cdots \times \mathbb{CP}^{k_{12}} : f_1(x) = 0, \dots, f_{15}(x) = 0\}$$



  is Calabi-Yau manifold

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

- geometric invariant "second Hodge number" $h^2 : \{\text{Calabi-Yau mf}\} \to \mathbb{Z}$
- Learn

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

- 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 $x_0 = \begin{pmatrix} 10^{179} & 10^{178} & 10^{177} & \dots & 10^{165} \\ \vdots & \vdots & \vdots & & \vdots \\ 10^{29} & 10^{28} & 10^{27} & \dots & 10^{15} \\ 10^{14} & 10^{13} & 10^{12} & \dots & 10^{0} \end{pmatrix} \in \mathbb{R}^{12\times 15}$

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

- $F : M \mapsto$ 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 $x_0 = \begin{pmatrix} 10^{179} & 10^{178} & 10^{177} & \dots & 10^{165} \\ \vdots & \vdots & \vdots & & \vdots \\ 10^{29} & 10^{28} & 10^{27} & \dots & 10^{15} \\ 10^{14} & 10^{13} & 10^{12} & \dots & 10^{0} \end{pmatrix} \in \mathbb{R}^{12\times 15}$

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

- $F : M \mapsto$ 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 $x_0 = \begin{pmatrix} 10^{179} & 10^{178} & 10^{177} & \dots & 10^{165} \\ \vdots & \vdots & \vdots & & \vdots \\ 10^{29} & 10^{28} & 10^{27} & \dots & 10^{15} \\ 10^{14} & 10^{13} & 10^{12} & \dots & 10^{0} \end{pmatrix} \in \mathbb{R}^{12\times 15}$

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

- $F : M \mapsto$ 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 $x_0 = \begin{pmatrix} 10^{179} & 10^{178} & 10^{177} & \dots & 10^{165} \\ \vdots & \vdots & \vdots & & \vdots \\ 10^{29} & 10^{28} & 10^{27} & \dots & 10^{15} \\ 10^{14} & 10^{13} & 10^{12} & \dots & 10^{0} \end{pmatrix} \in \mathbb{R}^{12\times 15}$

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

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

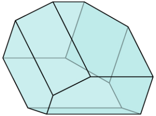
  E.g. $F\begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}$
- Compute F? For $M \in \mathbb{R}^{12\times 15}$ apply random permutations until get no bigger (Side note: computing F is slower than solving graph isomorphism problem)

	Original dataset	Randomly permuted
Inception	0.970 ± 0.009	0.844 ± 0.117
G-inv MLP	0.895 ± 0.029	0.914 ± 0.023
F +Inception	$\mathbf{0.975 \pm 0.007}$	$\mathbf{0.963 \pm 0.016}$

Inception [Erbin and Finotello, 2021]

Example 3: Kreuzer-Skarke toric variety list

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



$\rightsquigarrow$ (Suitable degree) Hypersurface is Calabi-Yau manifold $\mathrm{CY}(M)$

- Learn

$$h : \mathbb{R}^{4\times 26} \to \mathbb{Z}$$
$$M \mapsto h^2(\mathrm{CY}(M))$$

- x_0, U, F as before $\rightsquigarrow$

Model	Accuracy Original Dataset	Accuracy Permuted Dataset
Invariant MLP	$79.30 \pm 0.90\%$	$78.45 \pm 0.92\%$
MLP	$\mathbf{96.86 \pm 0.31\%}$	$92.04 \pm 0.54\%$
MLP+F	$96.66 \pm 0.30\%$	$\mathbf{95.37 \pm 0.37\%}$

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 g_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–skarke 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.

References III

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.

Image credit

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