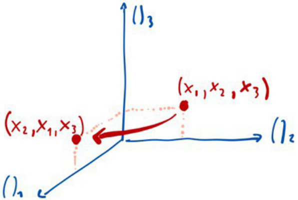
Group invariant machine learning by fundamental domain projections

Daniel Platt
3 May 2023
University of Nottingham Online Machine Learning Seminar

Abstract: In many applications one wants to learn a function that is invariant under a group action. For example, classifying images of digits, no matter how they are rotated. There exist many approaches in the literature to do this. I will mention two approaches that are very useful in many applications, but struggle if the group is big or acts in a complicated way. I will then explain our approach which does not have these two problems. The approach works by finding some "canonical representative" of each input element. In the example of images of digits, one may rotate the digit so that the brightest quarter is in the top-left, which would define a "canonical representative". In the general case, one has to define what that means. Our approach is useful if the group is big, and useless if the group is small, and I will present experiments for both cases. This is joint work with Benjamin Aslan and David Sheard.

Group actions

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



- $f : \mathrm{R}^3\ \ !\ \ \mathrm{R}$ group invariant : , $f(g\ \ x) = f(x)$ for all $g\ 2\ S_3$ and $x\ 2\ \mathrm{R}^3$
- Example:

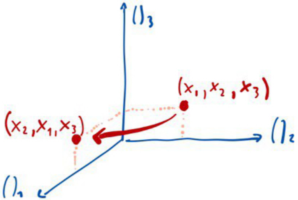
$$\max : \mathrm{R}^3\ !\ \ \mathrm{R}$$

$$(x_1;x_2;x_3)\ 7!\ \ \max f x_1;x_2;x_3 g$$

- Given many pairs $((x_1;x_2;x_3); \max f x_1;x_2;x_3 g)$ can train neural network NN
- Approximate max, but need not be group invariant
- Q1: how can find one 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:

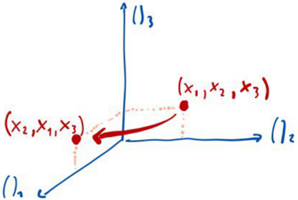
$$\max : \mathbb{R}^3 \to \mathbb{R}$$

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

- Given many pairs $((x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ can train neural network NN
- Approximate max, but need not be group invariant
- Q1: how can find one 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. $(1,2) \cdot (x_1,x_2,x_3) = (x_2,x_1,x_3)$



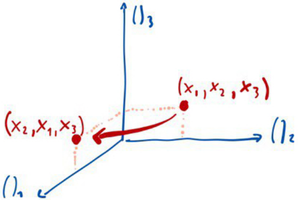
- $f : \mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow$ $f(g \cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

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

- Given many pairs $((x_1,x_2,x_3), \max\{x_1,x_2,x_3\})$ can train neural network NN
- Approximate max, but need not be group invariant
- Q1: how can find one 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. $(1,2)\cdot(x_1,x_2,x_3) = (x_2,x_1,x_3)$



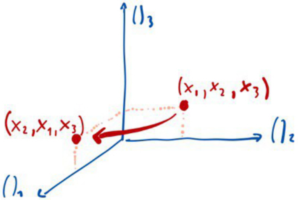
- $f:\mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow$ $f(g\cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

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

- Given many pairs $((x_1,x_2,x_3), \max\{x_1,x_2,x_3\})$ can train neural network NN
- Approximate max, but need not be group invariant
- Q1: how can find one 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. $(1,2)\cdot(x_1,x_2,x_3) = (x_2,x_1,x_3)$



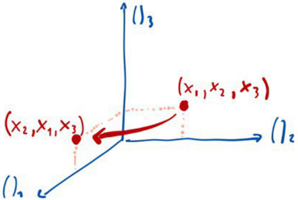
- $f : \mathbb{R}^3 \to \mathbb{R}$ group invariant $:\Leftrightarrow$ $f(g\cdot x) = f(x)$ for all $g \in S_3$ and $x \in \mathbb{R}^3$
- Example:

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

- Given many pairs $((x_1,x_2,x_3),\max\{x_1,x_2,x_3\})$ can train neural network NN
- Approximate max, but need not be group invariant
- Q1: how can find one 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 $((x_1,x_2,x_3),\max\{x_1,x_2,x_3\})$ can train neural network NN
- Approximate max, but need not be group invariant
- Q1: how can find one 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"):
$$L : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto \lambda_1 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ for } \lambda_1, \lambda_2 \in \mathbb{R}$$
has $L(g \cdot x) = g \cdot L(x)$ (equivariant). Define $NN = \text{pool} \circ L \circ \sigma \circ L \circ \cdots \circ \sigma \circ L$, where:
   - pool = some fixed group-invariant function $\mathbb{R}^3 \to \mathbb{R}$, e.g. $(x_1, x_2, x_3) \mapsto x_1 + x_2 + x_3$
   - σ = some non-linearity, e.g. ReLU

   Theorem: If $L : \mathbb{R}^3 \to \mathbb{R}^3$ is S_3-equivariant, then L is of this form.
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 $\Rightarrow$ 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 $(g \circ (x_1, x_2, x_3), \max\{x_1, x_2, x_3\})$ for all $g \in S_3$ to the training data
2. Restricting weights of neural networks [Zaheer et al., 2017] ("Deep Sets"):

$$L : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto \lambda_1 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{for } \lambda_1, \lambda_2 \in \mathbb{R}$$

   has $L(g \circ x) = g \circ L(x)$ (equivariant). Define $NN = \text{pool} \circ L \circ \sigma \circ L \circ \sigma \circ L$, where:
   - pool = some fixed group-invariant function $\mathbb{R}^3 \to \mathbb{R}$, e.g. $(x_1, x_2, x_3) \mapsto x_1 + x_2 + x_3$
   - σ = some non-linearity, e.g. ReLU

   Theorem: If $L : \mathbb{R}^3 \to \mathbb{R}^3$ is S_3-equivariant, then L is of this form.
3. Averaging techniques:
   Let $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant

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

   $\Rightarrow$ $\widetilde{NN}$ is group invariant $\Rightarrow$ 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"):

$$L : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto \lambda_1 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ for } \lambda_1, \lambda_2 \in \mathbb{R}$$

   has $L(g \cdot x) = g \cdot L(x)$ (equivariant). Define $NN = \text{pool} \circ L \circ \sigma \circ L \circ \cdots \circ \sigma \circ L$, where:
   - pool = some fixed group-invariant function $\mathbb{R}^3 \to \mathbb{R}$, e.g. $(x_1, x_2, x_3) \mapsto x_1 + x_2 + x_3$
   - σ = some non-linearity, e.g. ReLU

   Theorem: If $L : \mathbb{R}^3 \to \mathbb{R}^3$ is S_3-equivariant, then L is of this form.
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 $\Rightarrow$ 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 $(g\circ(x_1,x_2,x_3),\max\{x_1,x_2,x_3\})$ for all $g\in S_3$ to the training data
2. Restricting weights of neural networks [Zaheer et al., 2017] ("Deep Sets"):

$$L:\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}\mapsto \lambda_1\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}+\lambda_2\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix} \text{ for } \lambda_1,\lambda_2\in\mathbb{R}$$

   has $L(g\circ x)=g\circ L(x)$ (equivariant). Define $NN=\text{pool}\circ L\circ\sigma\circ L\circ\sigma\circ L$, where:
   - pool $=$some fixed group-invariant function $\mathbb{R}^3\to\mathbb{R}$, e.g. $(x_1,x_2,x_3)\mapsto x_1+x_2+x_3$
   - σ $=$some non-linearity, e.g. ReLU

   Theorem: If $L:\mathbb{R}^3\to\mathbb{R}^3$ is S_3-equivariant, then L is of this form.
3. Averaging techniques:

   Let $NN:\mathbb{R}^3\to\mathbb{R}$ be a neural network architecture, not necessarily invariant

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

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

   $\Rightarrow$ $\widetilde{NN}$ is group invariant $\Rightarrow$ 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"):

$$L : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto \lambda_1 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ for } \lambda_1, \lambda_2 \in \mathbb{R}$$

   has $L(g \cdot x) = g \cdot L(x)$ (equivariant). Define $NN = \text{pool} \circ L \circ \sigma \circ L \circ \sigma \circ L$, where:
   - pool = some fixed group-invariant function $\mathbb{R}^3 \to \mathbb{R}$, e.g. $(x_1, x_2, x_3) \mapsto x_1 + x_2 + x_3$
   - σ = some non-linearity, e.g. ReLU

   Theorem: If $L : \mathbb{R}^3 \to \mathbb{R}^3$ is S_3-equivariant, then L is of this form.
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 $\Rightarrow$ 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"):
$$L : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto \lambda_1 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ for } \lambda_1, \lambda_2 \in \mathbb{R}$$
has $L(g \cdot x) = g \cdot L(x)$ (equivariant). Define $NN = \text{pool} \circ L \circ \sigma \circ L \circ \cdots \circ \sigma \circ L$, where:
   - pool = some fixed group-invariant function $\mathbb{R}^3 \to \mathbb{R}$, e.g. $(x_1, x_2, x_3) \mapsto x_1 + x_2 + x_3$
   - σ = some non-linearity, e.g. ReLU

   Theorem: If $L : \mathbb{R}^3 \to \mathbb{R}^3$ is S_3-equivariant, then L is of this form.
3. Averaging techniques:
Let $NN : \mathbb{R}^3 \to \mathbb{R}$ be a neural network architecture, not necessarily invariant
$$\widetilde{NN} : \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto \sum_{g \in S_3} NN(g \cdot (x_1, x_2, x_3))$$

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"):

$$L : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \mapsto \lambda_1 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{ for } \lambda_1, \lambda_2 \in \mathbb{R}$$

   has $L(g \cdot x) = g \cdot L(x)$ (equivariant). Define $NN = \text{pool} \circ L \circ \sigma \circ L \circ \cdots \circ \sigma \circ L$, where:
   - pool = some fixed group-invariant function $\mathbb{R}^3 \to \mathbb{R}$, e.g. $(x_1, x_2, x_3) \mapsto x_1 + x_2 + x_3$
   - σ = some non-linearity, e.g. ReLU

   Theorem: If $L : \mathbb{R}^3 \to \mathbb{R}^3$ is S_3-equivariant, then L is of this form.
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; 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 define $\widetilde{NN} := NN \circ F$

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

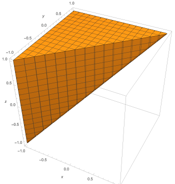
Train $\widetilde{NN}$ instead of NN

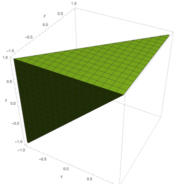
(Equivalent: train on data $(F(x), y)$ rather than (x, y))

How to get good F?

- $U \subset \mathbb{R}^N$ fundamental domain for $G \curvearrowright \mathbb{R}^N$:⇔
  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 → 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?

- $U \subseteq \mathbb{R}^N$ 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 $\overline{U}$
  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 $\Rightarrow$ 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?

- $U \subseteq \mathbb{R}^N$ 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 $\overline{U}$
  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?

- $U \subseteq \mathbb{R}^N$ 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, 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 $G \curvearrowright \mathbb{R}^N$:⟺
  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, 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?

- $U \subseteq \mathbb{R}^N$ 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 $\overline{U}$
  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 $\rightarrow$ 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 $G \curvearrowright \mathbb{R}^N$:⇔
  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, 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]: for any $G \leq S_n$ subgroup: 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
  $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $|x| = |g \cdot x|$
  $x_0 \in \mathbb{R}^n$ generic, define

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

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

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

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

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup: 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
  $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $|x| = |g \cdot x|$
  $x_0 \in \mathbb{R}^n$ generic, define

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

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

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

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

$$\overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\} \text{ same as before!}$$

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988]: for any $G \le S_n$ subgroup: 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
  $G \le S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $|x| = |g \cdot x|$
  $x_0 \in \mathbb{R}^n$ generic, define

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

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

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

  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

  $$\overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\} \text{ same as before!}$$

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup: 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
  $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $|x| = |g \cdot x|$
  $x_0 \in \mathbb{R}^n$ generic, define

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

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

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

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

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup: 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
  $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $|x| = |g \cdot x|$
  $x_0 \in \mathbb{R}^n$ generic, define

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

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

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

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

$$\overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\} \text{ same as before!}$$

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988] for any $G \leq S_n$ subgroup: 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
  $G \leq S_n \curvearrowright \mathbb{R}^n$ acts through isometries, i.e. $|x| = |g \cdot x|$
  $x_0 \in \mathbb{R}^n$ generic, define

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

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

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

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

$$\overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\} \text{ same as before!}$$

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988]: for any $G \subseteq S_n$ subgroup: 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
  $G \subseteq S_n \curvearrowright \mathbb{R}^n$ 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

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

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

- Approach 1: Combinatorial Fundamental Domain
  [Dixon and Majeed, 1988] for any $G \subseteq S_n$ subgroup: 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
  $G \subseteq S_n \curvearrowright \mathbb{R}^n$ 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

$$\overline{U} = \{(y_1, y_2, y_3) \in \mathbb{R}^3 : y_1 \geq y_2 \geq y_3\}$$ 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 $90^\circ, 180^\circ, 270^\circ$

  - 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)

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

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

Example 1: Rotated MNIST

- 28×28 pixel images showing a digit, possibly rotated by $90^\circ, 180^\circ, 270^\circ$
  - Learn
    $$h : \mathbb{R}^{28 \times 28} \to \{0, 1, 2, \dots, 9\}$$
    $$x \mapsto \text{the digit shown in } x$$
- Have $Z_4 \curvearrowright \mathbb{R}^{28 \times 28}$ by rotation and h is Z_4-invariant (note $Z_4 \subset S_{28 \cdot 28} = S_{784}$)
- Define U (fundamental domain) and F (projection): (small lie, x_0 not generic)

$$x_0 = \begin{pmatrix} 4 & 4 & \cdots & 3 & 3 & \cdots \\ 4 & 4 & \cdots & 3 & 3 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ 2 & 2 & \cdots & 1 & 1 & \cdots \\ 2 & 2 & \cdots & 1 & 1 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \end{pmatrix}, \quad \overline{U} := \{x \in \mathbb{R}^{28 \times 28} : \langle x, x_0 \rangle = \max_{g \in S_4} \langle g \circ x, x_0 \rangle\}$$

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

Example 1: Rotated MNIST

- 28×28 pixel images showing a digit, possibly rotated by $90^\circ, 180^\circ, 270^\circ$
  - Learn
    $$h : \mathbb{R}^{28 \times 28} \to \{0, 1, 2, \dots, 9\}$$
    $$x \mapsto \text{the digit shown in } x$$
- Have $\mathbb{Z}_4 \curvearrowright \mathbb{R}^{28 \times 28}$ by rotation and h is $\mathbb{Z}_4$-invariant (note $\mathbb{Z}_4 \le S_{28 \cdot 28} = S_{784}$)
- Define $\bar{U}$ (fundamental domain) and F (projection): (small lie, x_0 not generic)

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

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

Example 1: Rotated MNIST

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

- Learn

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

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

(pre-processing useful for very small models)

- Have $Z_4 \curvearrowright \mathbb{R}^{28 \times 28}$ by rotation and h is Z_4-invariant (note $Z_4 \subseteq S_{28 \cdot 28} = S_{784}$)
- Define U (fundamental domain) and F (projection): (small lie, x_0 not generic)

$$x_0 = \begin{pmatrix} 4 & 4 & \cdots & 3 & 3 & \cdots \\ 4 & 4 & \cdots & 3 & 3 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \\ 2 & 2 & \cdots & 1 & 1 & \cdots \\ 2 & 2 & \cdots & 1 & 1 & \cdots \\ \vdots & \vdots & & \vdots & \vdots & \end{pmatrix}; \quad \overline{U} := \{x \in \mathbb{R}^{28 \times 28} : \langle x, x_0 \rangle = \max_{g \in S_4} \langle g x, x_0 \rangle\}$$

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

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

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

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

  is Calabi-Yau manifold

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

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

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

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

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

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

  is Calabi-Yau manifold

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

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

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

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

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

  is Calabi-Yau manifold

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

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

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

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

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

  is Calabi-Yau manifold

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

- geometric invariant "second Hodge number" $h^2 : \{\text{Calabi-Yau mfd}\} \to \mathbb{Z}$
- 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 $x_0 = \begin{pmatrix} 10^{179} & 10^{178} & 10^{177} & \cdots & 10^{165} \\ \vdots & \vdots & \vdots & & \vdots \\ 10^{29} & 10^{28} & 10^{27} & \cdots & 10^{15} \\ 10^{14} & 10^{13} & 10^{12} & \cdots & 10^{0} \end{pmatrix} \in \mathbb{R}^{12 \times 15}$

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

$$= \left\{ M \in \mathbb{R}^{12 \times 15} : \begin{array}{l} M \text{ is lexicographically smaller} \\ \preceq g \cdot M \text{ for all } g \in S_{12} \times S_{15} \end{array} \right\}$$

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

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

- Compute F? For $M \in \mathbb{R}^{12 \times 15}$ apply random permutations until get no smaller

  (Side note: F in polynomial time $\Leftrightarrow$ graph ismomorphism problem (unsolved))

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

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

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

$$\approx \left\{ M \in \mathbb{R}^{12 \times 15} : \begin{array}{l} M \text{ is lexicographically smaller} \\ g \cdot M \text{ for all } g \in S_{12} \times S_{15} \end{array} \right\}$$

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

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

- Compute F? For $M \in \mathbb{R}^{12 \times 15}$ apply random permutations until get no smaller (Side note: F in polynomial time $\Leftrightarrow$ graph ismomorphism problem (unsolved))

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

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

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

$$= \left\{ M \in \mathbb{R}^{12\times 15} : \begin{array}{l} M \text{ is lexicographically smaller} \\ g \cdot M \text{ for all } g \in S_{12} \times S_{15} \end{array} \right\}$$

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

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

- Compute F? For $M \in \mathbb{R}^{12\times 15}$ apply random permutations until get no smaller

  (Side note: F in polynomial time graph ismomorphism problem (unsolved))

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

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

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

$$= \left\{ M \in \mathbb{R}^{12\times 15} : \begin{array}{l} M \text{ is lexicographically smaller} \\ g\, M \text{ for all } g \in S_{12} \times S_{15} \end{array} \right\}$$

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

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

- Compute F? For $M \in \mathbb{R}^{12\times 15}$ apply random permutations until get no smaller

  (Side note: F in polynomial time $\Leftrightarrow$ graph ismomorphism problem (unsolved))

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

Calabi-Yau manifold $CY(M)$

- Learn

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

- x_0, U, F as before

First line from
[Berglund et al., 2021]

Thank you for the attention!

References I

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.

Dixon, J. D. and Majeed, A. (1988).
Coset representatives for permutation groups.
Portugaliae mathematica, 45:61{68.

Erbin, H. and Finotello, R. (2021).
Machine learning for complete intersection calabi-yau manifolds: a methodological study.
Physical Review D, 103(12):126014.

References II

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