Group Invariant Machine Learning through Near-Isometries*

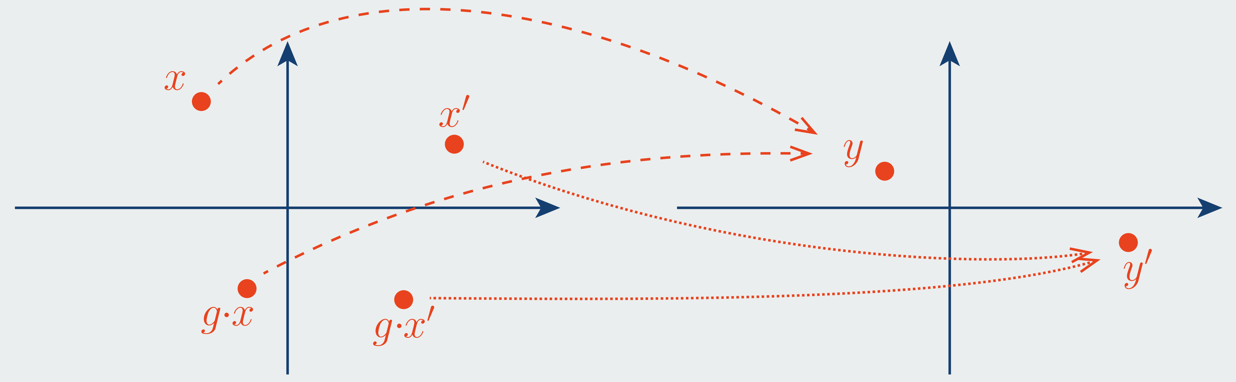
Benjamin Aslan[1], Daniel Platt[2], David Sheard[1]

[1]University College London [2]Imperial College London

Problem statement

Given:
Input/output data (x_i, y_i), and a group G acting on the input space
Objective:
Find an approximation f for the data that is *invariant* under the group action, i.e.

$$f(g \cdot x) = f(x) \text{ for all } g \in G$$



Question:
How can we use machine learning to train a model which has this invariance?

Classical approaches

1. **Data augmentation [1]:** for each pair (x, y) and each $g \in G$, add the pair $(g \cdot x, y)$ to the data pairs; then search for the best approximation f ignoring all invariance $\to$ the result will be nearly invariant under the group action
2. **Restrict architecture [2]:** for a neural network, being invariant under the group action is equivalent to relations between the learnable parameters; impose this restriction while searching for the best approximation
3. **Pooling [3]:** any map $h : \mathbb{R}^n \to \mathbb{R}^k$ can be made invariant under the group action by pooling:

$$h_{\text{pool}}(x) = \sum_{g \in G} h(g \cdot x) \text{ for } x \in \mathbb{R}^n$$

Can apply other pooling functions, and find the best approximation over all functions h and all pooling functions at the same time

References

[1] Shuxiao Chen, Edgar Dobriban, and Jane Lee. A group-theoretic framework for data augmentation. *Advances in Neural Information Processing Systems*, 33, 2020.
[2] Siamak Ravanbakhsh, Jeff Schneider, and Barnabás Póczos. Equivariance through parameter-sharing. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2892–2901, 2017.
[3] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.
[4] Yang-Hui He. Machine-learning the string landscape. *Physics Letters B*, 774:564–568, November 2017.
[5] Kieran Bull, Yang-Hui He, Vishnu Jejjala, and Challenger Mishra. Getting cicy high. *Physics Letters B*, 795:700 – 706, 2019.

Near-isometries

- Map H is an *isometry* if it preserves distances, i.e.
$$\operatorname{dist}(u, v) = \operatorname{dist}(H(u), H(v)) \text{ for all } u, v$$
- The quotient space $\mathbb{R}^n/G$ is a metric space (under very mild assumptions on the group action)
- **Idea:** choose an isometry $H : \mathbb{R}^n/G \to \mathbb{R}^m$ and approximate the data pairs
$$(H(x_i), y_i)_{i \in I}$$
by a function f
  - use any machine learning model for this approximation
  - apply the function $f \circ H$ to new, unseen data points
- Exact isometries are hard to find, but always have one map that is nearly an isometry: *projection onto a fundamental domain (PFD)*.
  A fundamental domain is a simply-connected set $U \subset \mathbb{R}^n$ which intersects every G-orbit exactly once $\to$ the map $x \mapsto (G \cdot x) \cap U$ is well-defined and nearly an isometry

Experiment 1: Hodge numbers of Calabi-Yau manifolds

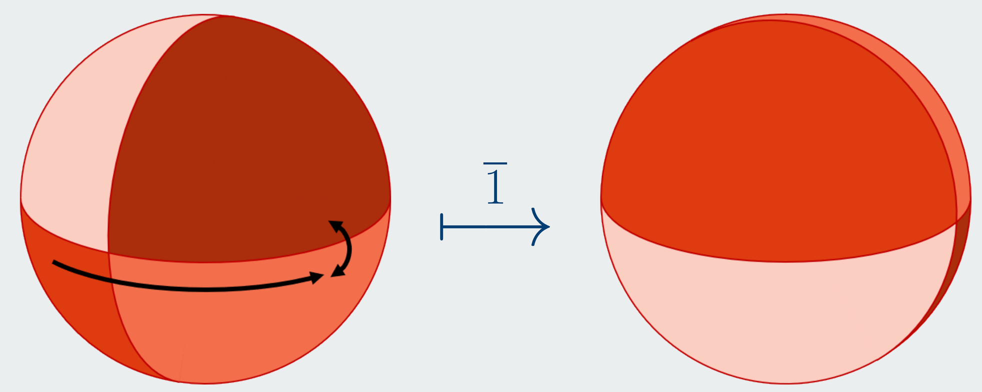
Complete Intersection Calabi-Yau manifolds are geometric objects that can be encoded by a matrix $A \in \mathbb{R}^{12 \times 15}$. If rows/columns are permuted, the matrix encodes *the same object*. Each manifold naturally has a Hodge number $h^{1,1}$ that is expensive to compute.

Many approaches to learn the map $A \mapsto h^{1,1}$, never get a map that is invariant under row/column permutations [4, 5]. Here, $G = S_{12} \times S_{15}$ has more than 10^{20} elements!

We train on randomly permuted matrices comparing three techniques. We use *PFD* as a near-isometry:

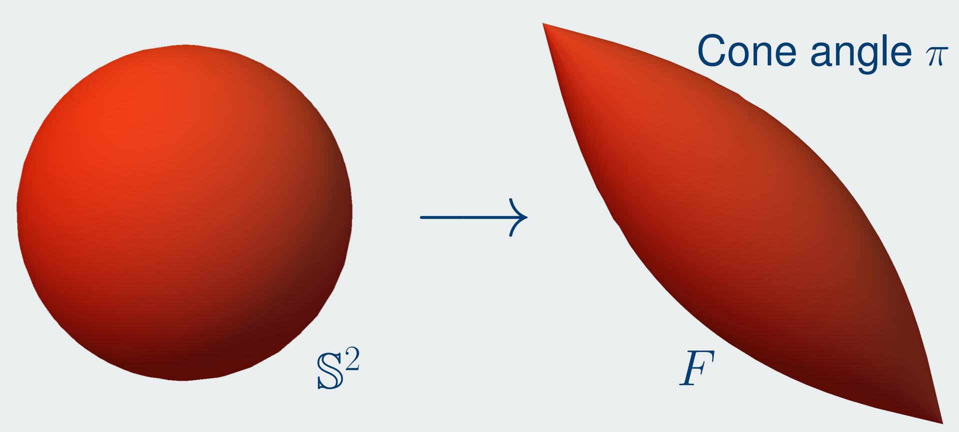
	Test accuracy
without any preprocessing	0.18
with data augmentation	0.27
with near-isometry	**0.62**

$\mathbb{Z}_4$ acts on $\mathbb{R}^{2 \times 2}$ by cyclically permuting the coordinates, fixing $\mathbf{v} = (1, 1, 1, 1)$, and acting orthogonally on its complement.



The action on $\mathbb{S}^2 \subset \mathbf{v}^\perp \subset \mathbb{R}^{2\times 2}$ is generated by a glide reflection through angle $\pi/2$.

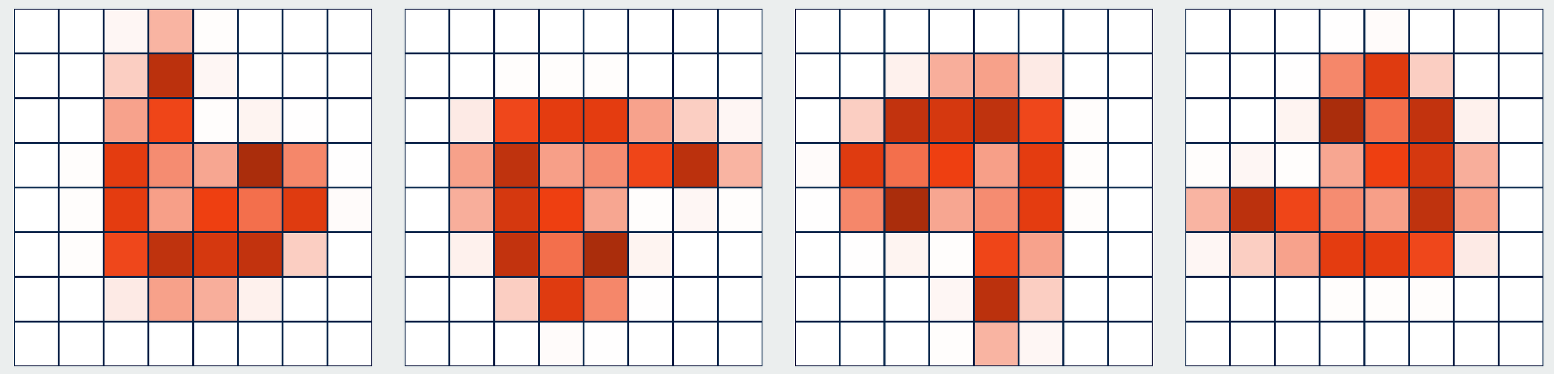
To embed $\mathbb{R}^{2\times 2}/\mathbb{Z}_4$ in $\mathbb{R}^m$: find $\mathbb{S}^2/\mathbb{Z}_4 \hookrightarrow \mathbb{S}^{m-2}$, then take the cone $\mathbf{v}^\perp/\mathbb{Z}_4 \hookrightarrow \mathbb{R}^{m-1}$, then cross with $\mathbb{R}$ to get $\mathbb{R}^{2\times 2}/\mathbb{Z}_4 \hookrightarrow \mathbb{R}^m$.



First we embed $F = \mathbb{S}^2/\langle R \rangle \hookrightarrow \mathbb{R}^3$, where R is a rotation by π.

Experiment 2: rotated handwritten digit recognition

The group $G = \mathbb{Z}_4$ acts on 8×8 pixel images by rotations (see the four rotations of an image showing the digit "6" below)

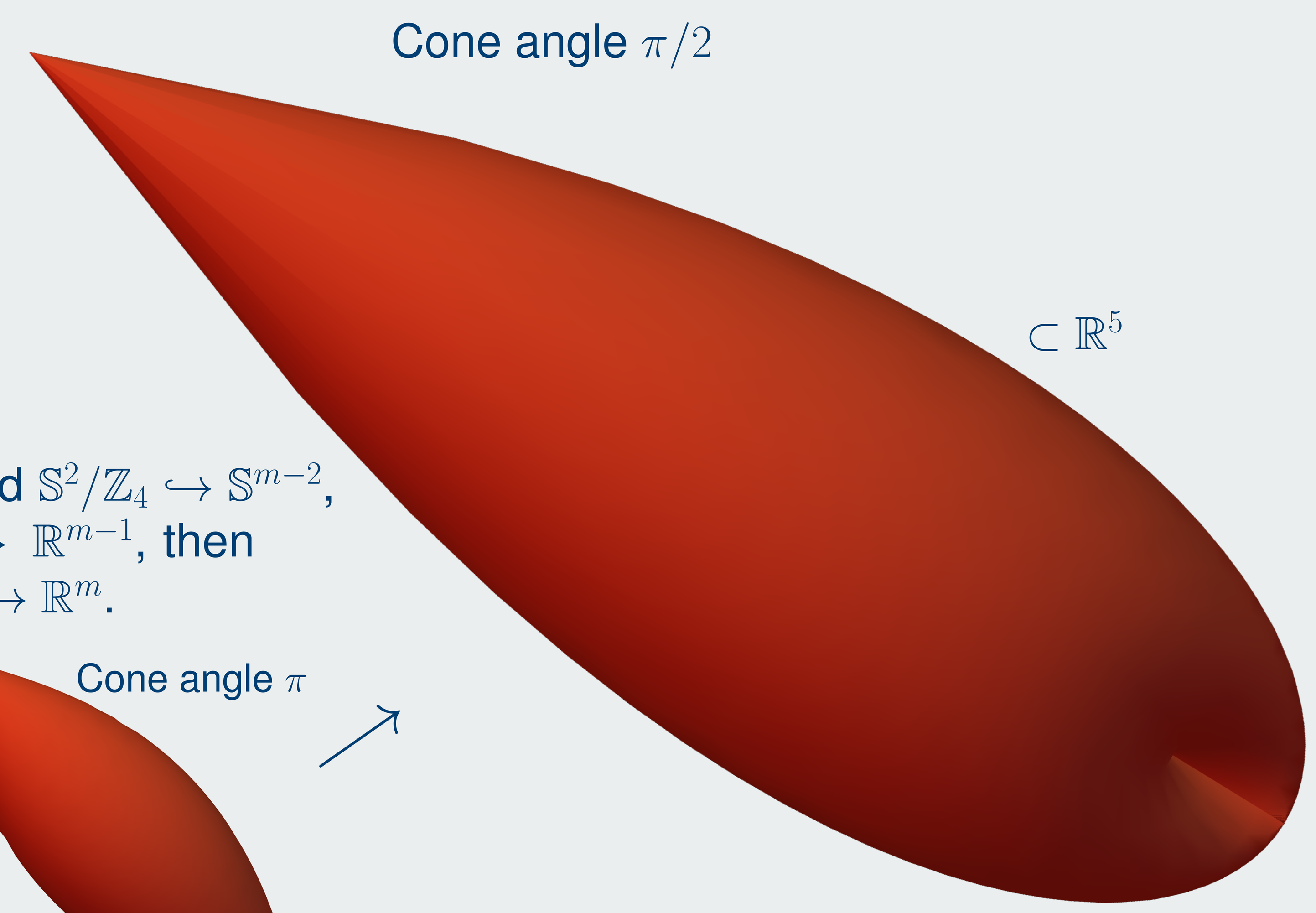


We train digit recognition with two near-isometries:
- $H_1 : \mathbb{R}^{8\times 8}/\mathbb{Z}_4 \to \mathbb{R}^{8\times 8}$ rotate image so that the top-left quadrant has greatest total brightness (*PFD*)
- $H_2 : \mathbb{R}^{8\times 8}/\mathbb{Z}_4 \to \mathbb{R}^{2080}$ obtained by observing that $\mathbb{R}^{8\times 8}/\mathbb{Z}_4 = \mathbb{R} \times \text{Cone}(V)$ for a singular space V that can be embedded into Euclidean space using the Veronese embedding

Train neural networks on pairs (1) (x, y), (2) $(H_1(x), y)$), and (3) $(H_2(x), y)$, where x is a randomly rotated 8×8 image and y is the digit in the image (between 0 and 9):

	Test accuracy
(x, y)	0.87
$(H_1(x), y)$	0.92
$(\mathbf{H_2(x), y})$	**0.94**

The construction of the map H_2 for the case of 2×2 images is shown below, the higher-dimensional analogue of this was used for our experiment.



Next we embed $F/\langle A \rangle$ in $\mathbb{R}^5$ using the Veronese embedding, where A is the antipodal map. Overall the image is an embedding of $V = \mathbb{S}^2/\mathbb{Z}_4$ in $\mathbb{R}^5$.

*This work was supported by the Engineering and Physical Sciences Research Council [EP/L015234/1]. The EPSRC Centre for Doctoral Training in Geometry and Number Theory (The London School of Geometry and Number Theory), University College London