On Learning Sets of Symmetric Elements

Haggai Maron[^1], Or Litany[^2], Gal Chechik[^1,3], Ethan Fetaya[^3]

Motivation and Overview
Set Symmetry

Previous work (DeepSets, PointNet) targeted training a deep network over sets

\[
\left\{ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \ldots \right\} \rightarrow \text{Deep Net}
\]
Set+Elements symmetry

Both the set and its elements have symmetries.

\[
\{ \text{Input set} \} \rightarrow \text{Deep Net}
\]

**Main challenge:** What architecture is optimal when elements of the set have their own symmetries?
Deep Symmetric sets

Input image set → Output
Set symmetry:
Order invariance/equivariance
Set symmetry:
Order invariance/equivariance
Set symmetry:
Order invariance/equivariance
Element symmetry:
Translation invariance/equivariance
Element symmetry:
Translation invariance/equivariance
Element symmetry: Translation invariance/equivariance
Applications

1D signals | 2D images | 3D pointclouds | Graph
This paper

A principled approach for learning sets of complex elements (graphs, point clouds, images)

Characterize maximally expressive linear layers that respect the symmetries (DSS layers)

Prove universality results

Experimentally demonstrate that DSS networks outperform baselines
Previous work
Deep sets [Zaheer et al. 2017]
Deep sets [Zaheer et al. 2017]

Siamese

CNN

CNN

CNN
Deep sets [Zaheer et al. 2017]
Deep sets [Zaheer et al.]
Previous work: information sharing

Aittala and Durand, ECCV 2018
Sridhar et al., NeurIPS 2019
Liu et al., ICCV 2019
Our approach
Invariance

Many Learning tasks are invariant to natural transformations (symmetries)

More formally. Let $H \leq S_n$ be a subgroup:

$f : \mathbb{R}^n \to \mathbb{R}$ is **invariant** if $f(\tau \cdot x) = f(x)$, for all $\tau \in H$

e.g. image classification
Equivariance

Let $H \leq S_n$ be a subgroup:

**Equivariant** if $f(\tau \cdot x) = \tau \cdot f(x)$,

e.g. edge detection
Invariant neural networks

• Invariant by construction
Deep Symmetric Sets

$x_1, \ldots, x_n \in \mathbb{R}^d$ with symmetry group $G \le S_d$

Want to be invariant/equivariant to both $G$ and the ordering

Formally the symmetry group is $H = S_n \times G \le S_{nd}$
Main challenges

- What is the space of linear equivariant layers for specific $H = S_N \times G$?
Main challenges

- What is the space of linear equivariant layers for a given $H = S_N \times G$?

- Can we compute these operators efficiently?
Main challenges

- What is the space of linear equivariant layers for a given $H = S_N \times G$?
- Can we compute these operators efficiently?
- Do we lose expressive power?
Main challenges

- What is the space of linear equivariant layers for a given $H = S_N \times G$?

- Can we compute these operators efficiently?

- Do we lose expressive power?
Theorem: Any linear $S_N \times G$-equivariant layer $L : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$ is of the form

$$L(X)_i = L_1^G(x_i) + \sum_{j \neq i} L_2^G(x_j)$$

where $L_1^G, L_2^G$ are linear $G$-equivariant functions.

We call these layers *Deep Sets for Symmetric elements layers* (DSS)
DSS for images

\( x_1, \ldots, x_n \) are images

\( G \) is the group of 2D circular translations

\( G \)-equivariant layers are convolutions
DSS for images

\(x_1, \ldots, x_n\) are images

\(G\) is the group of 2D circular translations

\(G\)-equivariant layers are convolutions
DSS for images

$x_1, \ldots, x_n$ are images

$G$ is the group of $2D$ circular translations

$G$-equivariant layers are convolutions
DSS for images

$x_1, \ldots, x_n$ are images

$G$ is the group of 2D circular translations

$G$-equivariant layers are convolutions
DSS for images

Siamese part

Information sharing part

Single DSS layer
Expressive power

Theorem

If G-equivariant networks are universal approximators for G-equivariant functions, then so are DSS networks for $S_N \times G$-equivariant functions.
Expressive power

Theorem
If G-equivariant networks are universal approximators for G-equivariant functions, then so are DSS networks for $S_N \times G$-equivariant functions.

• Main tool:
  • Noether’s Theorem (Invariant theory)
  • For any finite group $H$, the ring of invariant polynomials $\mathbb{R}[x_1, \ldots, x_n]^H$ is finitely generated.
  • Generators can be used to create continuous unique encodings for elements in $\mathbb{R}^{n \times d}/H$
Results
Signal classification
### Image selection

<table>
<thead>
<tr>
<th>Noise type and strength</th>
<th>Late Aggregation Siamese+DS</th>
<th>Early Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSS (sum)</td>
<td>DSS (max)</td>
</tr>
<tr>
<td>Gaussian $\sigma = 10$</td>
<td>77.2% ± 0.37</td>
<td>78.48% ± 0.48</td>
</tr>
<tr>
<td>Gaussian $\sigma = 30$</td>
<td>65.89% ± 0.66</td>
<td>68.35% ± 0.55</td>
</tr>
<tr>
<td>Gaussian $\sigma = 50$</td>
<td>59.24% ± 0.51</td>
<td>62.6% ± 0.45</td>
</tr>
<tr>
<td>Occlusion 10%</td>
<td>82.15% ± 0.45</td>
<td>83.13%± 1.00</td>
</tr>
<tr>
<td>Occlusion 30%</td>
<td>77.47% ± 0.37</td>
<td>78% ± 0.89</td>
</tr>
<tr>
<td>Occlusion 50%</td>
<td>76.2% ± 0.82</td>
<td>77.29% ± 0.40</td>
</tr>
</tbody>
</table>
### Shape selection

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data type</th>
<th>Late Aggregation Siamese+DS</th>
<th>Early Aggregation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF101</td>
<td>Images</td>
<td>36.41% ± 1.43</td>
<td>76.6% ± 1.51</td>
</tr>
<tr>
<td>Dynamic Faust</td>
<td>Point-clouds</td>
<td>22.26% ± 0.64</td>
<td>28.71% ± 0.64</td>
</tr>
<tr>
<td>Dynamic Faust</td>
<td>Graphs</td>
<td>26.53% ± 1.99</td>
<td>44.24% ± 1.28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>DSS (sum)</th>
<th>DSS (max)</th>
<th>DSS (Sridhar)</th>
<th>DSS (Aittala)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76.39% ± 1.01</td>
<td>60.15% ± 0.76</td>
<td>54.26% ± 1.66</td>
<td>26.43% ± 3.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>42.45% ± 1.32</td>
<td>28.71% ± 0.64</td>
<td>30.54% ± 1.27</td>
<td>26.66% ± 4.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>44.24% ± 1.28</td>
<td>53.16% ± 1.47</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Point clouds**

**Graphs**
Conclusions

A general framework for learning sets of complex elements

Generalizes many previous works

Expressivity results

Works well in many tasks and data types