ICML 2020

On Learning Sets of Symmetric Elements

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

[1] Nvidia Research [2] Stanford University [3] Bar-Ilan University

Motivation and Overview

Set Symmetry

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

$$\left\{ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \dots \right\}$$

Input set

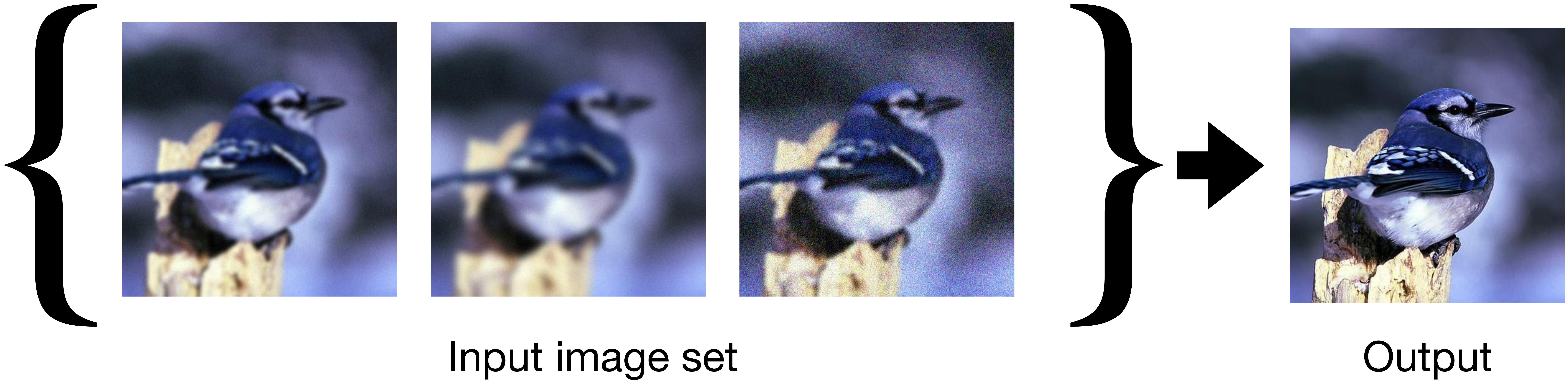
Set+Elements symmetry

Both the set and its elements have symmetries.

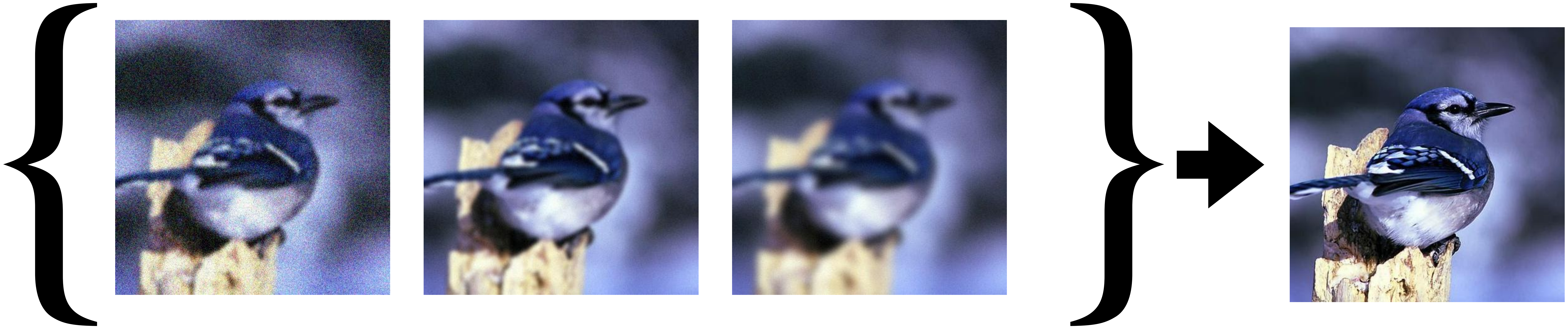
Input set

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

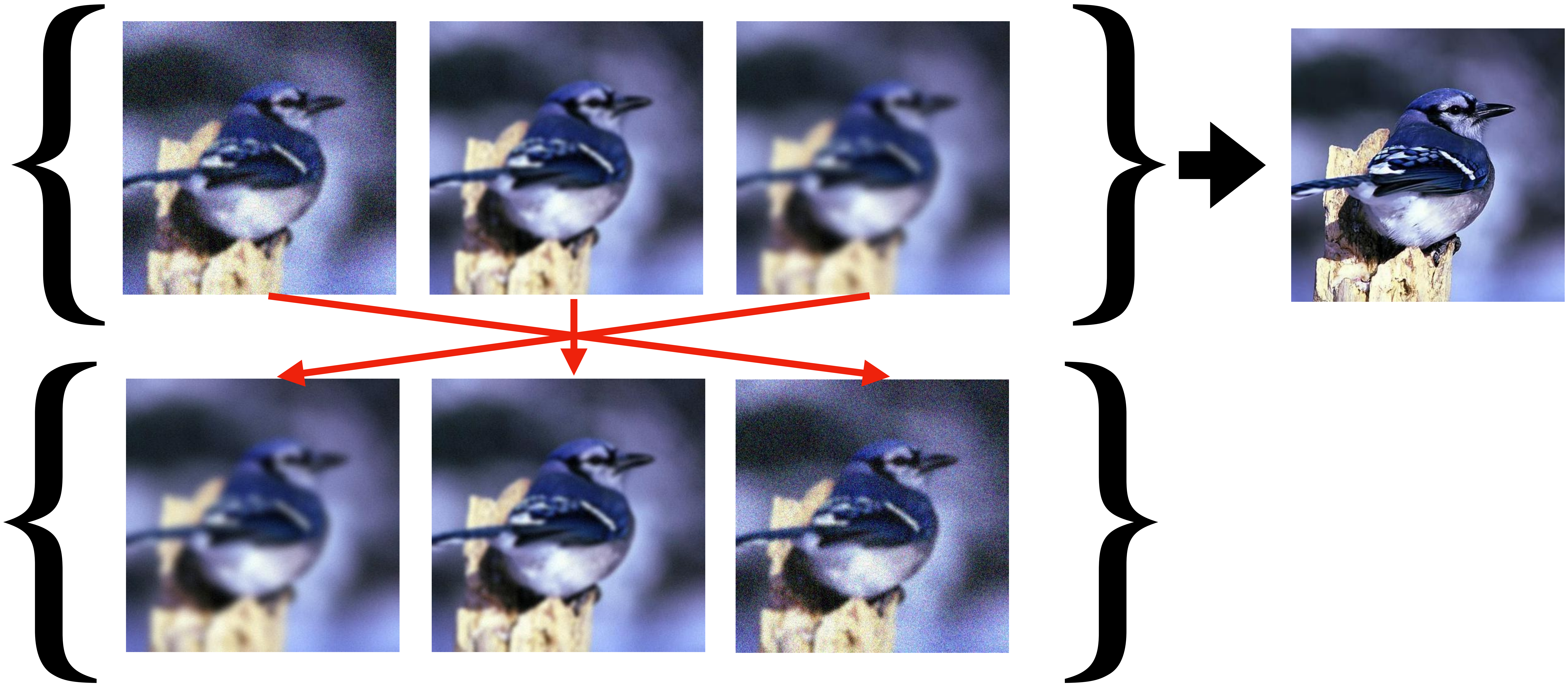
Deep Symmetric sets



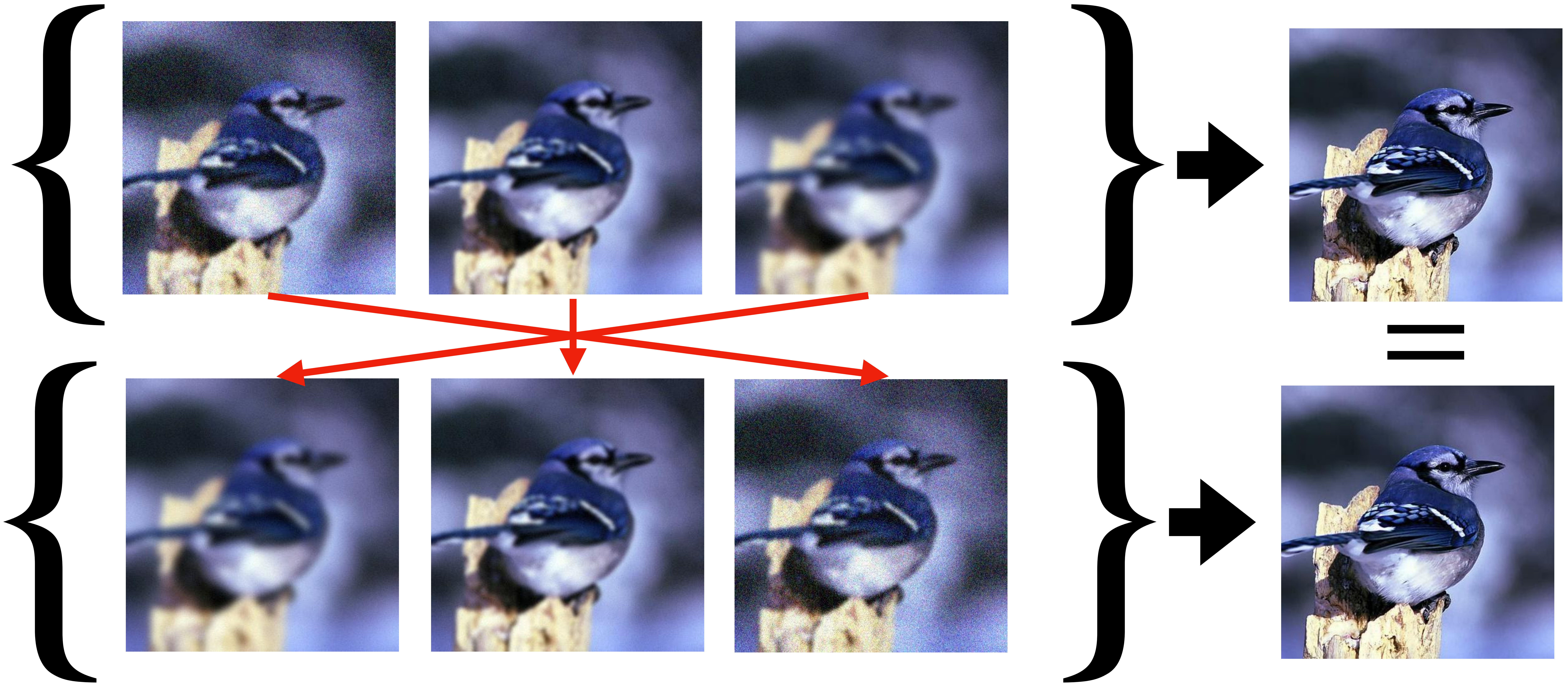
Set symmetry:
Order invariance/equivariance



Set symmetry:
Order invariance/equivariance

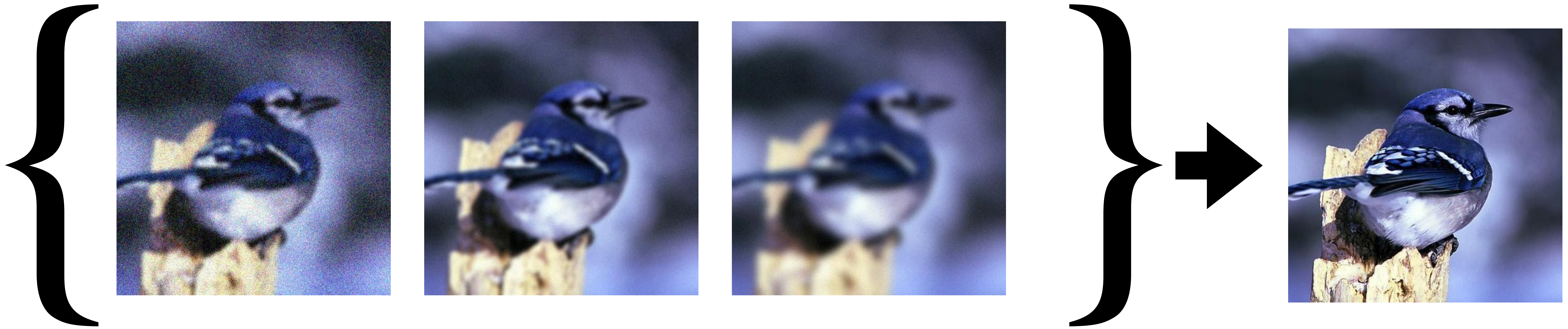


Set symmetry:
Order invariance/equivariance

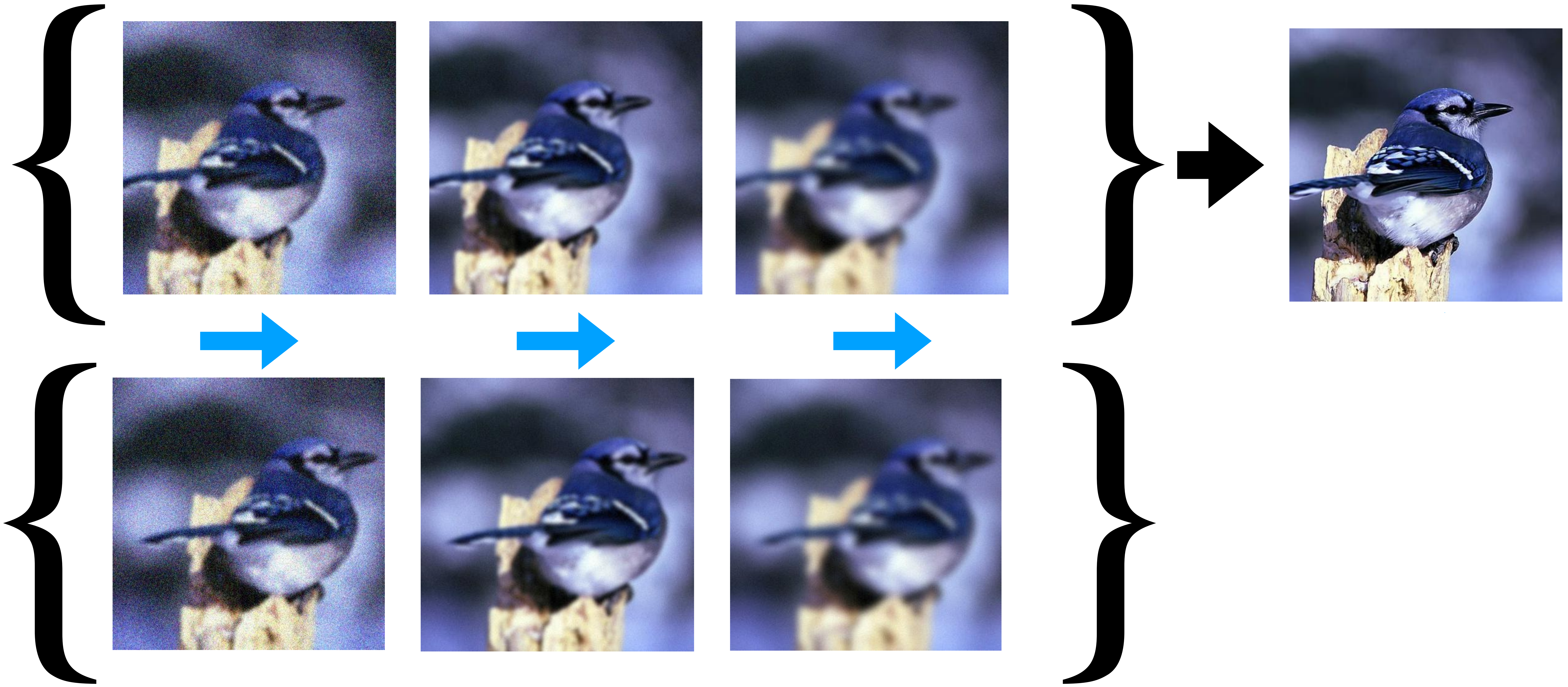


Element symmetry:

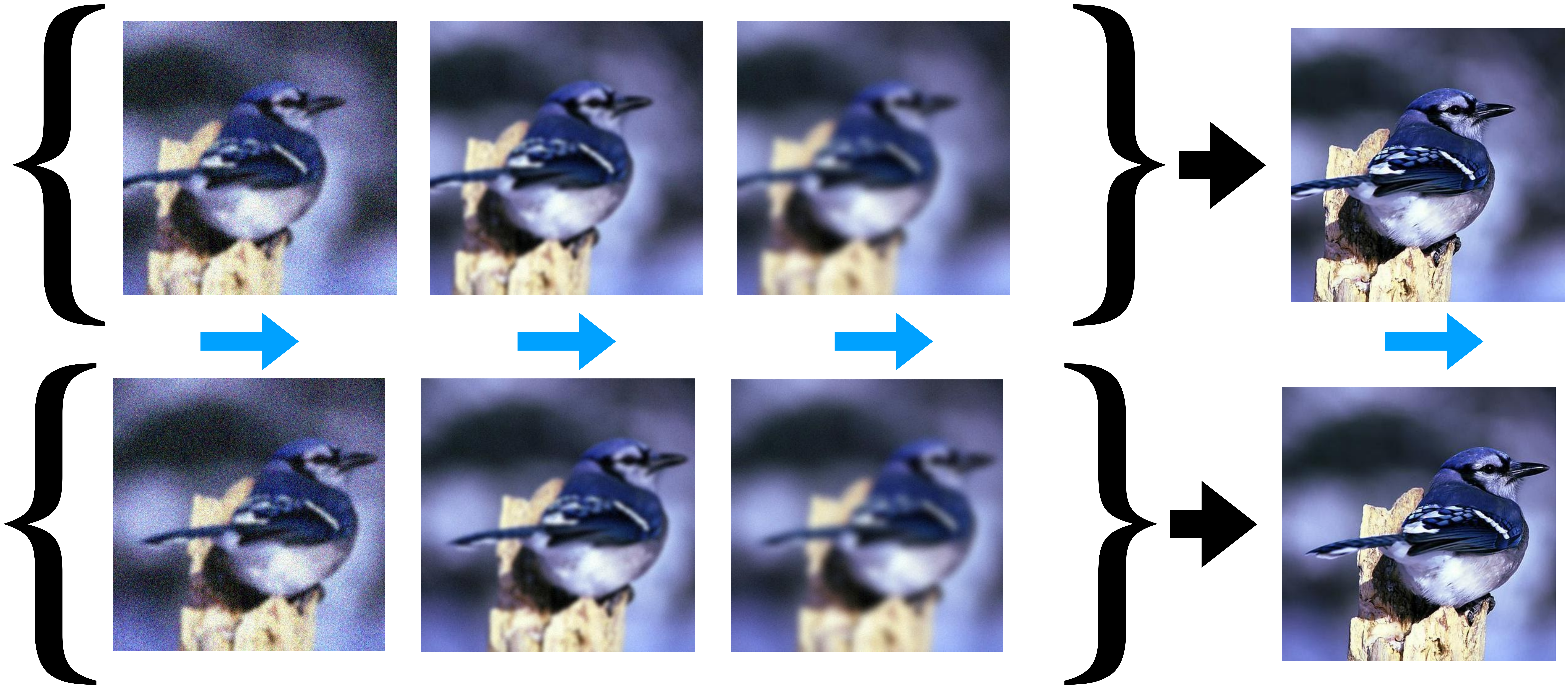
Translation invariance/equivariance



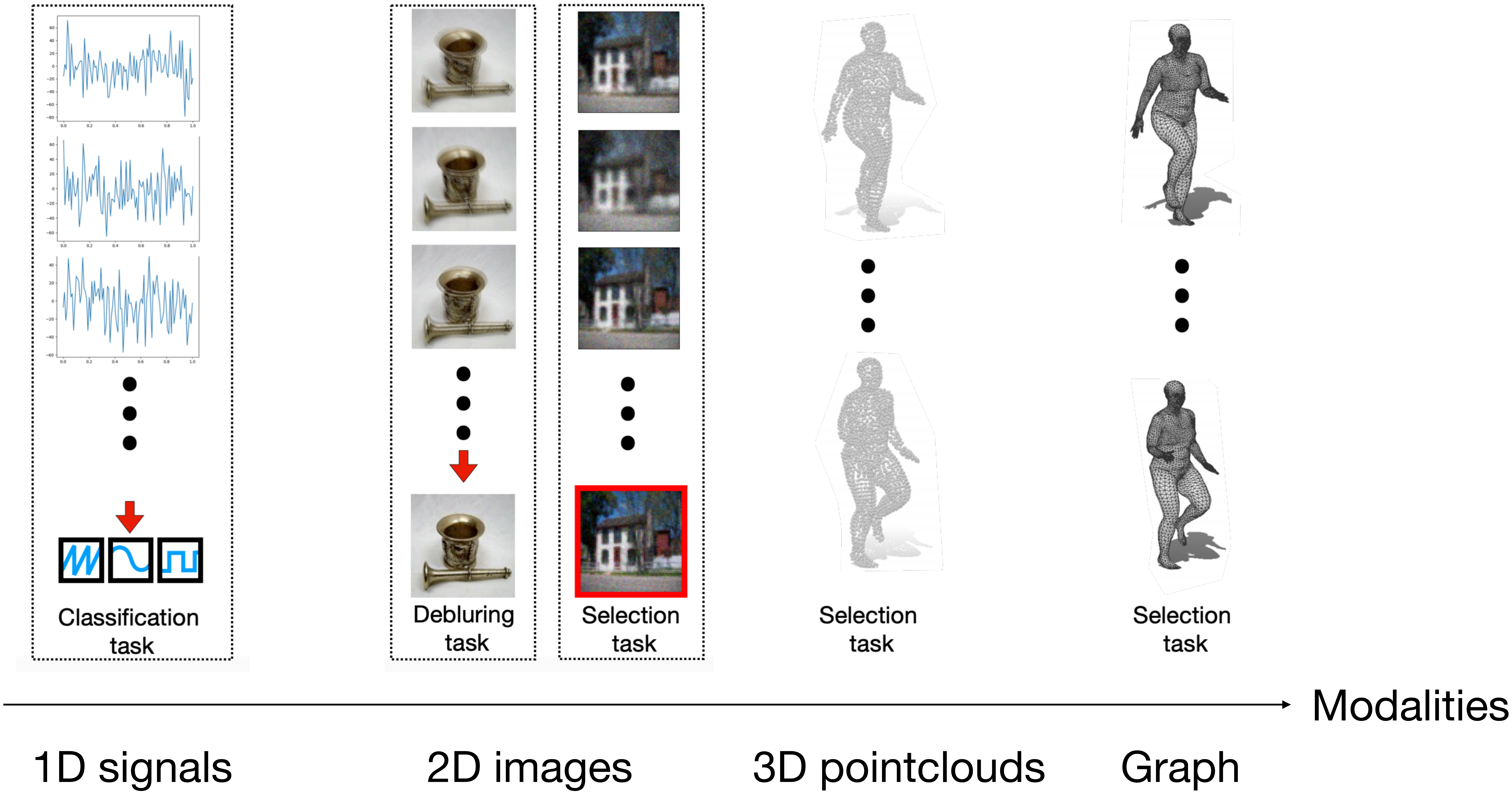
Element symmetry:
Translation invariance/equivariance



Element symmetry:
Translation invariance/equivariance



Applications



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**)

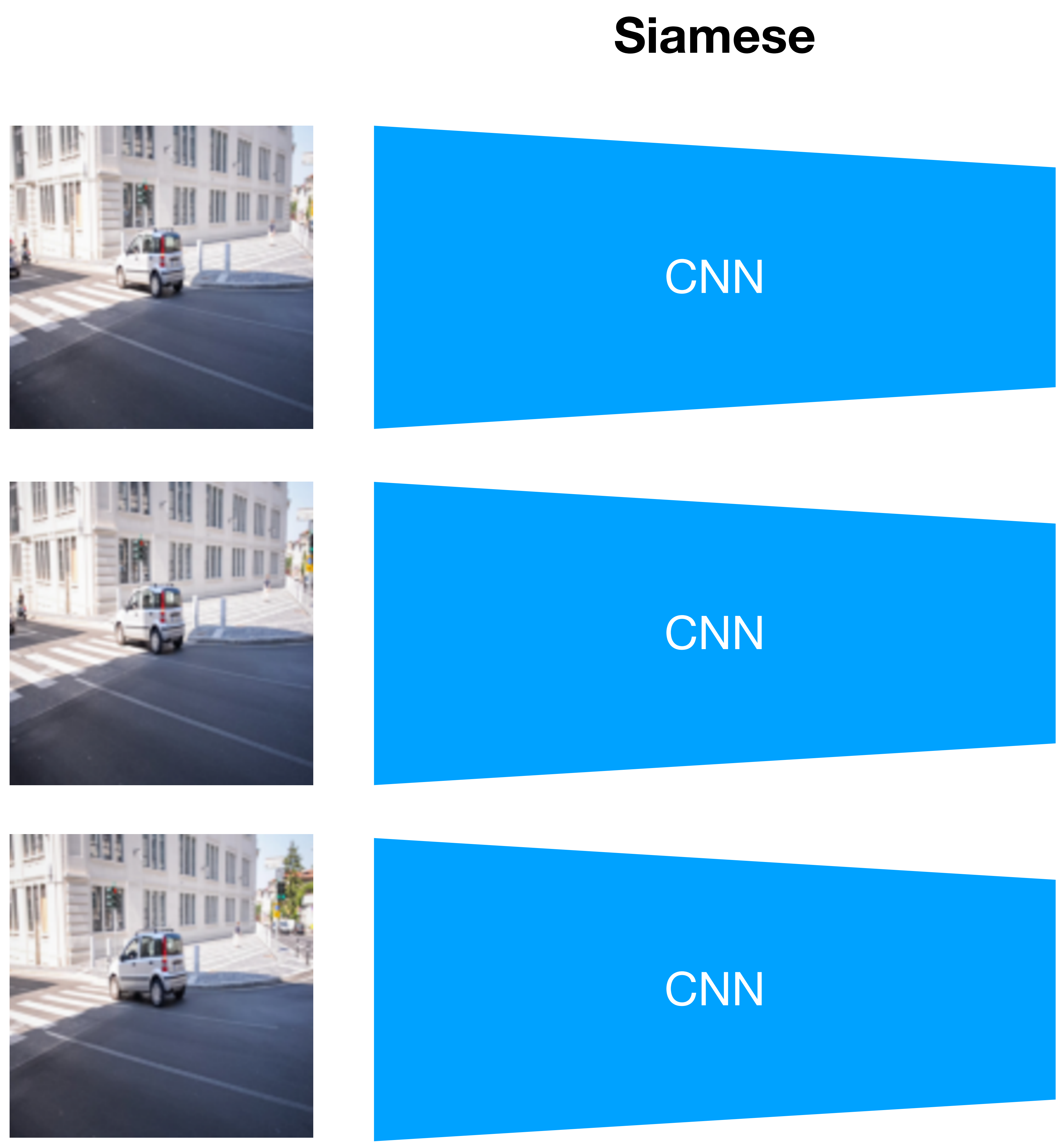
Prove universality results

Experimentally demonstrate that **DSS networks** outperform baselines

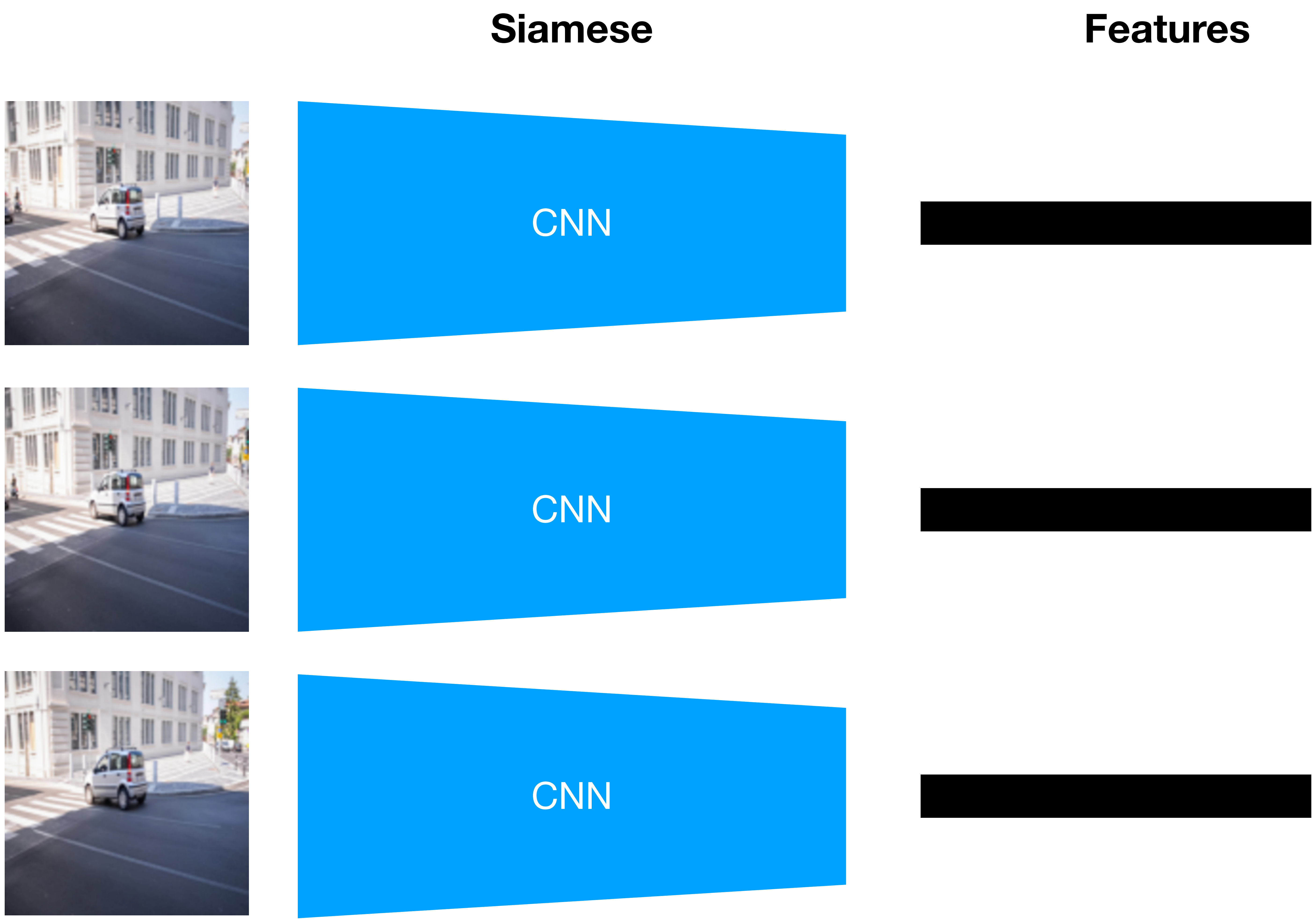
Previous work

Deep sets [Zaheer et al. 2017]

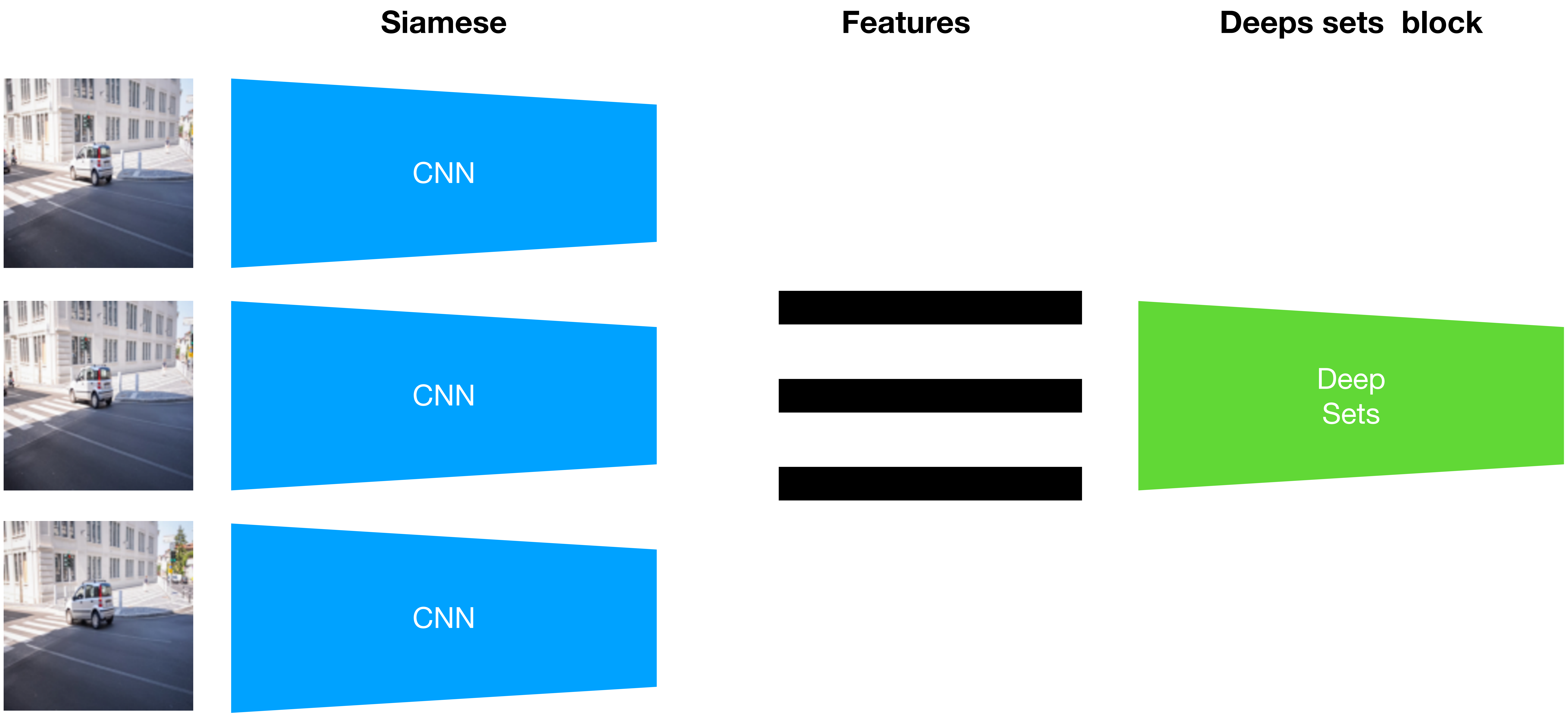
Deep sets [Zaheer et al. 2017]



Deep sets [Zaheer et al. 2017]



Deep sets [Zaheer et al.]

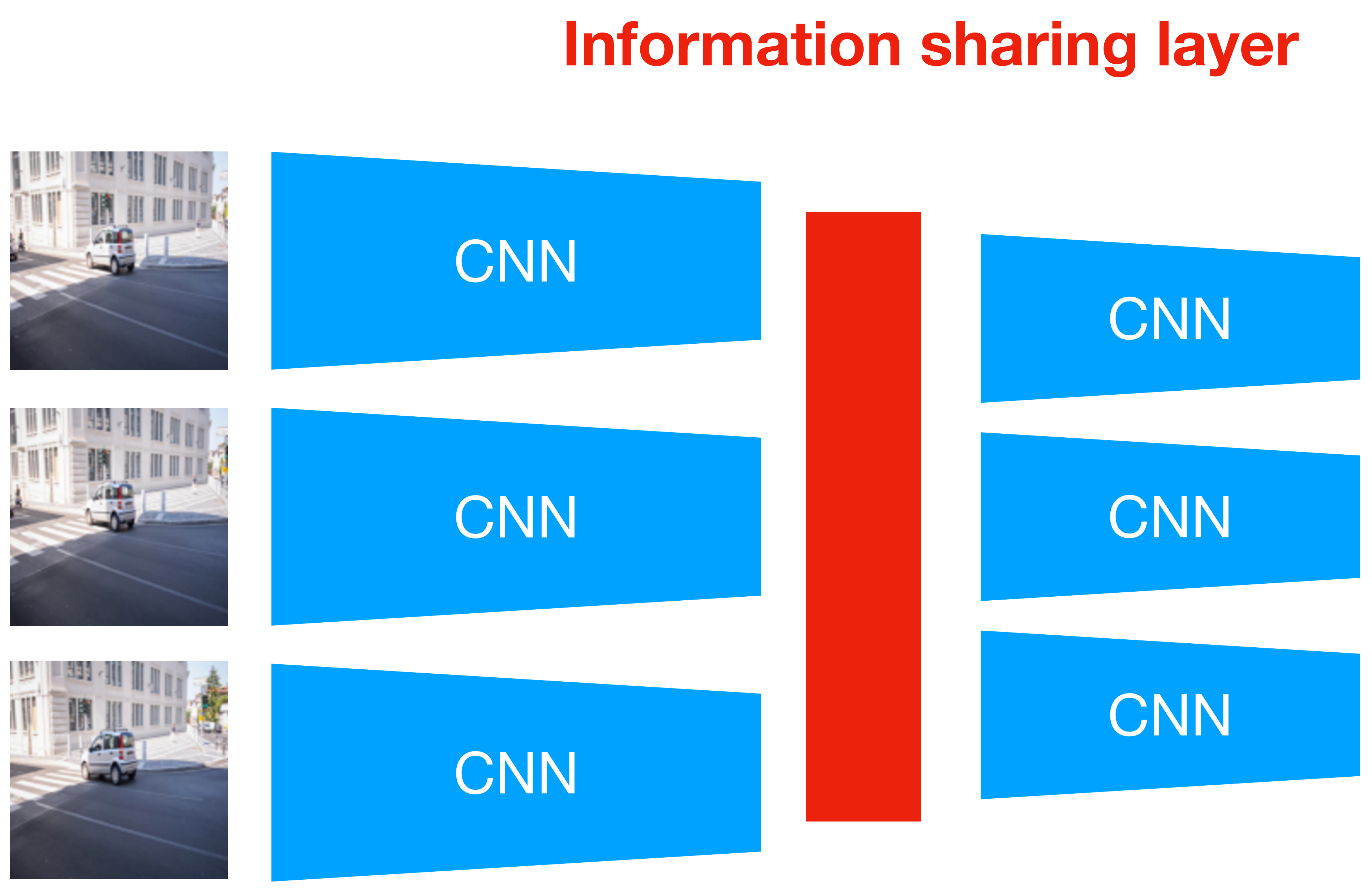


Previous work: information sharing

Aittala and Durand, ECCV 2018

Sridhar et al., NeurIPS 2019

Liu et al., ICCV 2019



Our approach

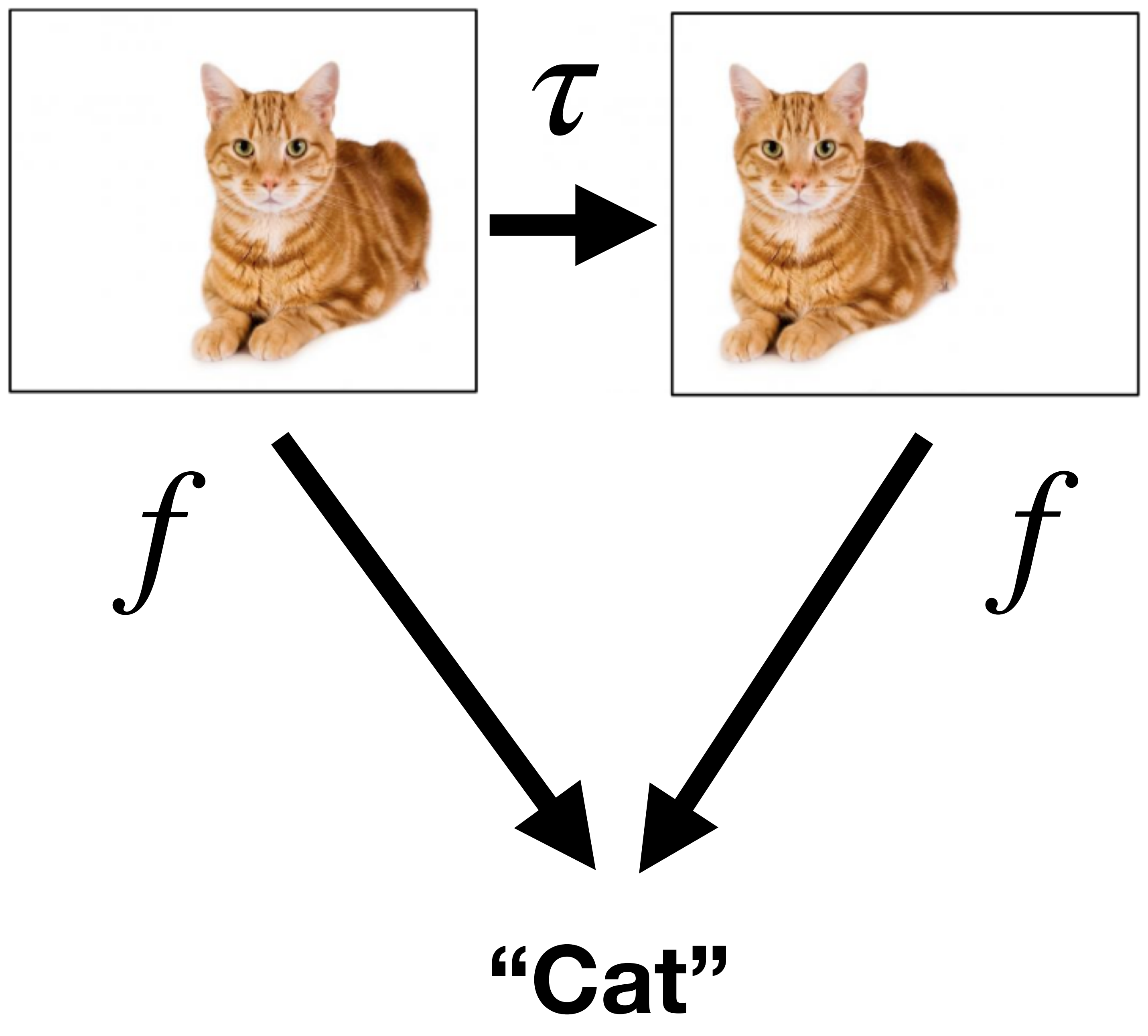
Invariance

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

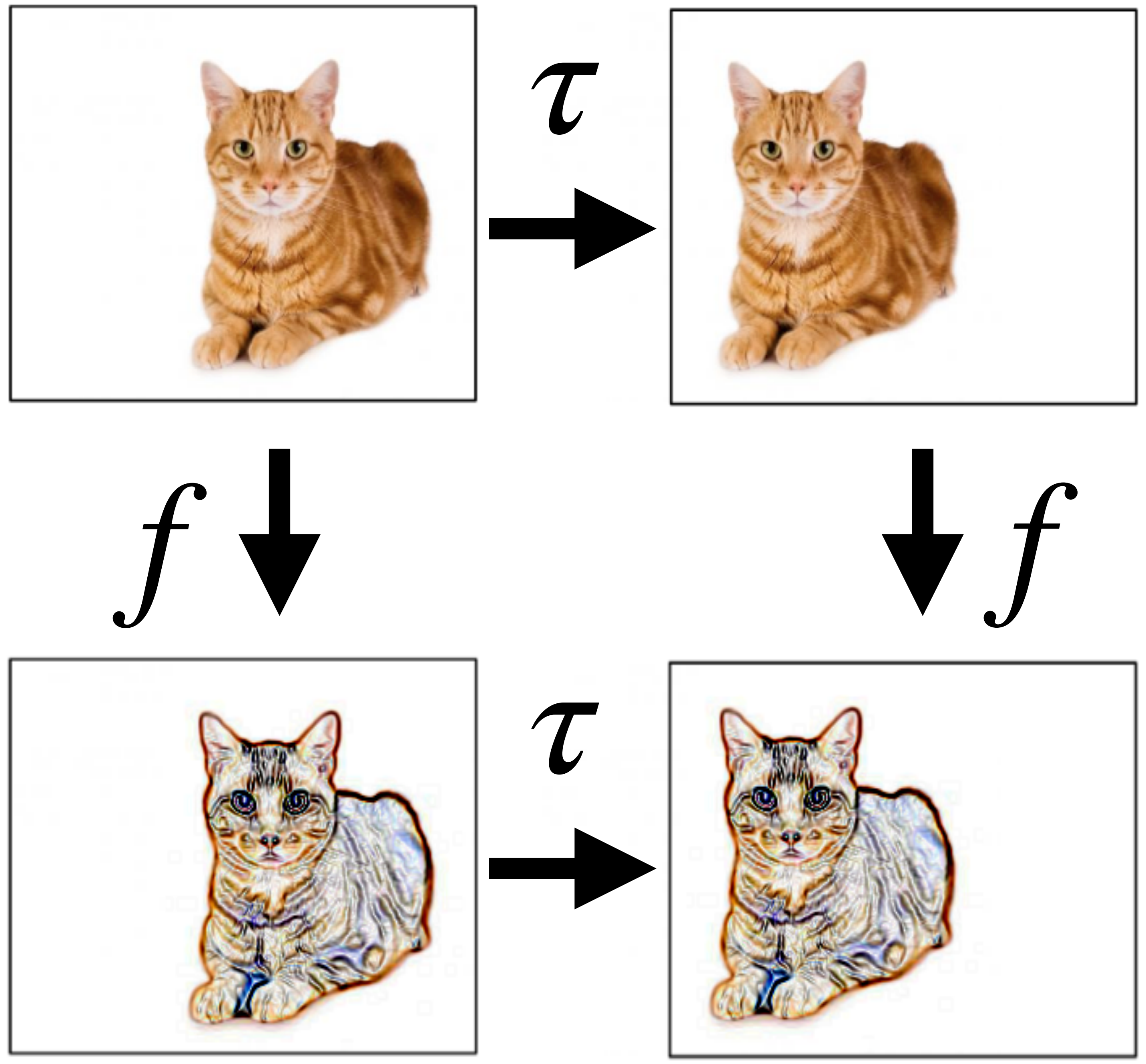


Equivariance

Let $H \leq S_n$ be a subgroup:

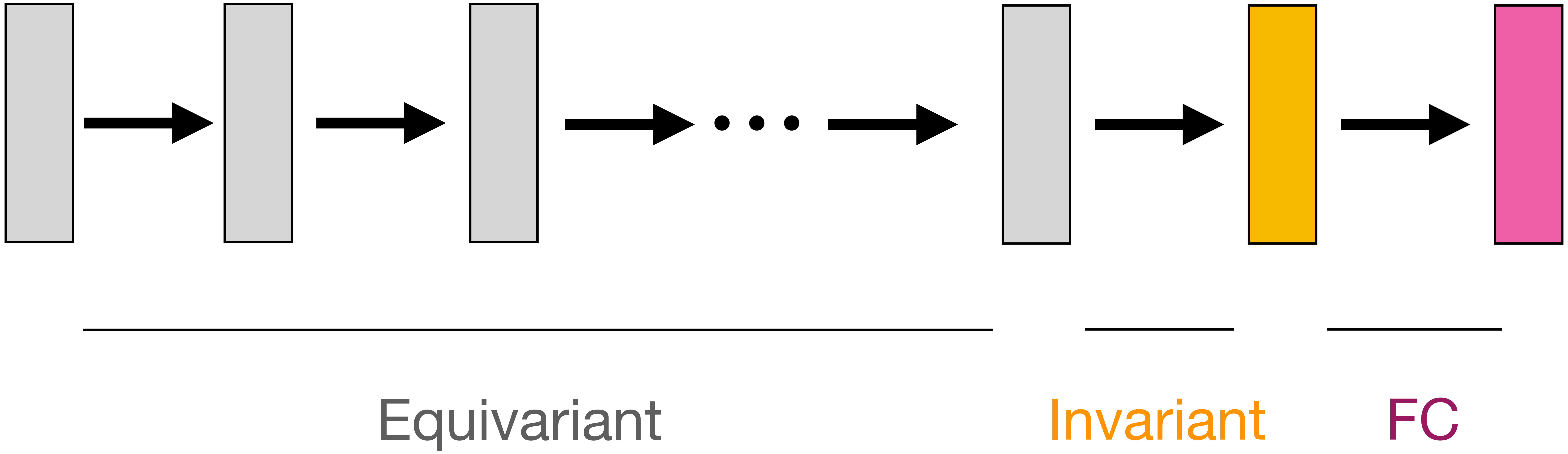
Equivariant if $f(\tau \cdot x) = \tau \cdot f(x)$,

e.g. edge detection



Invariant neural networks

- Invariant by construction

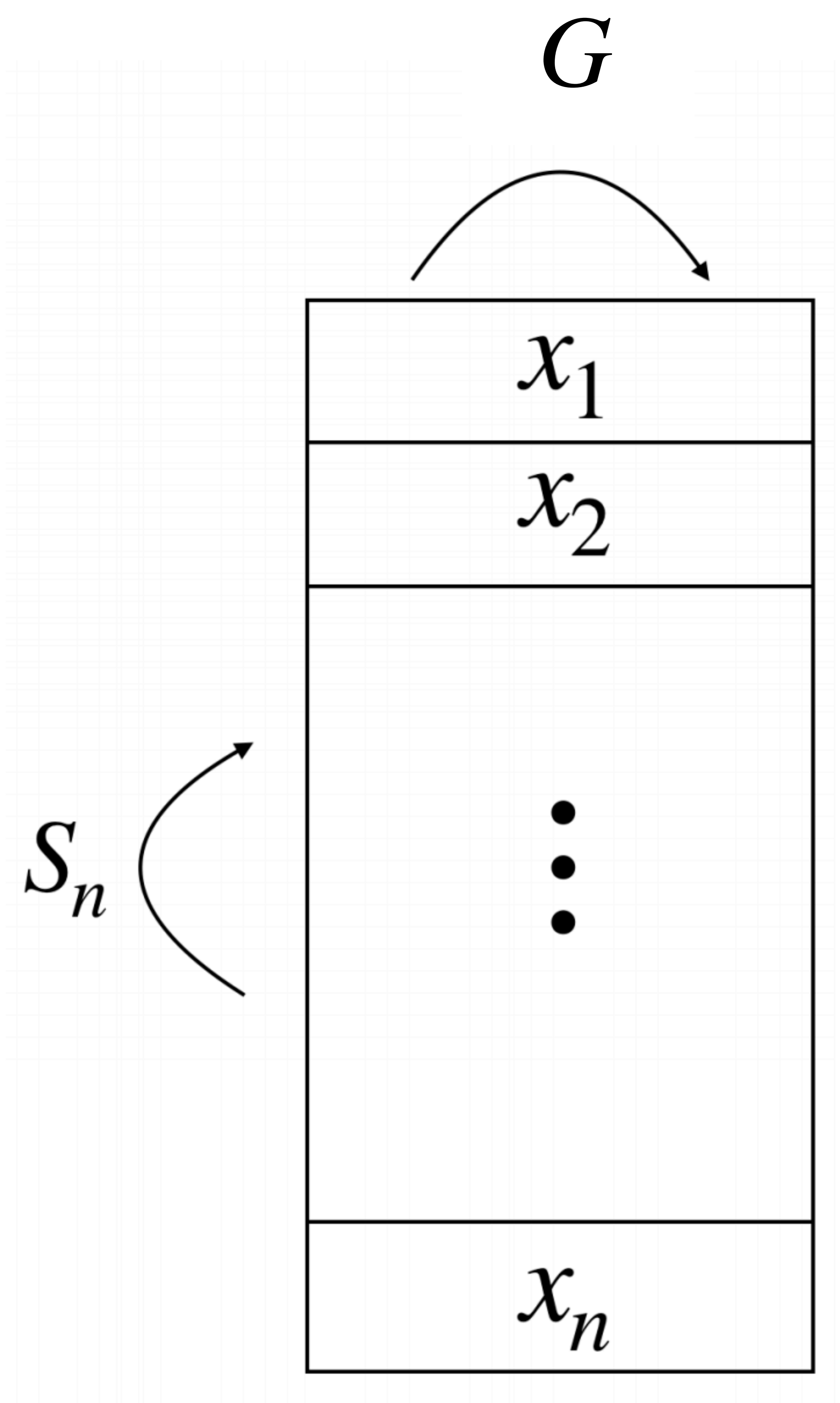


Deep Symmetric Sets

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

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

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



Main challenges

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

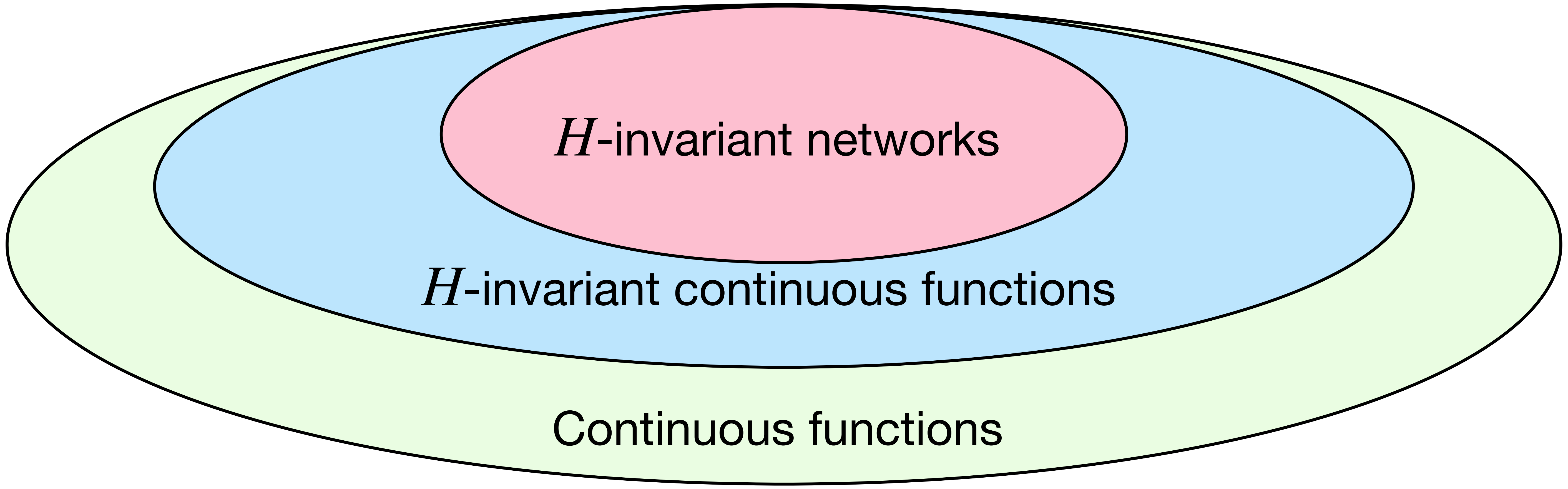
Main challenges

- 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?

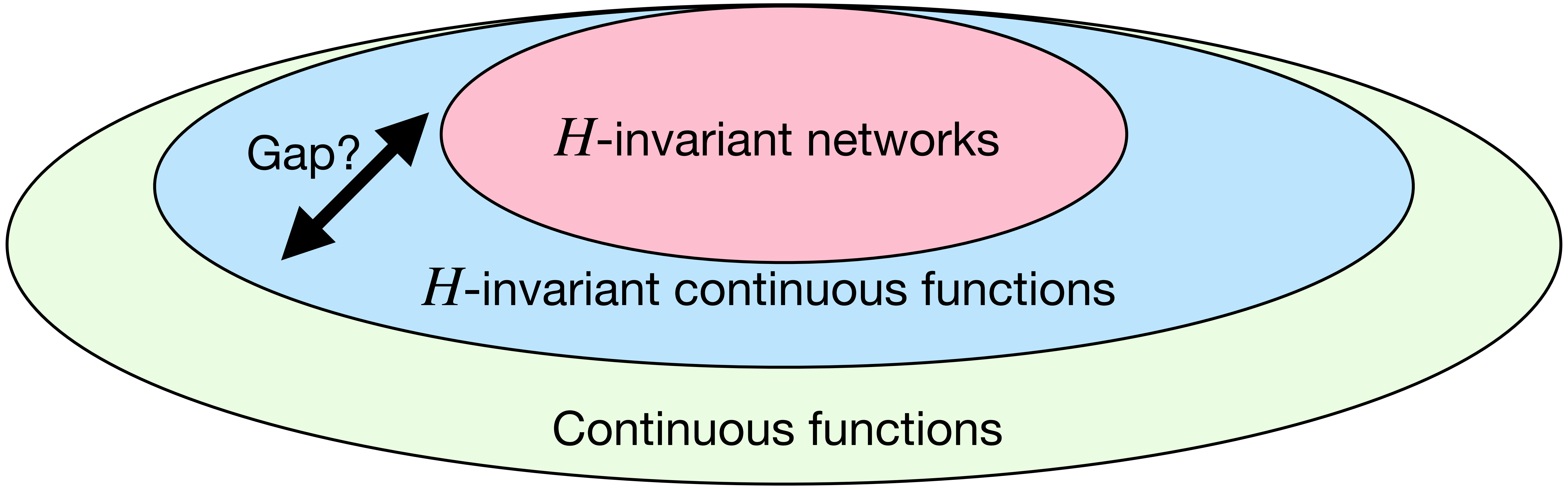


Main challenges

- 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?



H-equivariant layers

Theorem: Any linear $S_N \times G$–equivariant layer $L : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ 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

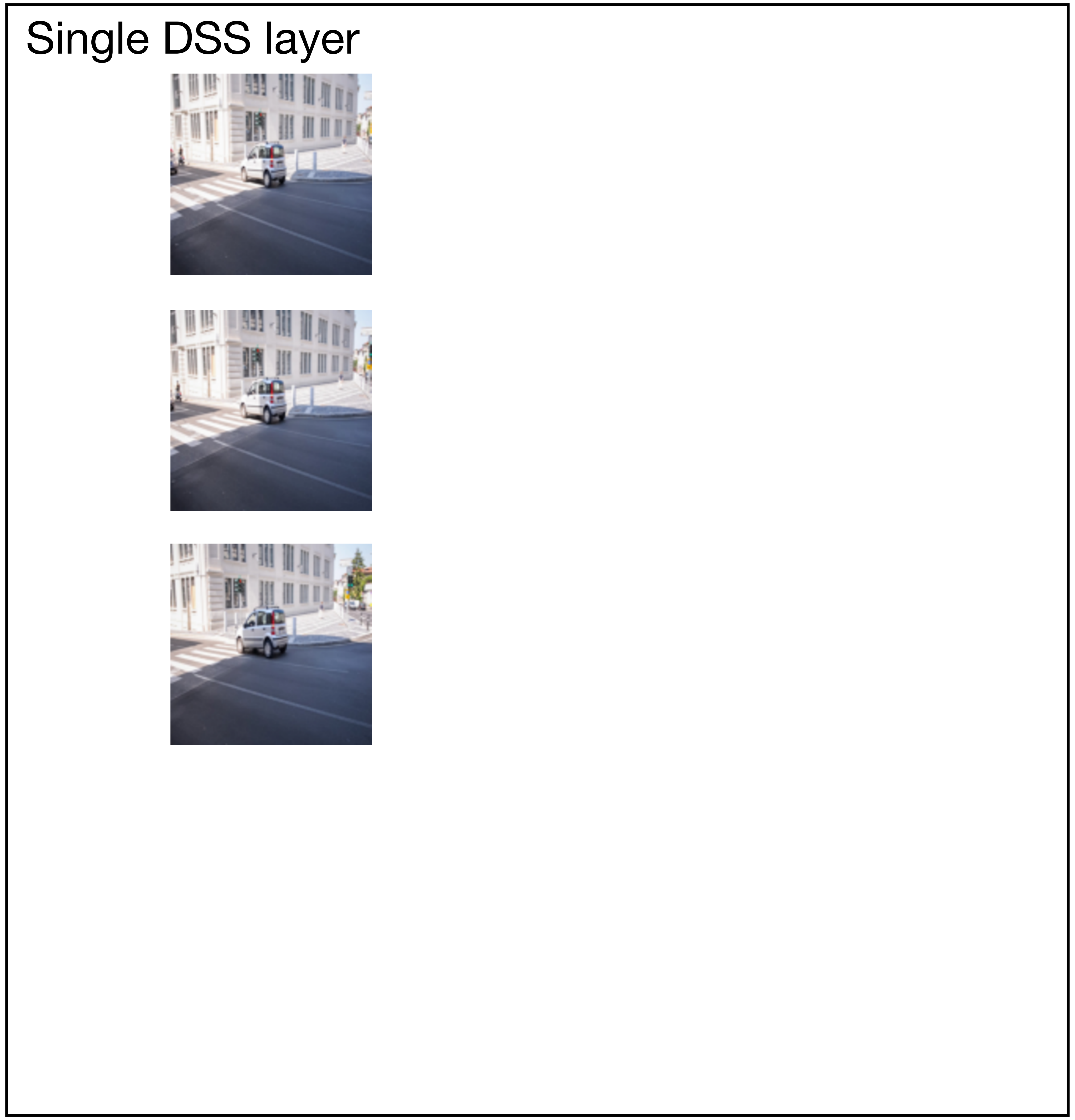
We call these layers ***Deep Sets for Symmetric elements layers*** (DSS)

DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

G-equivariant layers are convolutions

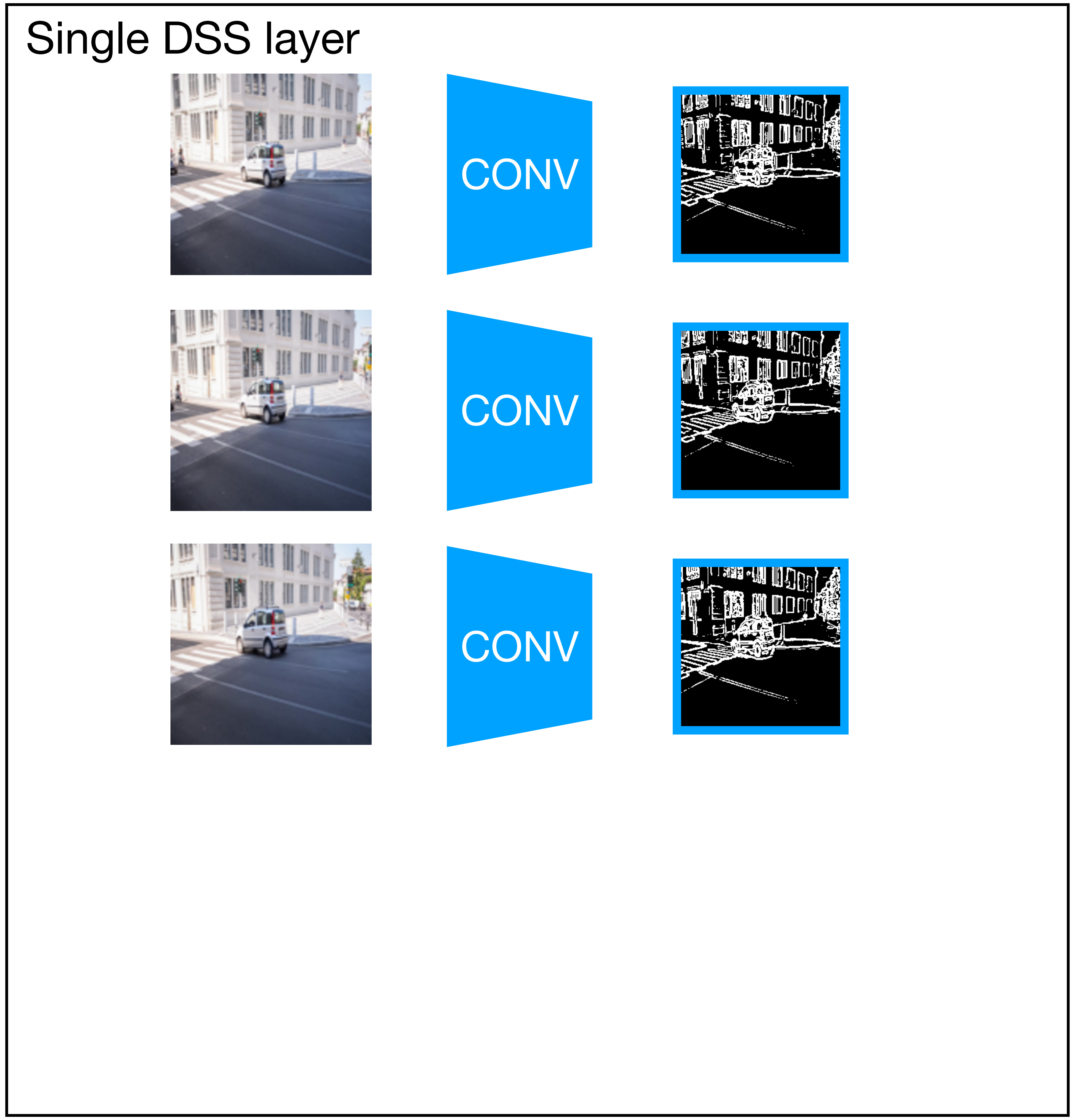


DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

G-equivariant layers are convolutions

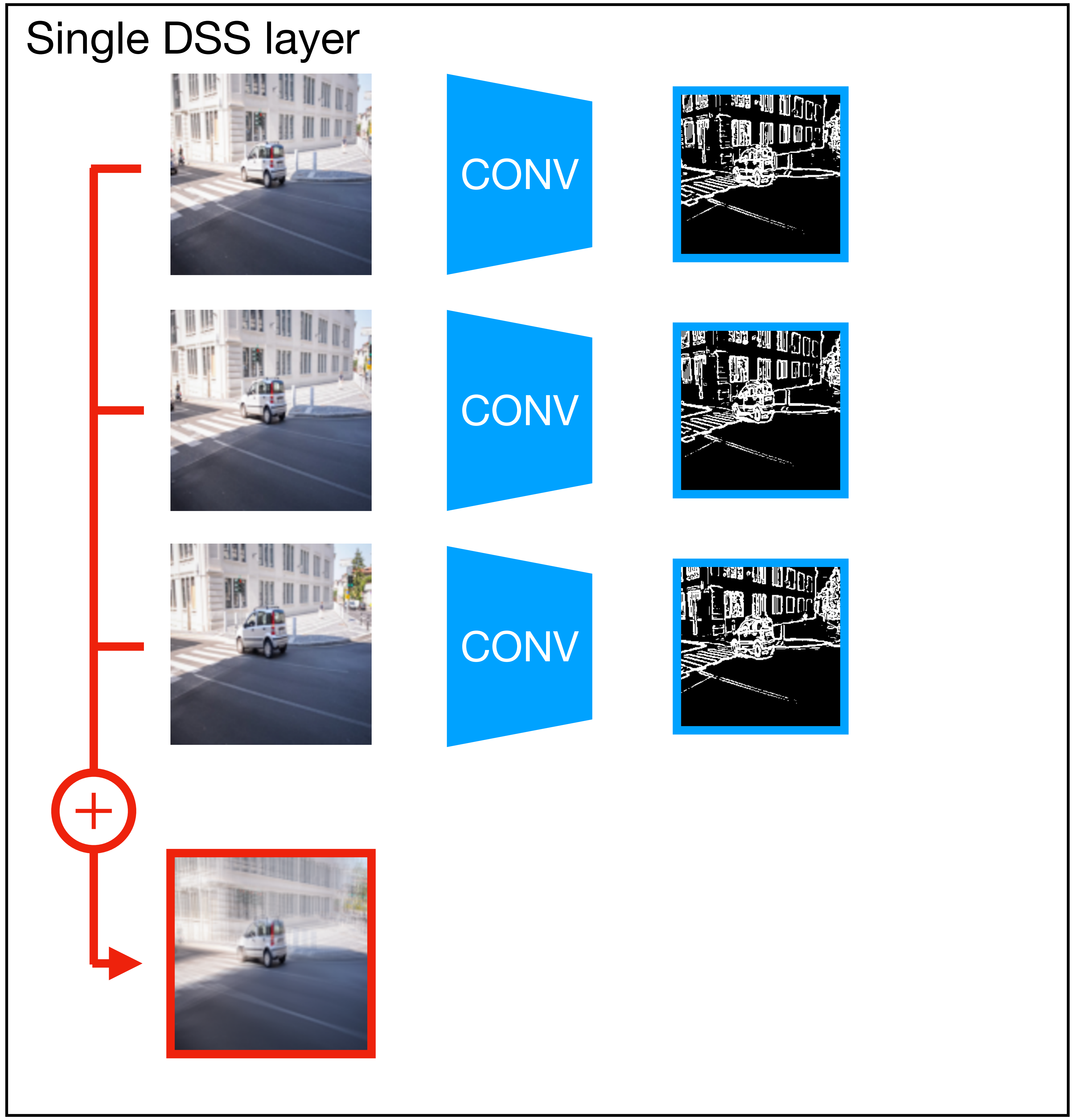


DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

G-equivariant layers are convolutions

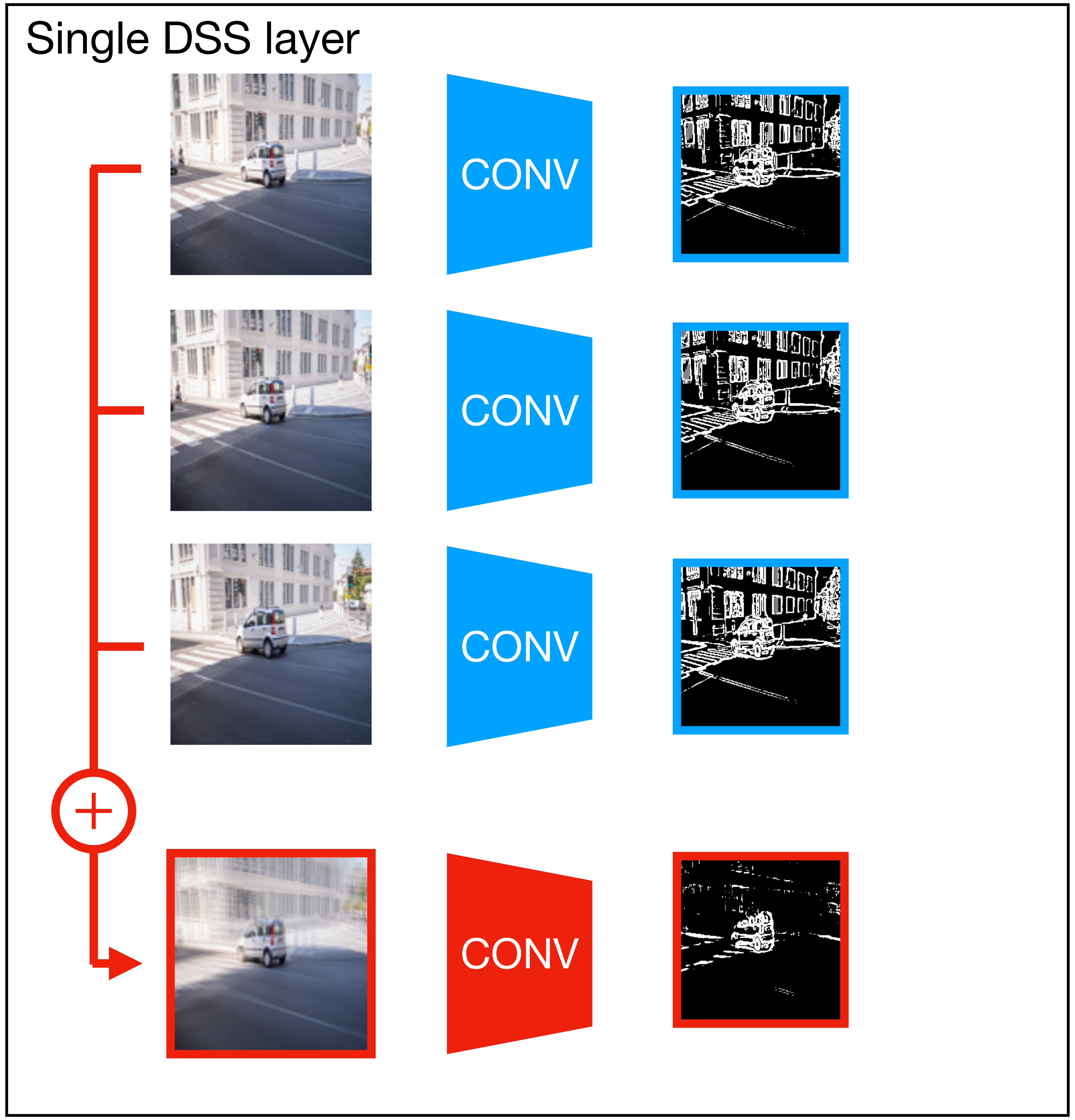


DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

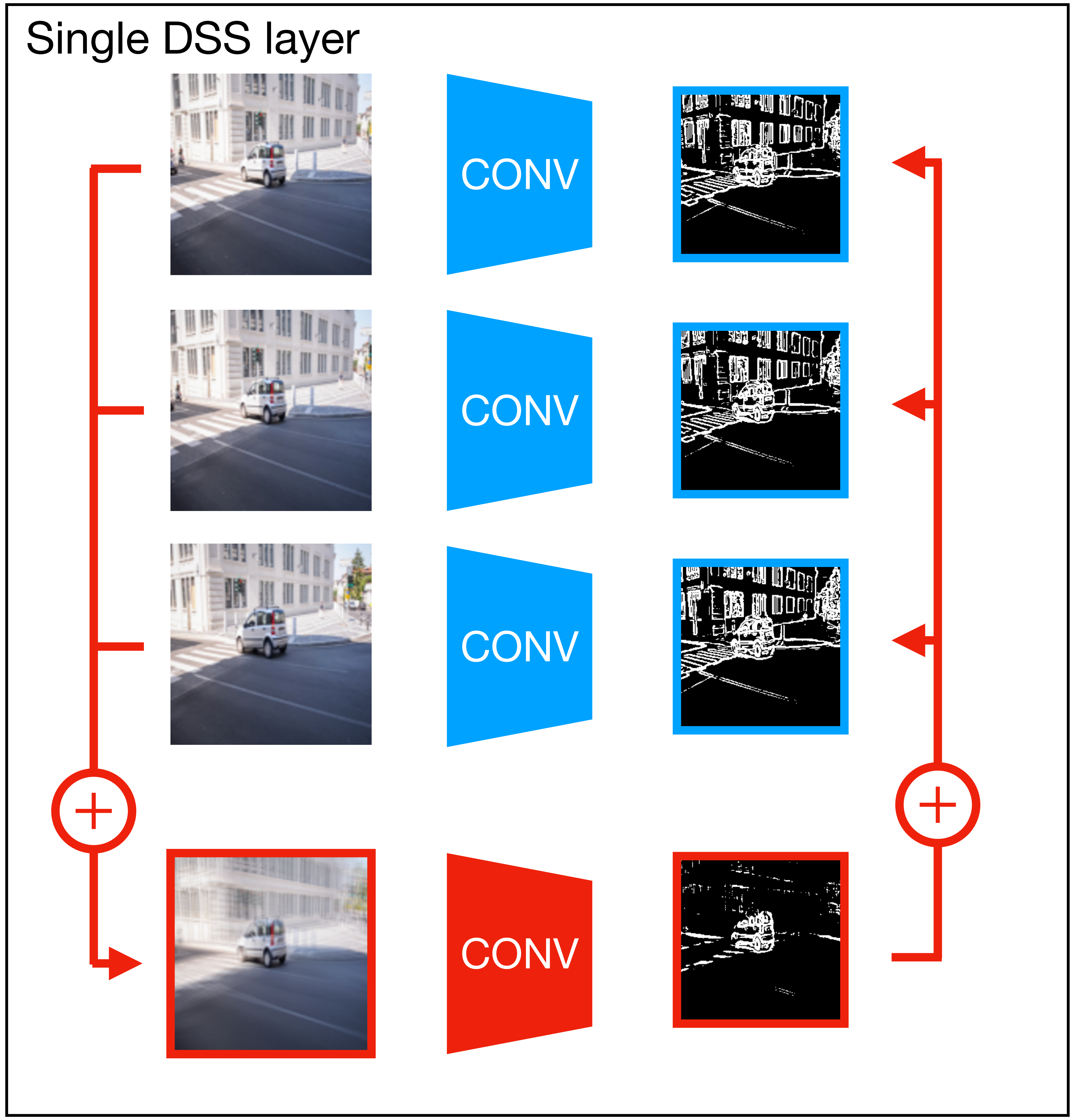
G-equivariant layers are convolutions



DSS for images

Siamese part

Information sharing part



Expressive power

Theorem

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

Expressive power

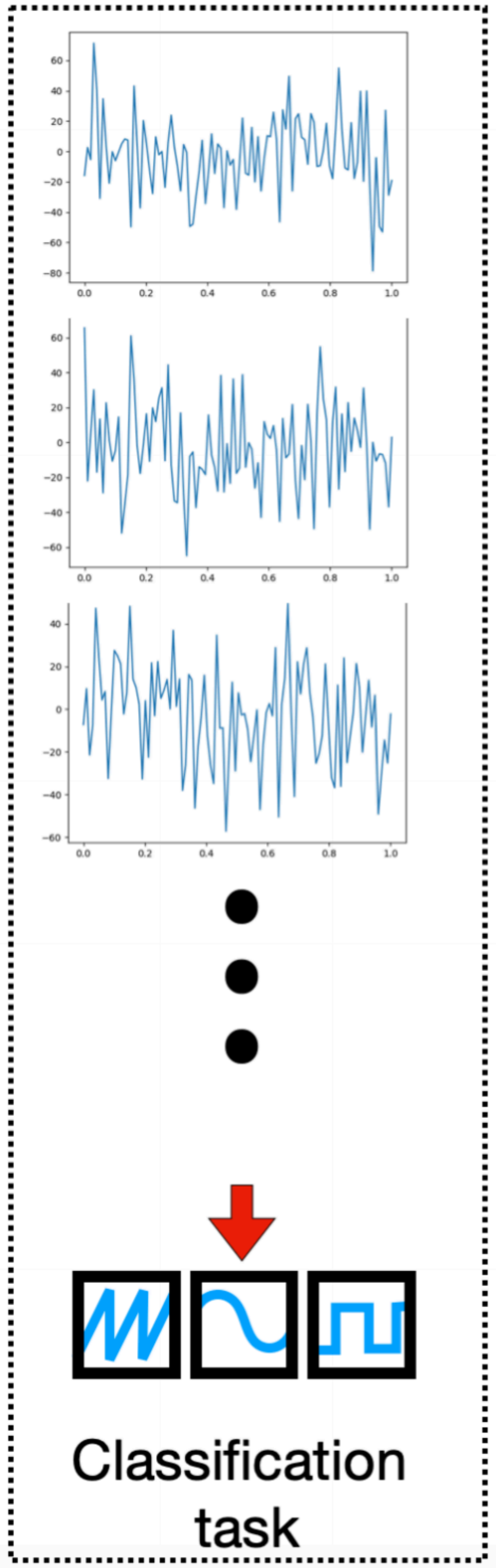
Theorem

If G-equivariant networks are universal appoximators 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 $\mathbb{R}[x_1, ..., x_n]^H$ is finitely generated.
  - Generators can be used to create continuous unique encodings for elements in $\mathbb{R}^{n\times d}/H$

Results

Signal classification



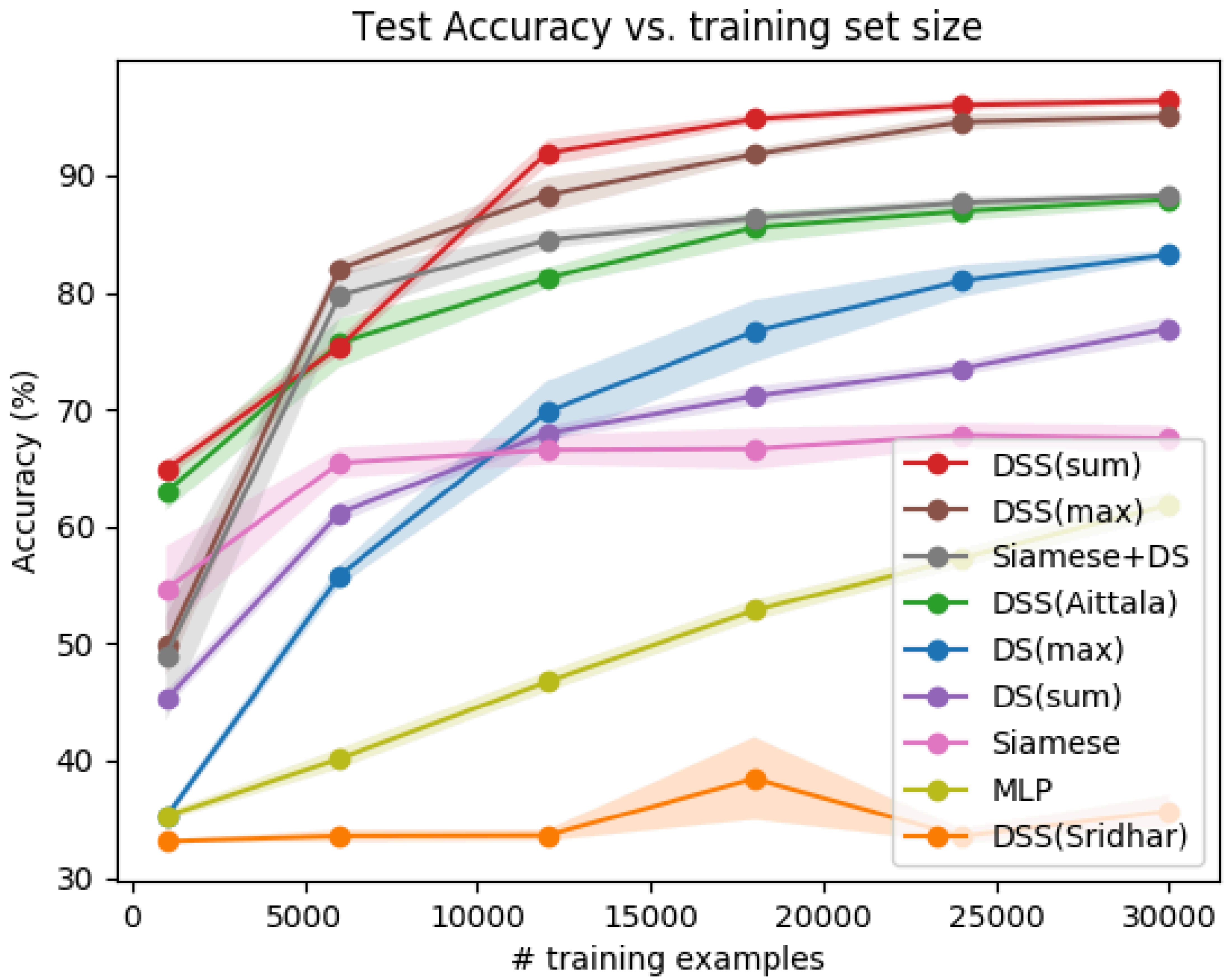
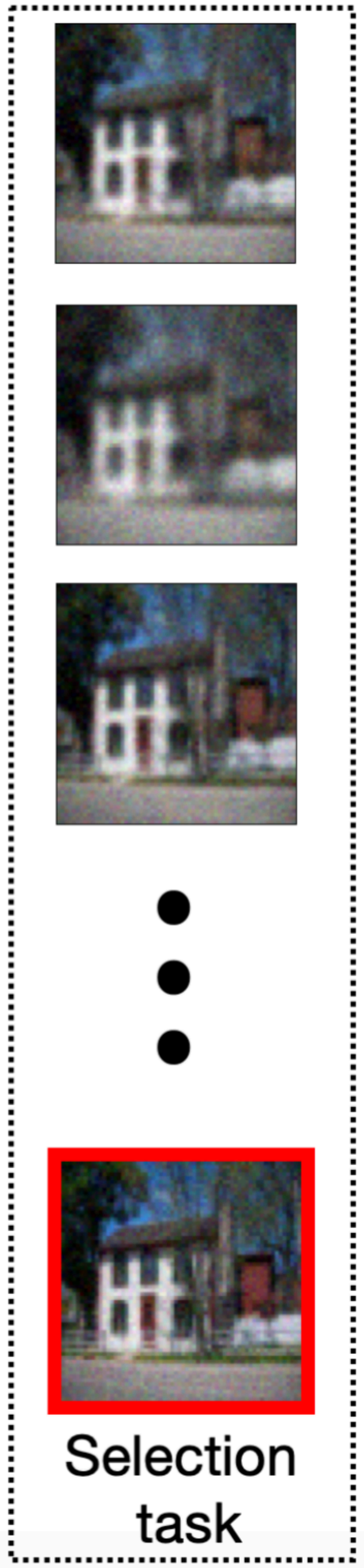


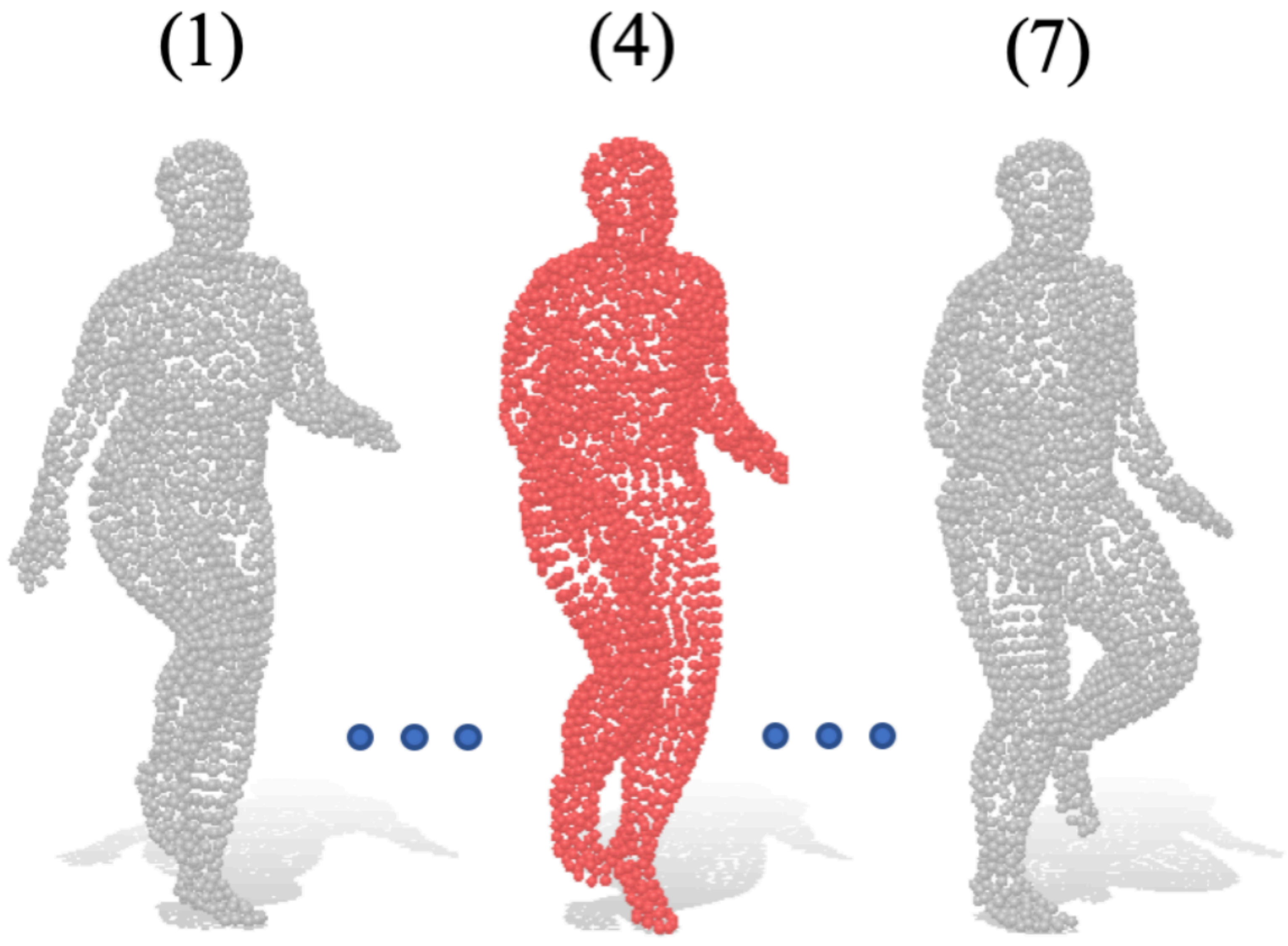
Image selection



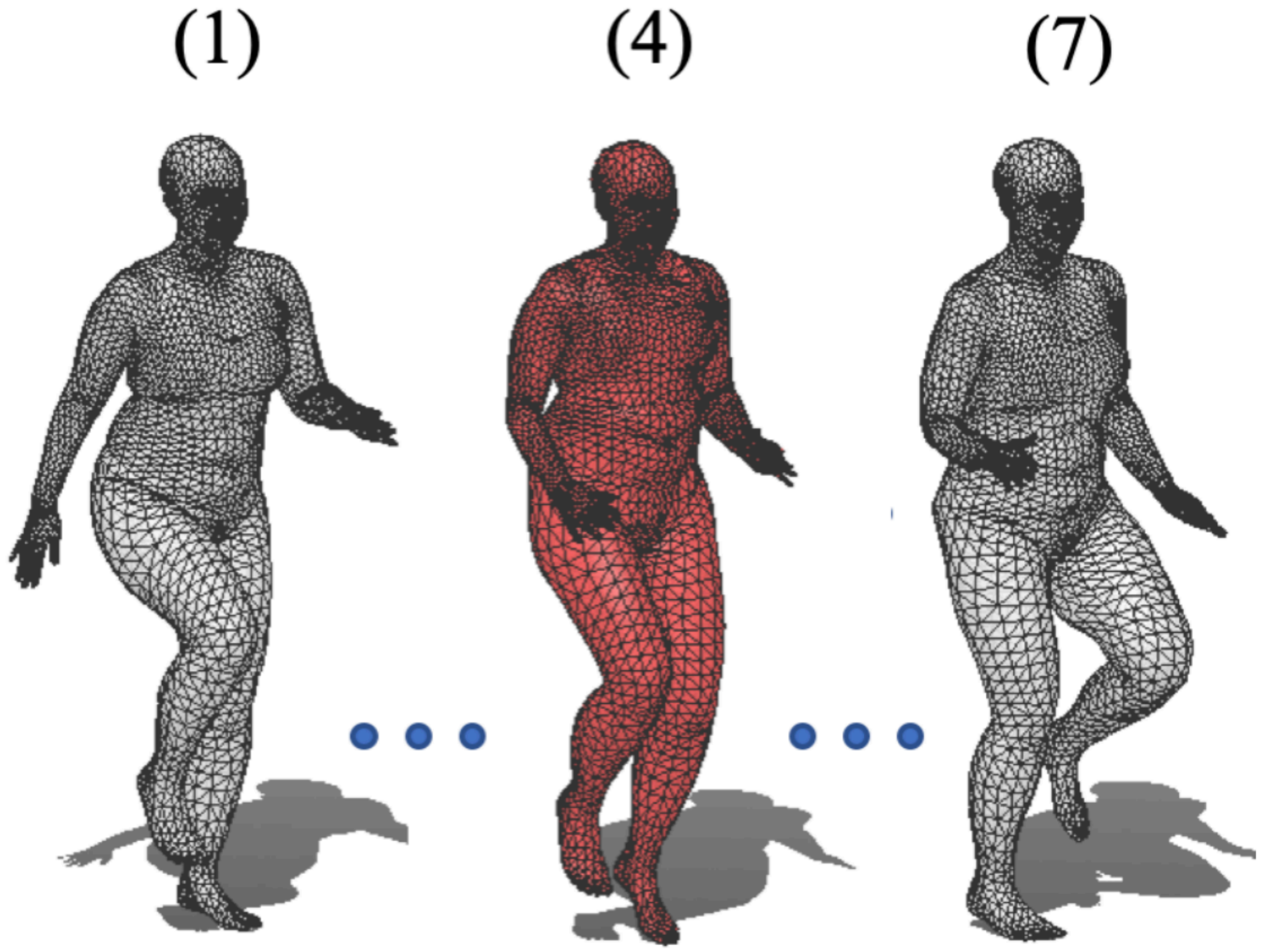
Noise type and strength	Late Aggregation	Early Aggregation			
	Siamese+DS	DSS (sum)	DSS (max)	DSS (Sridahr)	DSS (Aittala)
Gaussian $\sigma = 10$	77.2% ± 0.37	**78.48%** ± 0.48	77.99% ± 1.1	76.8% ± 0.25	78.34% ± 0.49
Gaussian $\sigma = 30$	65.89% ± 0.66	**68.35%** ± 0.55	67.85% ± 0.40	61.52% ± 0.54	66.89% ± 0.58
Gaussian $\sigma = 50$	59.24% ± 0.51	**62.6%** ± 0.45	61.59% ± 1.00	55.25% ± 0.40	62.02% ± 1.03
Occlusion 10%	82.15% ± 0.45	83.13%± 1.00	**83.27** ± 0.51	83.21% ± 0.338	83.19% ± 0.67
Occlusion 30%	77.47% ± 0.37	78% ± 0.89	78.69% ± 0.32	**78.71%** ± 0.26	78.27% ± 0.67
Occlusion 50%	76.2% ± 0.82	**77.29%** ± 0.40	76.64% ± 0.45	77.04% ± 0.75	77.03% ± 0.58

Shape selection

Dataset	Data type	Late Aggregation Siamese+DS	Early Aggregation DSS (sum)	DSS (max)	DSS (Sridhar)	DSS (Aittala)
UCF101	Images	36.41% $\pm$ 1.43	76.6% $\pm$ 1.51	76.39% $\pm$ 1.01	60.15% $\pm$ 0.76	**77.96%** $\pm$ 1.69
Dynamic Faust	Point-clouds	22.26% $\pm$ 0.64	42.45% $\pm$ 1.32	28.71% $\pm$ 0.64	**54.26%** $\pm$ 1.66	26.43% $\pm$ 3.92
Dynamic Faust	Graphs	26.53% $\pm$ 1.99	44.24% $\pm$ 1.28	30.54% $\pm$ 1.27	**53.16%** $\pm$ 1.47	26.66% $\pm$ 4.25



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