December 16th 2020, String Data Workshop

Leveraging Permutation Group Symmetries for Designing Equivariant Neural Networks

Haggai Maron
NVIDIA Research

NVIDIA®

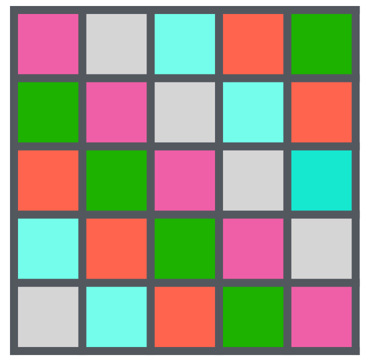
Goal

Given elements $x \in \mathbb{R}^n$ with symmetry group $H \leq S_n$

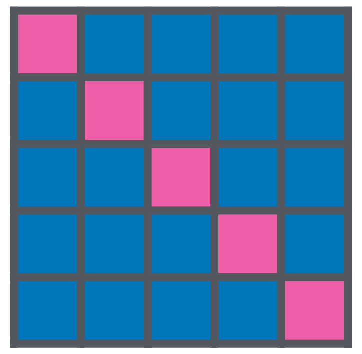
Construct useful **equivariant/invariant** models

Examples:

— Translation equivariance, $H = C_n$ —> CNNs



— Permutation equivariance, $H = S_n$ —> DeepSets



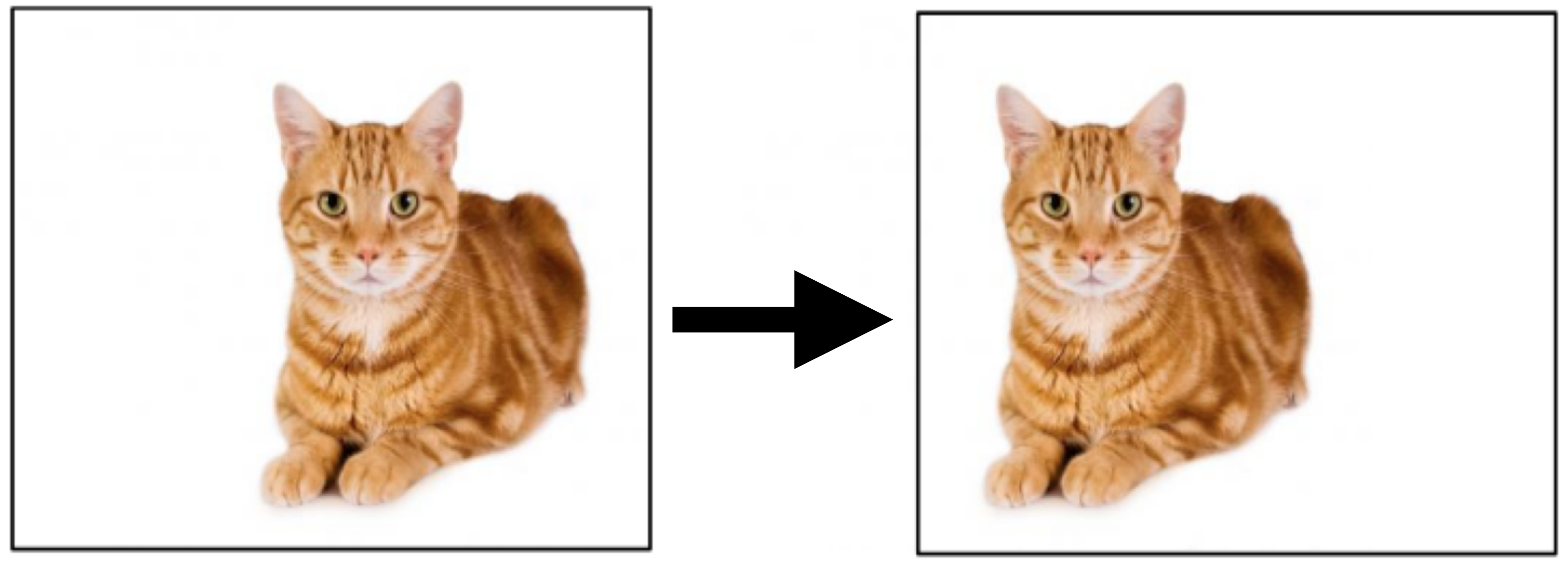
— General H?

Outline

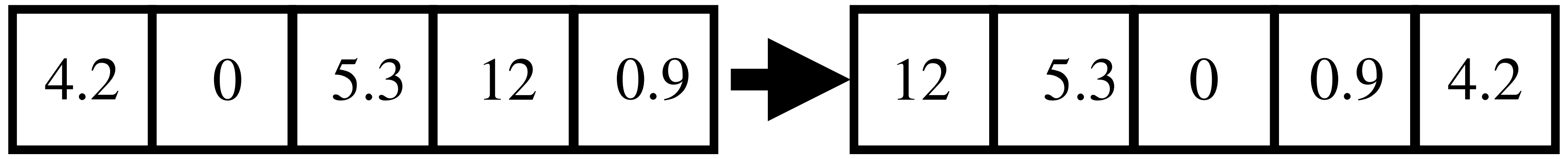
- Symmetry induced parameter sharing
- Examples:
  - Parameter sharing for learning **graphs**
  - Parameter sharing for learning **sets of symmetric elements**

Permutation group actions

- Permutation actions can model natural transformations on vectors
- Examples:
  - x is an image, transformation=translation

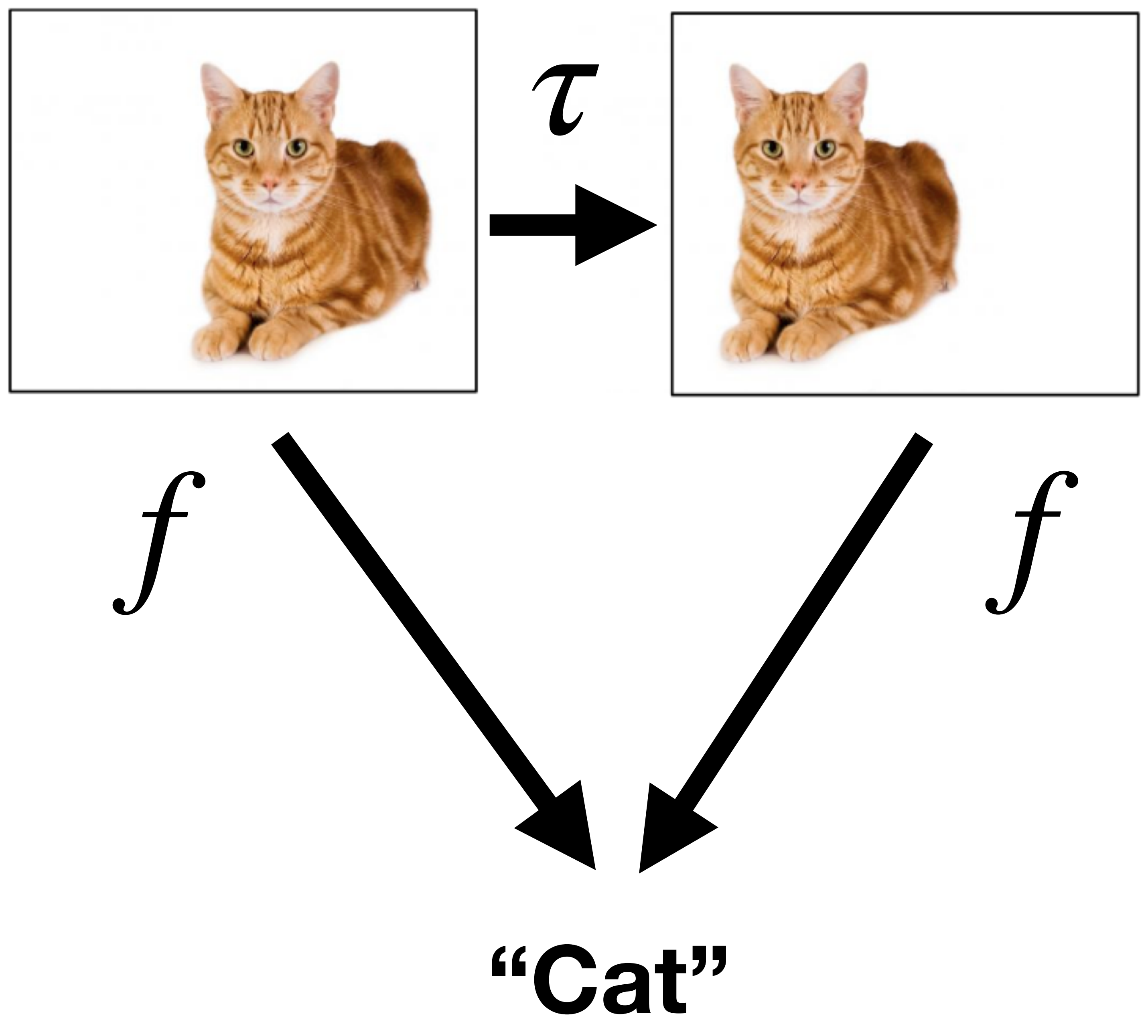


  - x encodes set elements, transformation=reordering



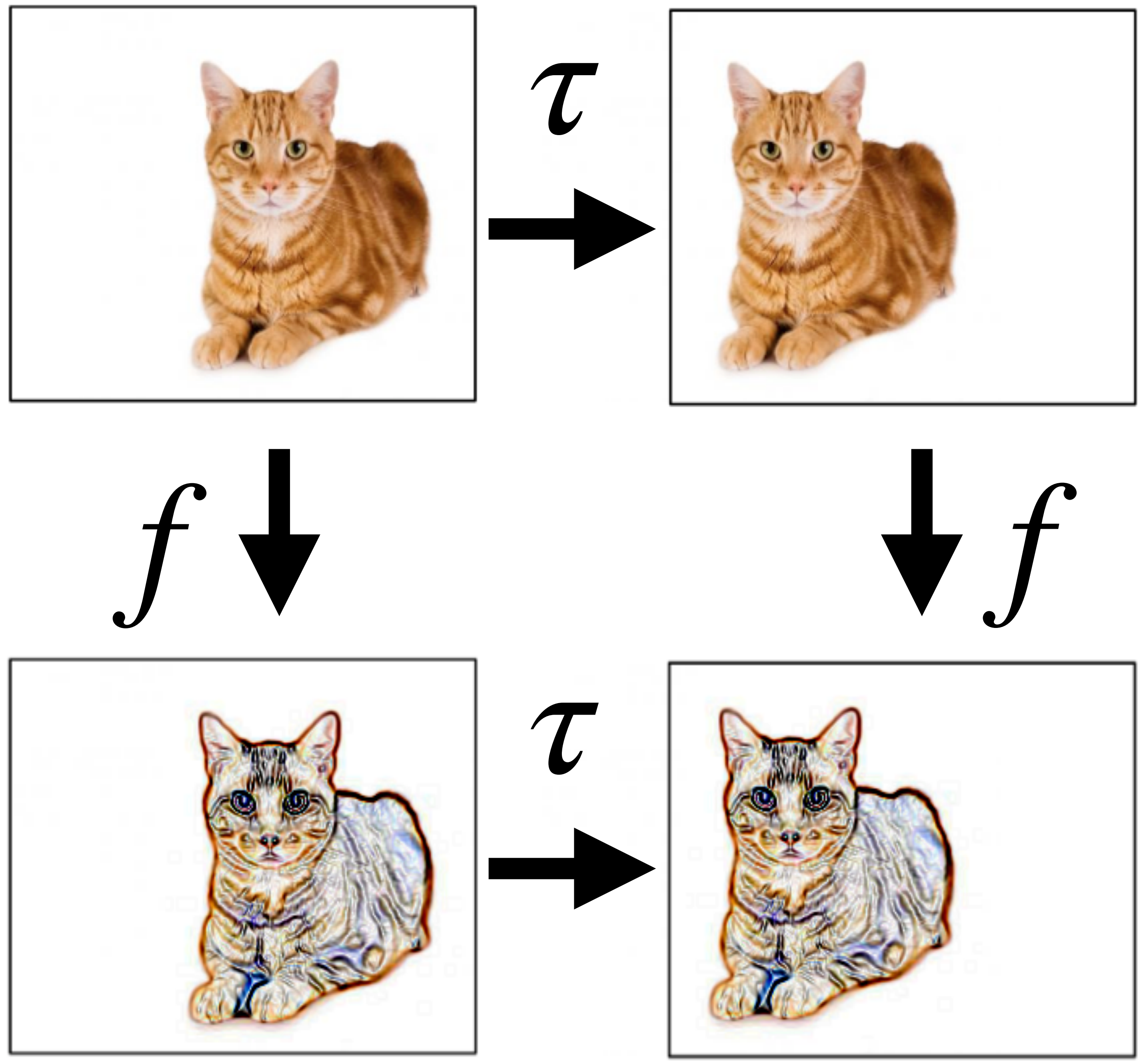
Invariance

- 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



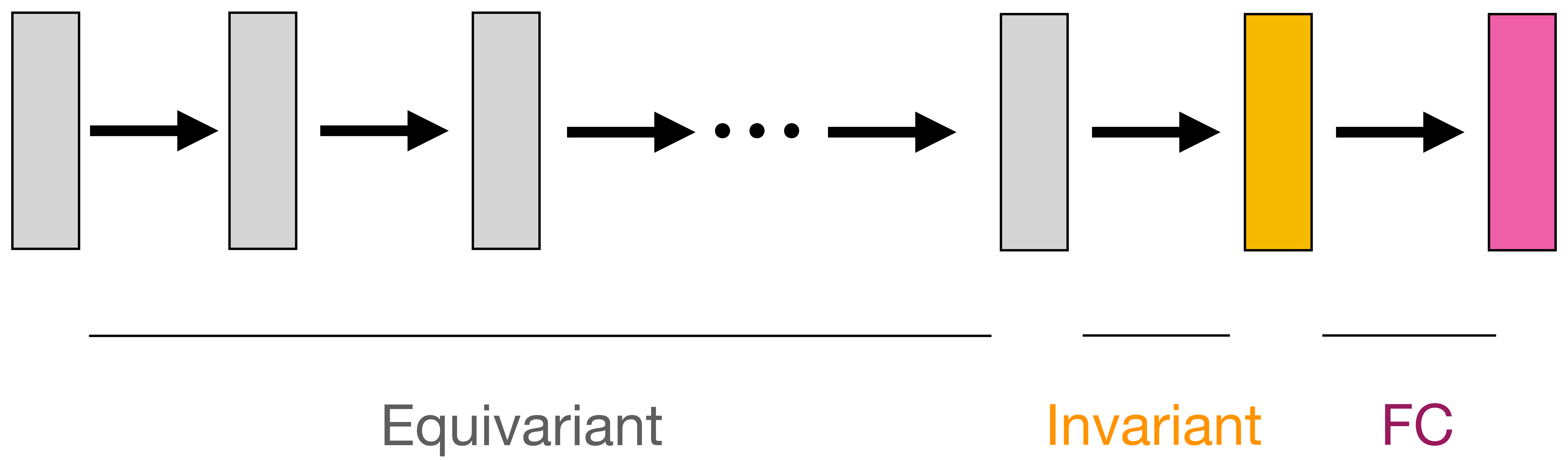
Equivariance

- Let $H \leq S_n$ be a subgroup:
- $f : \mathbb{R}^n \to \mathbb{R}^n$ is equivariant if $f(\tau \cdot x) = \tau \cdot f(x)$,
- e.g. edge detection



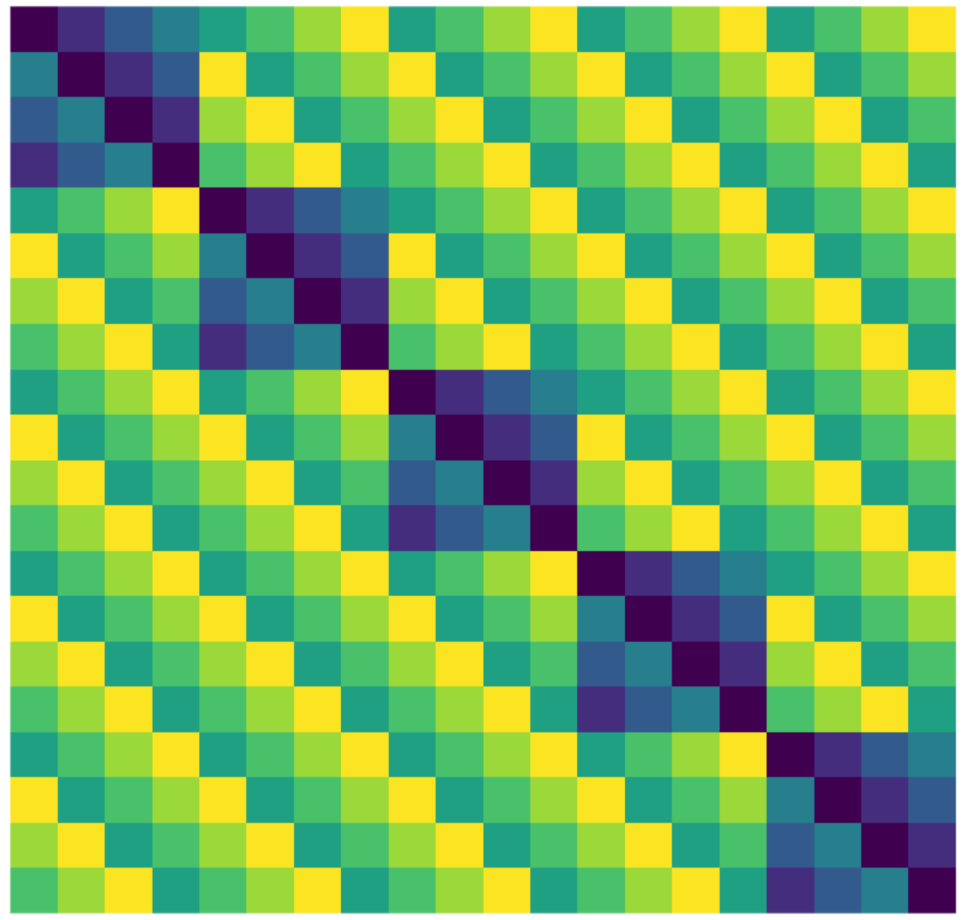
Invariant neural networks

- Invariant by construction



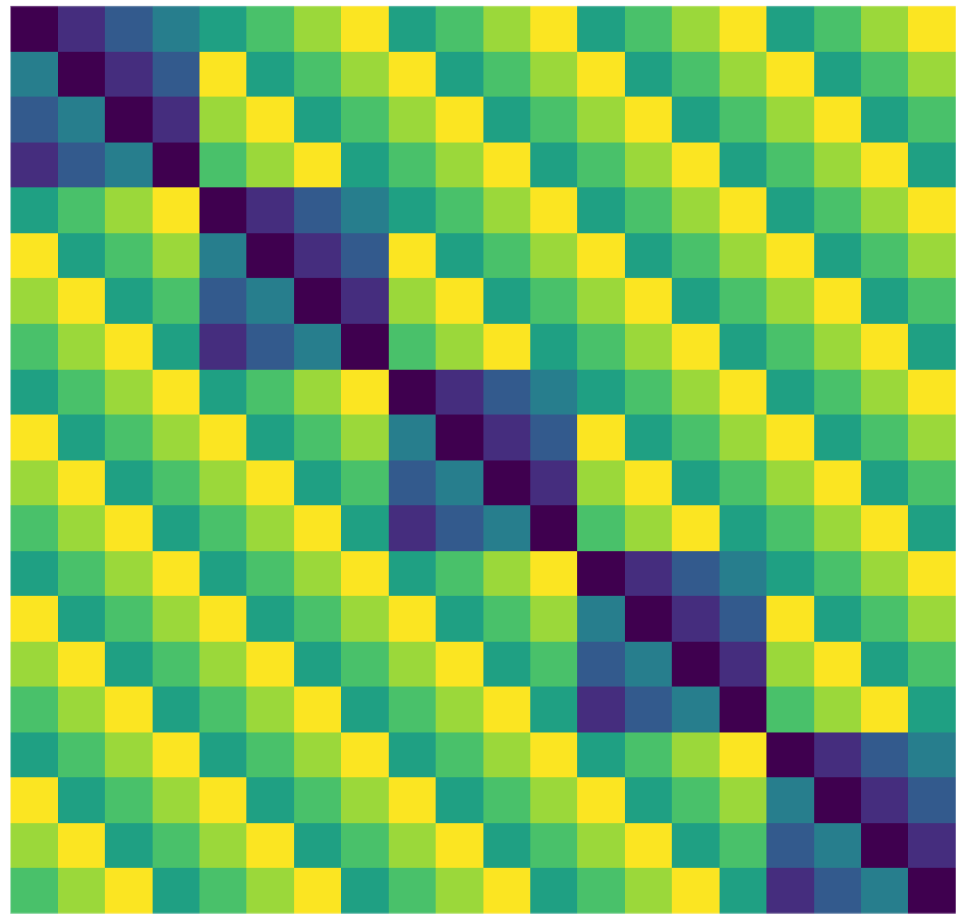
Challenges

- What is the space of linear equivariant layers for specific H?

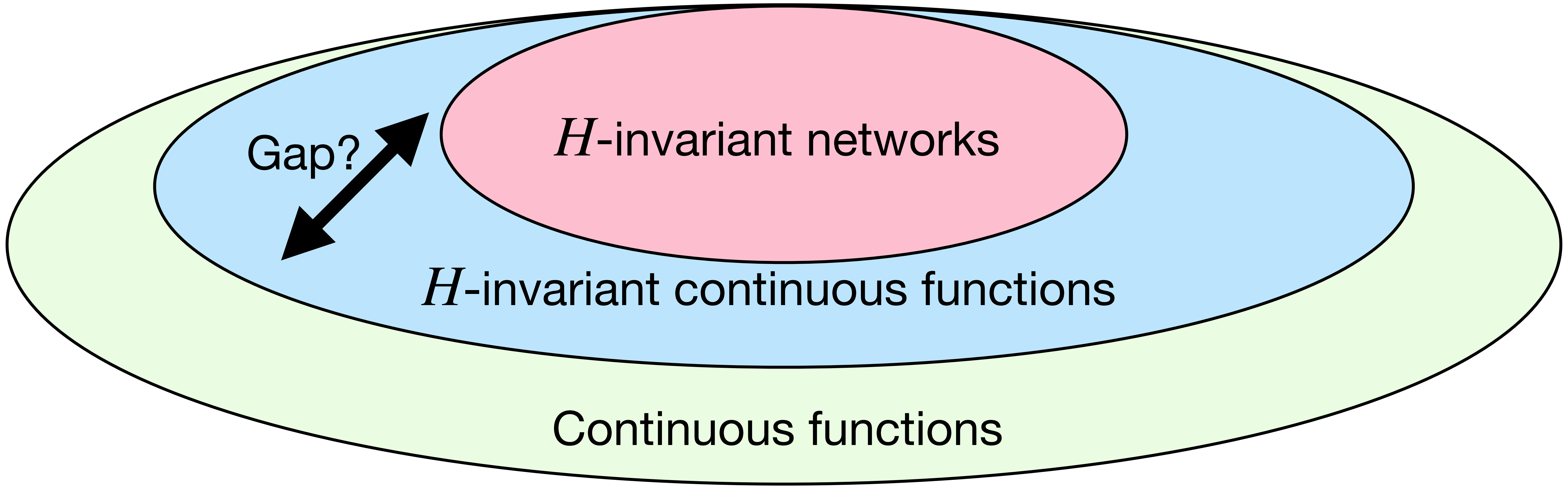


Challenges

- What is the space of linear equivariant layers for specific H?



- Do we lose expressive power?

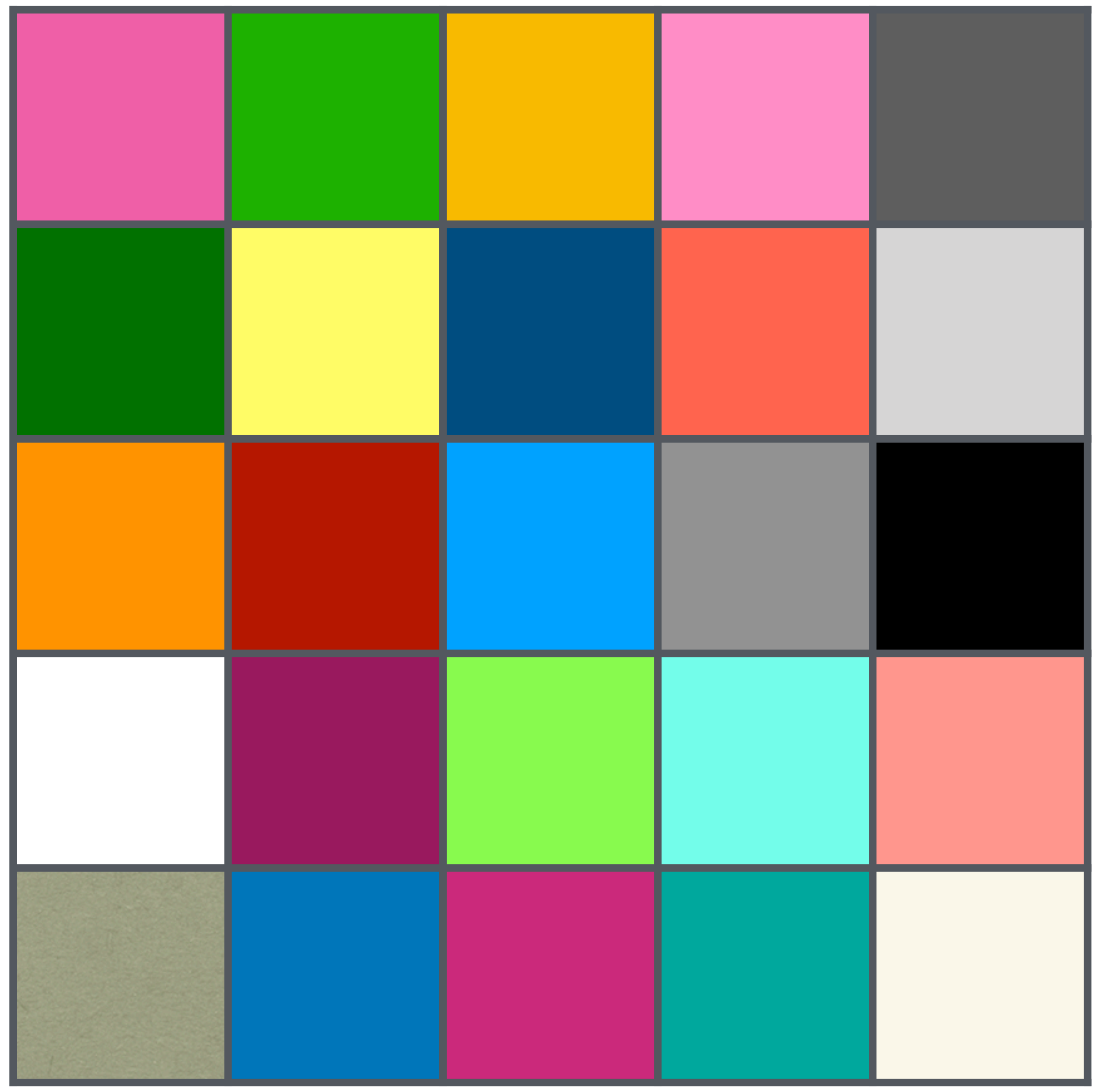


Equivariance and parameter sharing

$$\mathscr{L}(x) = \sigma(L(x)) = \sigma(A \, x + b)$$

Structure?

$A =$



Symmetry induced parameter sharing

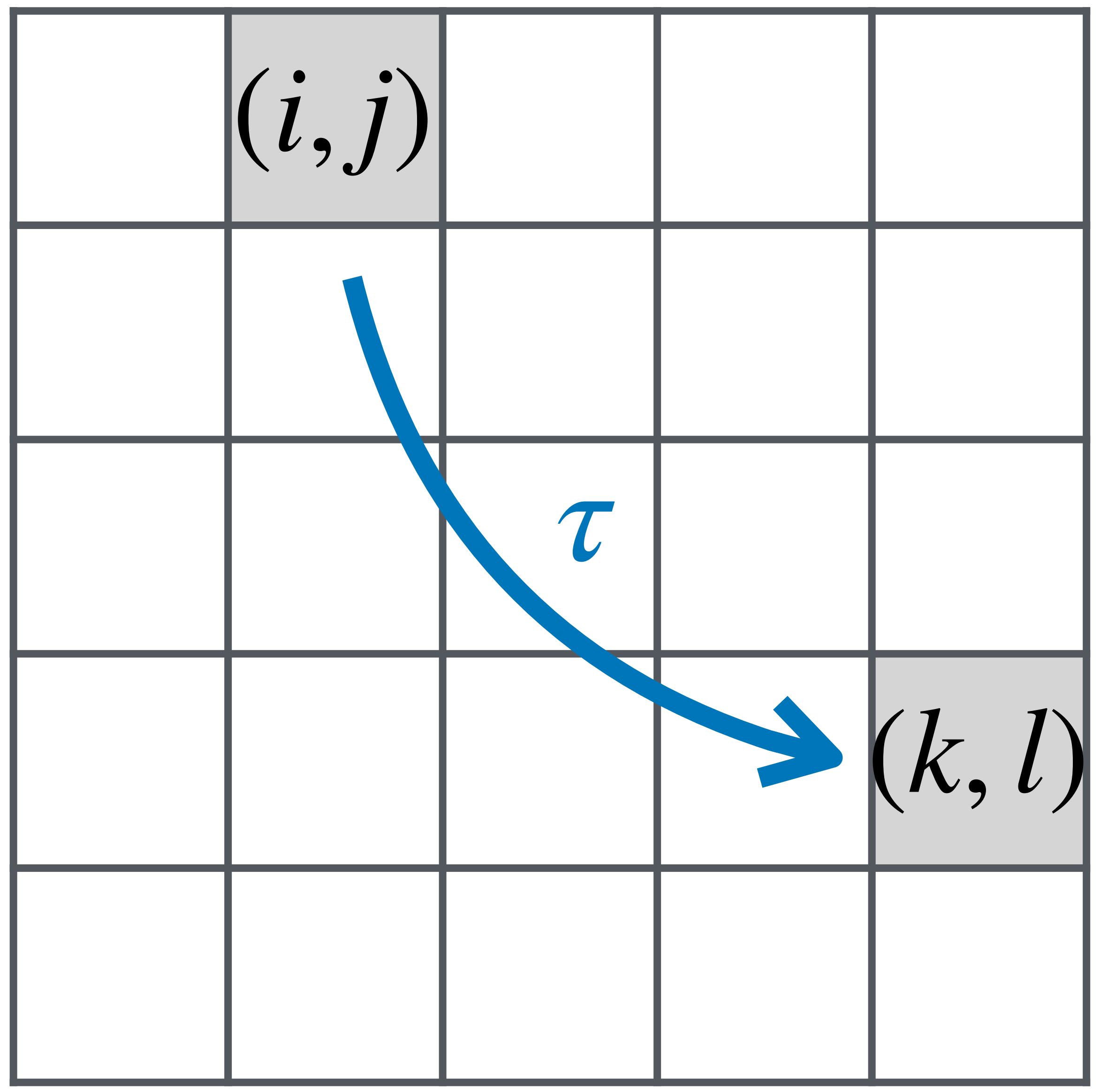
Definition: A parameter matrix $A \in \mathbb{R}^{n \times n}$ has an H-induced parameter sharing scheme if it is induced by H in the following way:

$$A_{ij} = A_{kl}$$

$$\Longleftrightarrow$$

$$(\tau(i), \tau(j)) = (k, l) \text{ for some } \tau \in H$$

$A =$



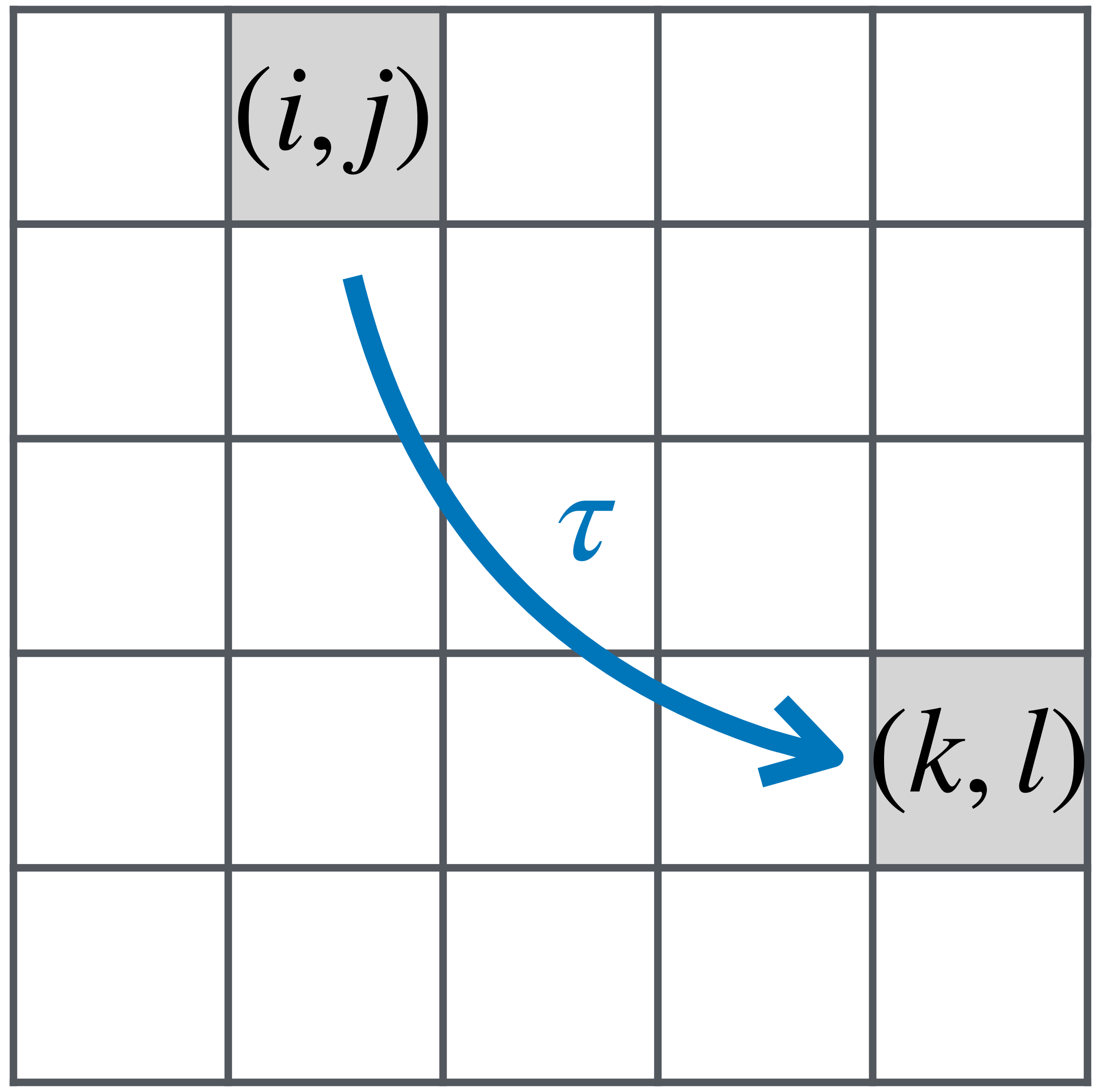
Symmetry induced parameter sharing

Theorem: A matrix $L \in \mathbb{R}^{n \times n}$ represents a *maximal* linear H-equivariant operator if and only if its parameter sharing scheme is induced by H.

$$A_{ij} = A_{kl}$$

$$\iff$$

$$(\tau(i), \tau(j)) = (k, l) \text{ for some } \tau \in H$$

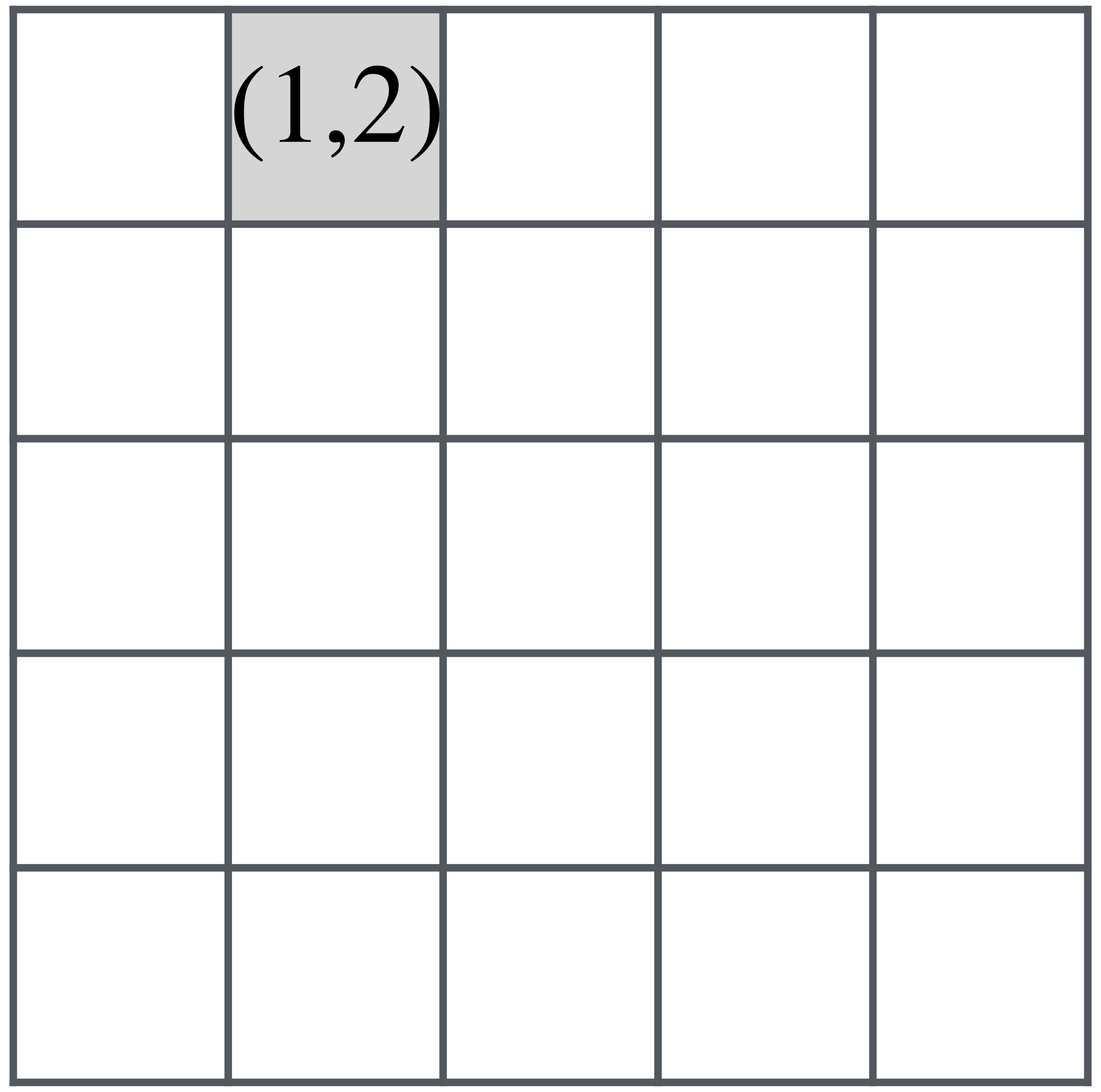
$A =$ 

Wood & Shawe-Taylor 1996, Ravanbakhsh et al. 2017, M. et al. 2019

Examples

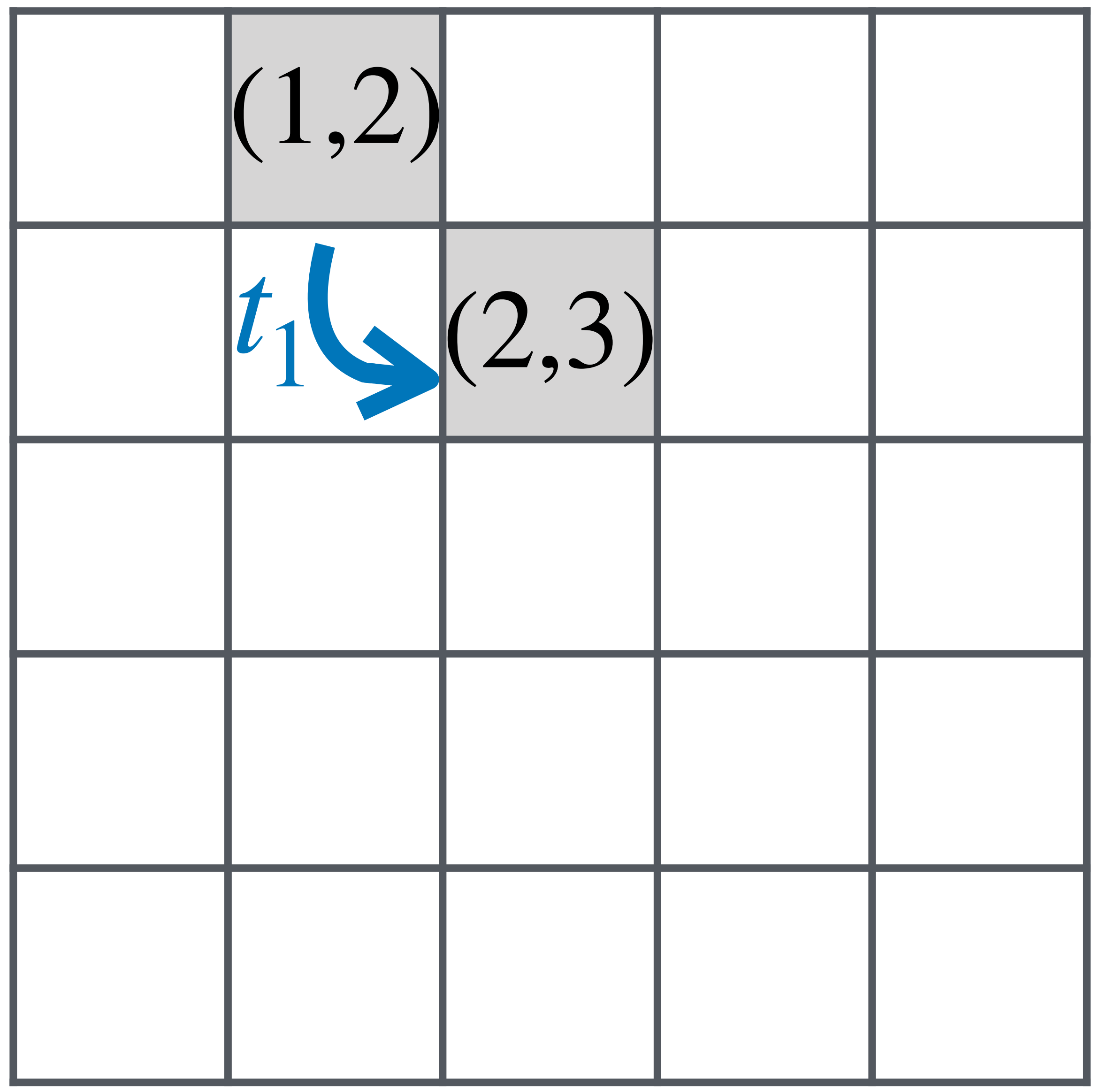
- Translations $H = C_n = \{t_0, \ldots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$

$A =$



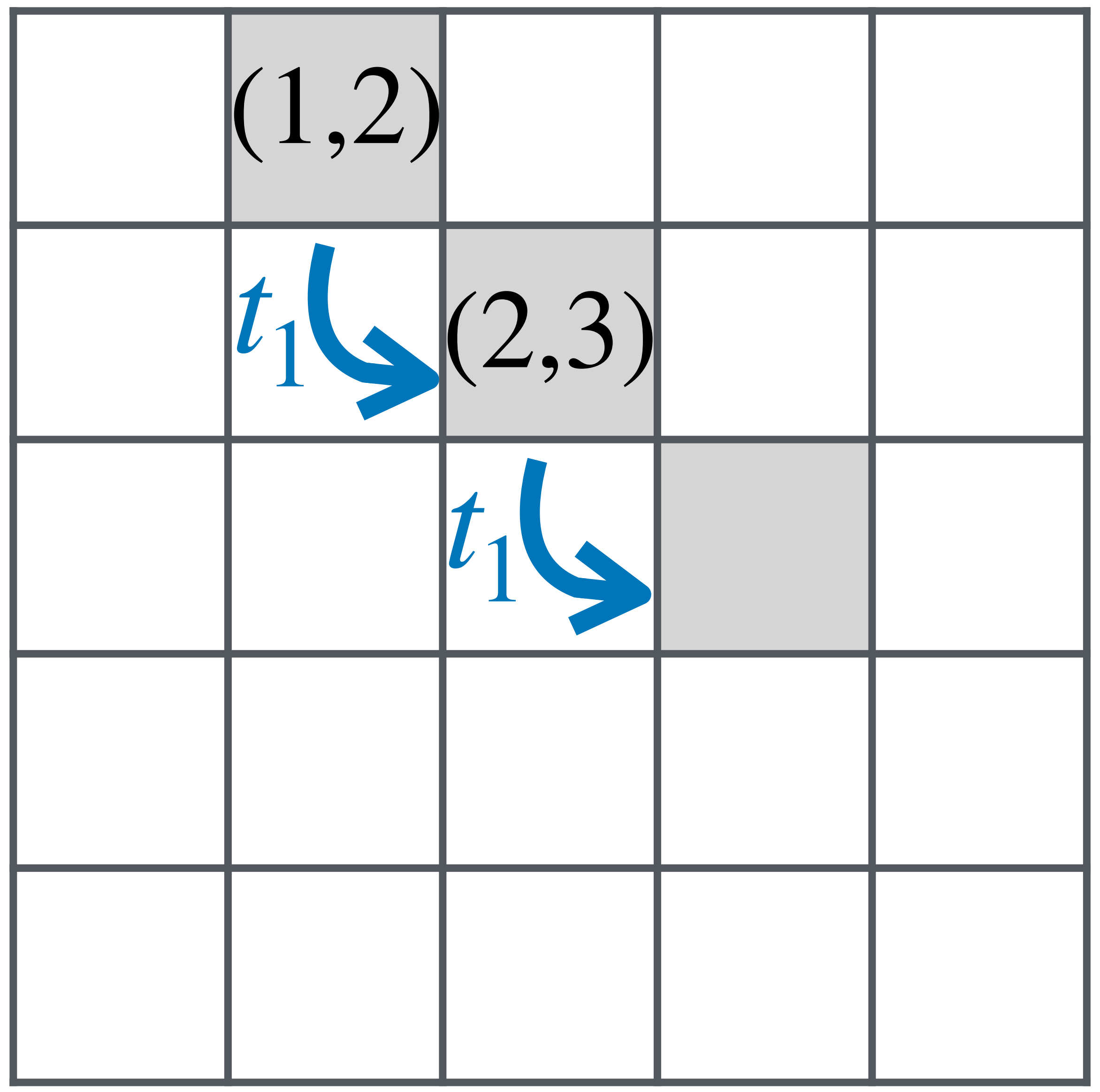
Examples

- Translations $H = C_n = \{t_0, \dots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$

$A =$ 

Examples

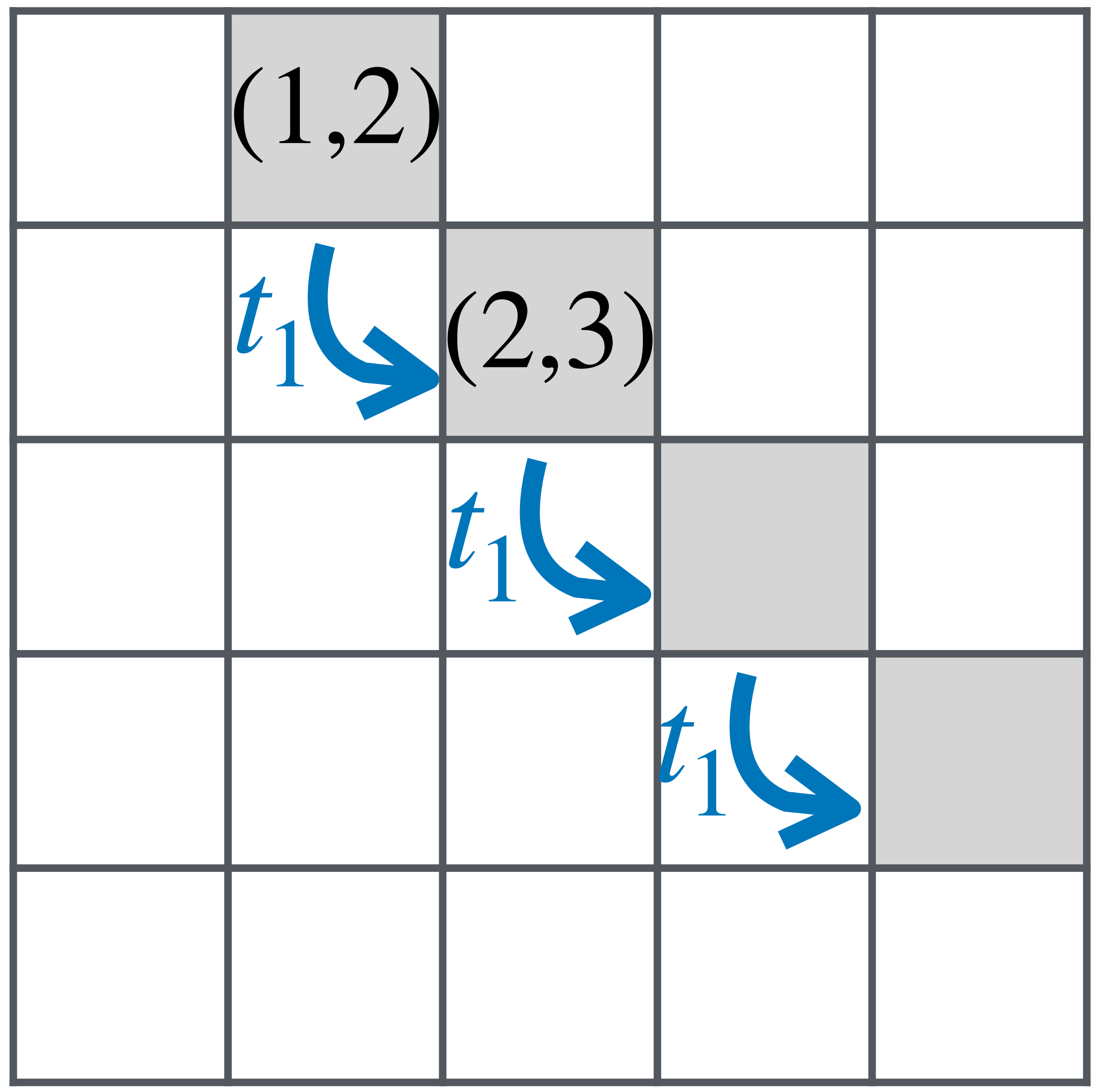
- Translations $H = C_n = \{t_0, \ldots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$

$A =$ 

Examples

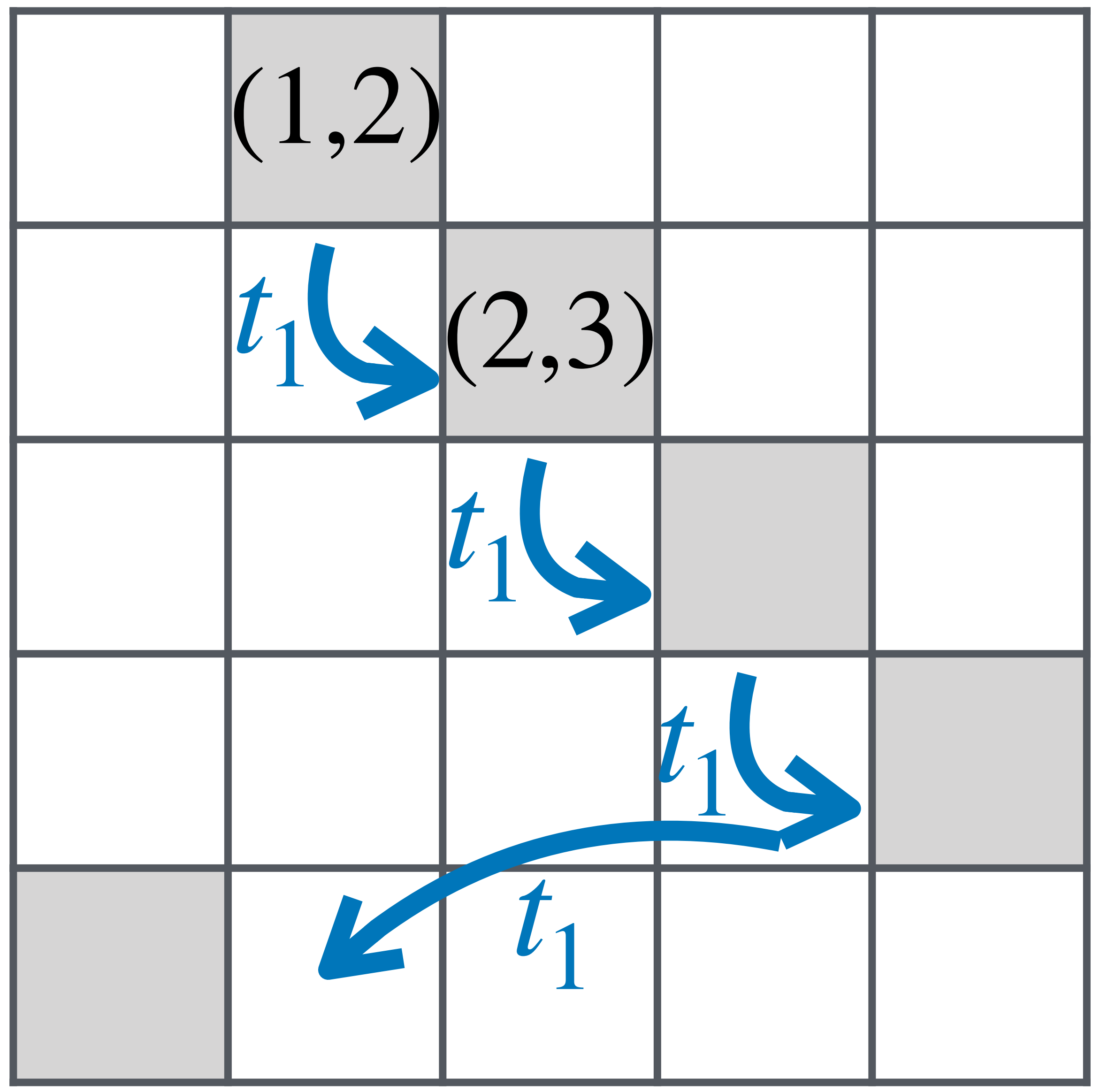
- Translations $H = C_n = \{t_0, \ldots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$

$A =$



Examples

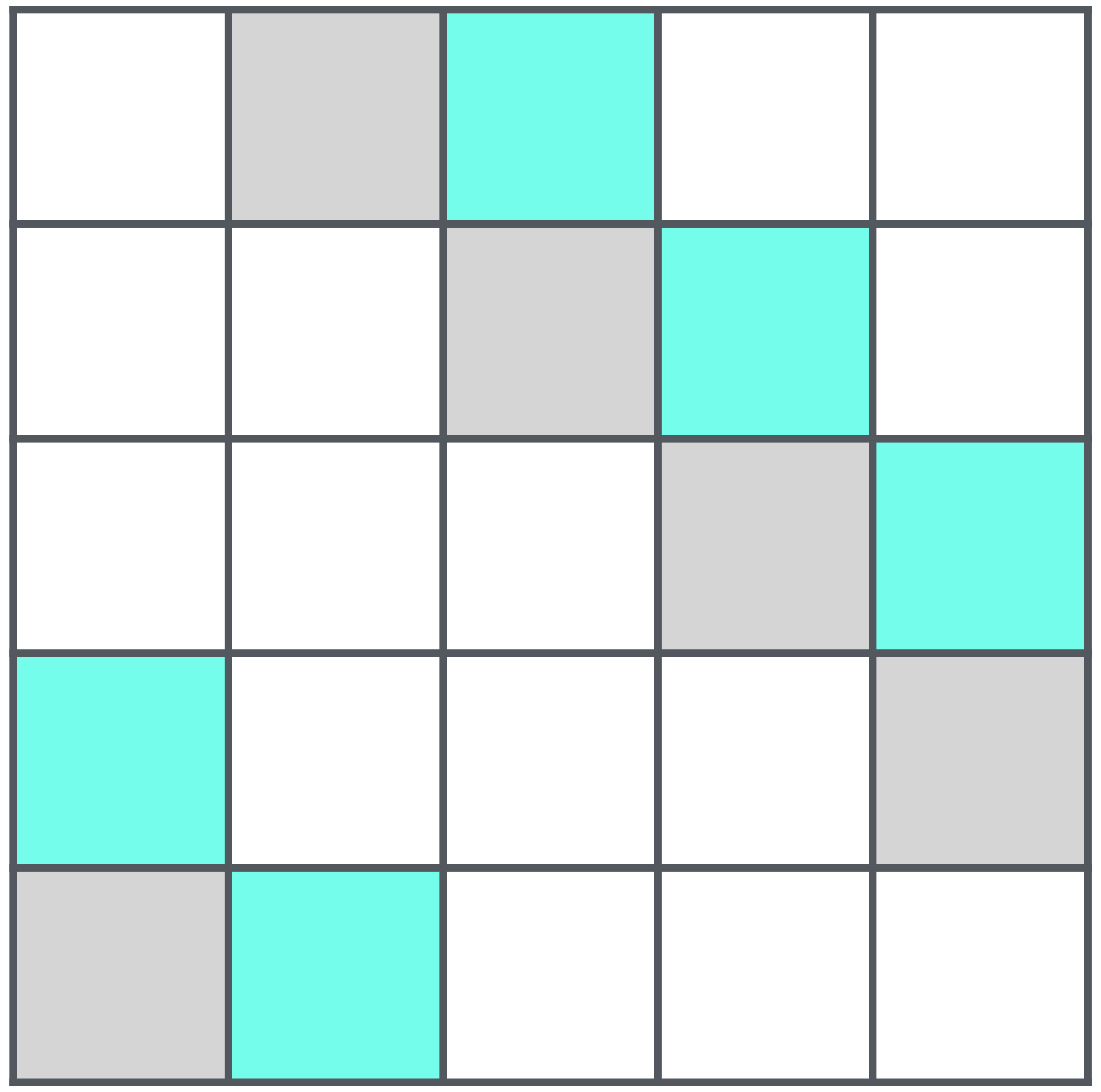
- Translations $H = C_n = \{t_0, \ldots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$

$A =$ 

Examples

- Translations $H = C_n = \{t_0, \ldots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$
- n parameters

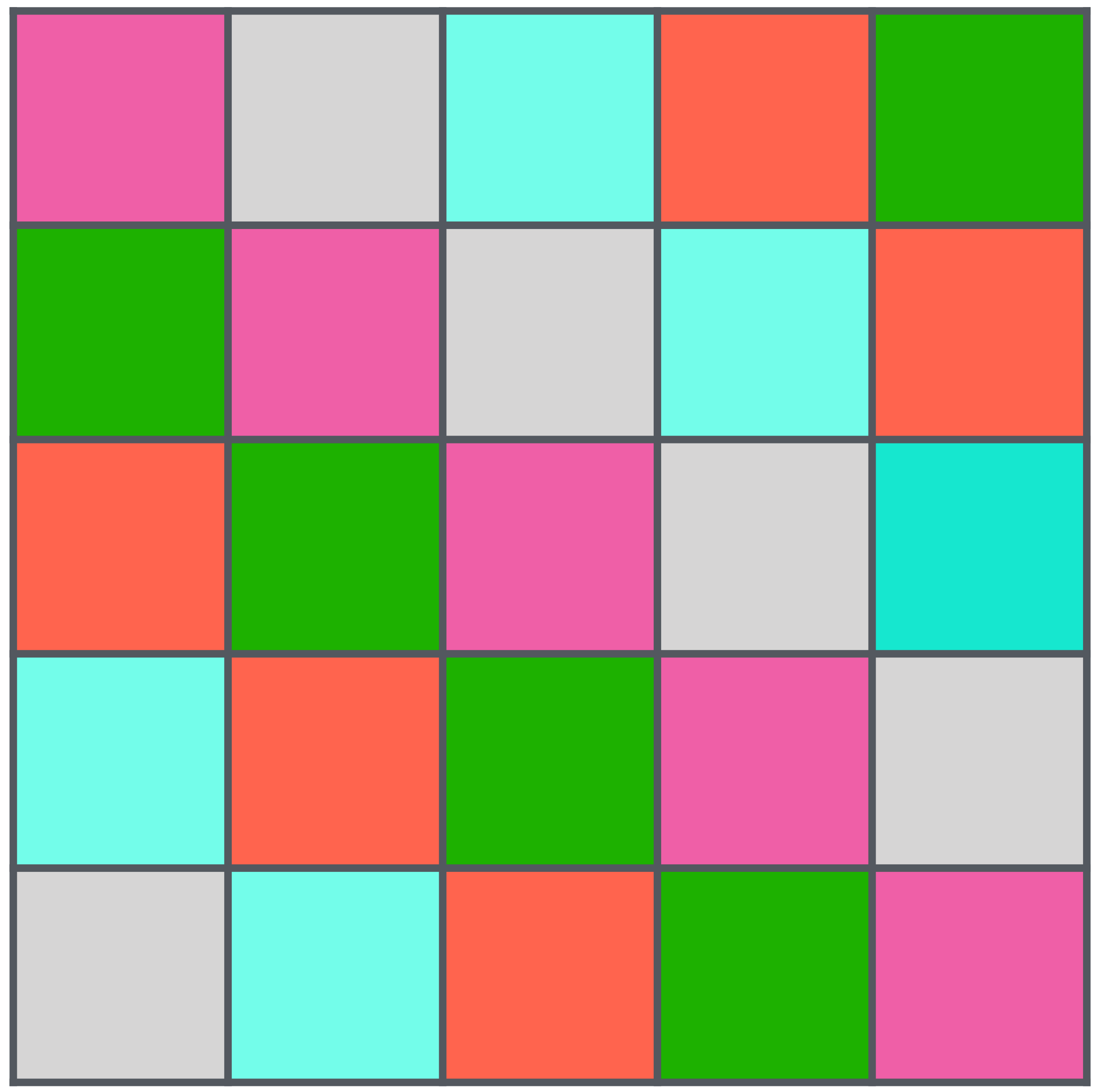
$A =$



Examples

- Translations $H = C_n = \{t_0, \ldots t_{n-1}\}$
- $t_i(j) = i + j \pmod{n}$
- n parameters

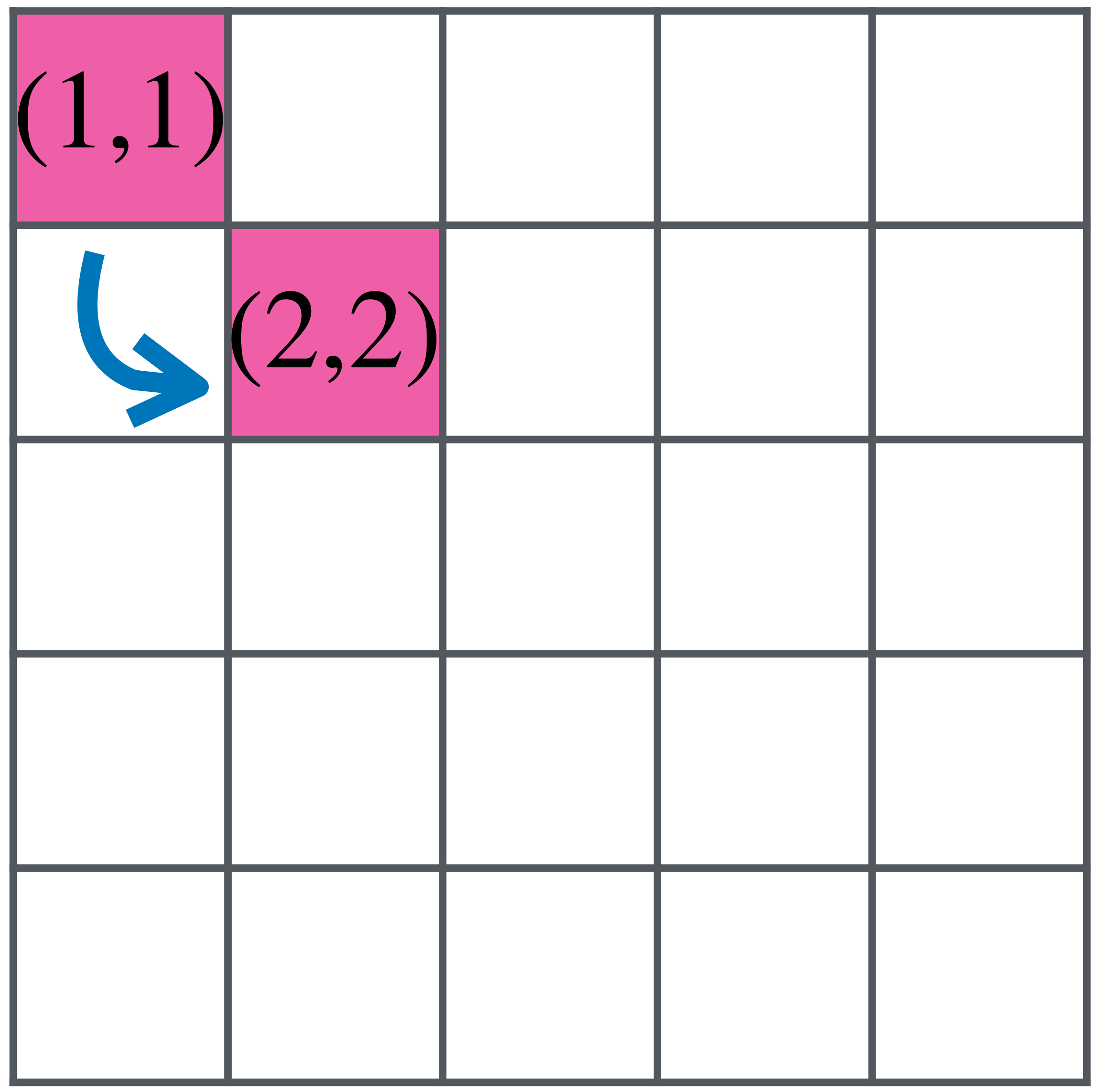
$A =$



Examples

- General permutations $H = S_n$

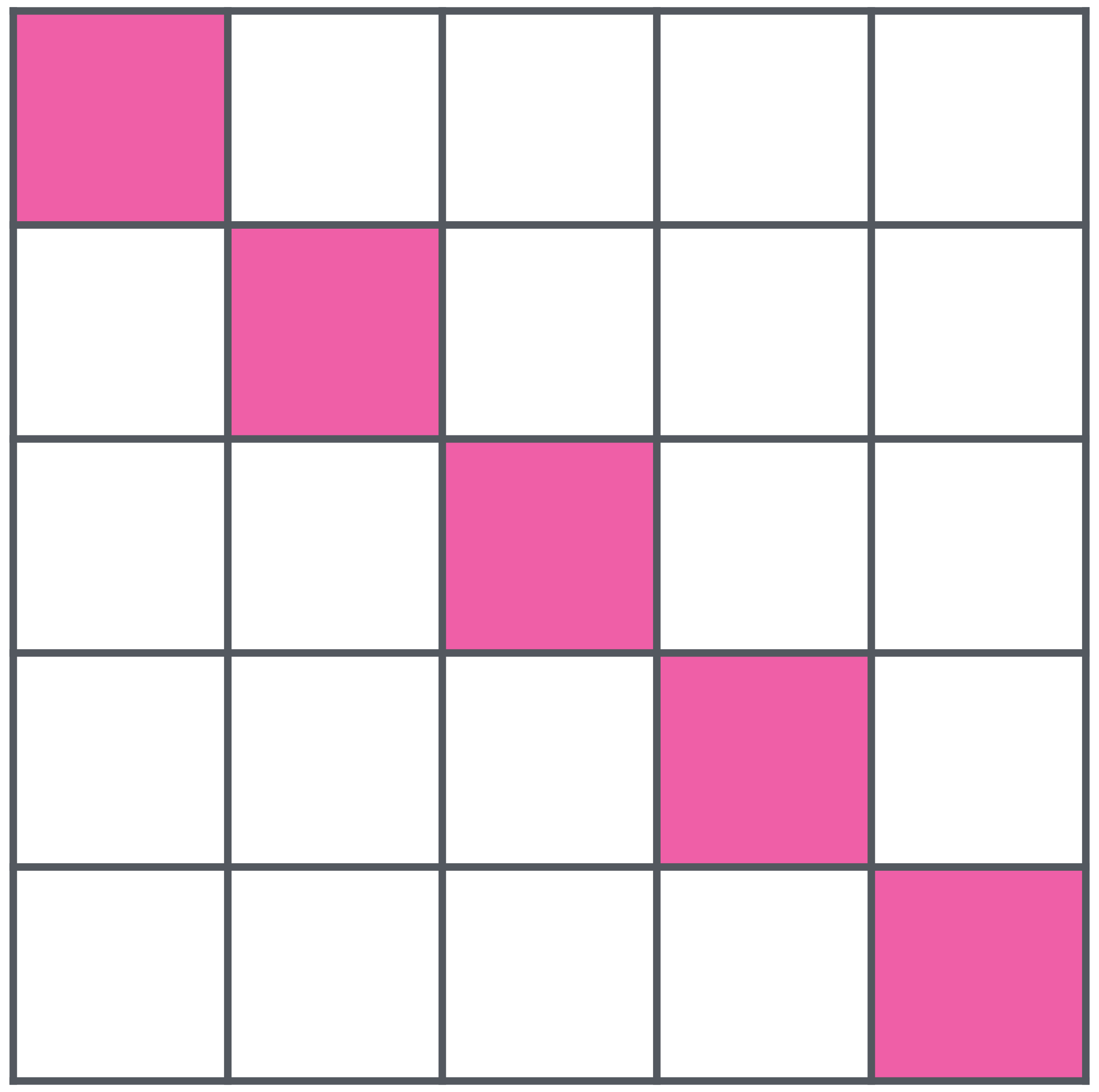
$$A =$$



Zaheer et al. 2017

Examples

- General permutations $H = S_n$

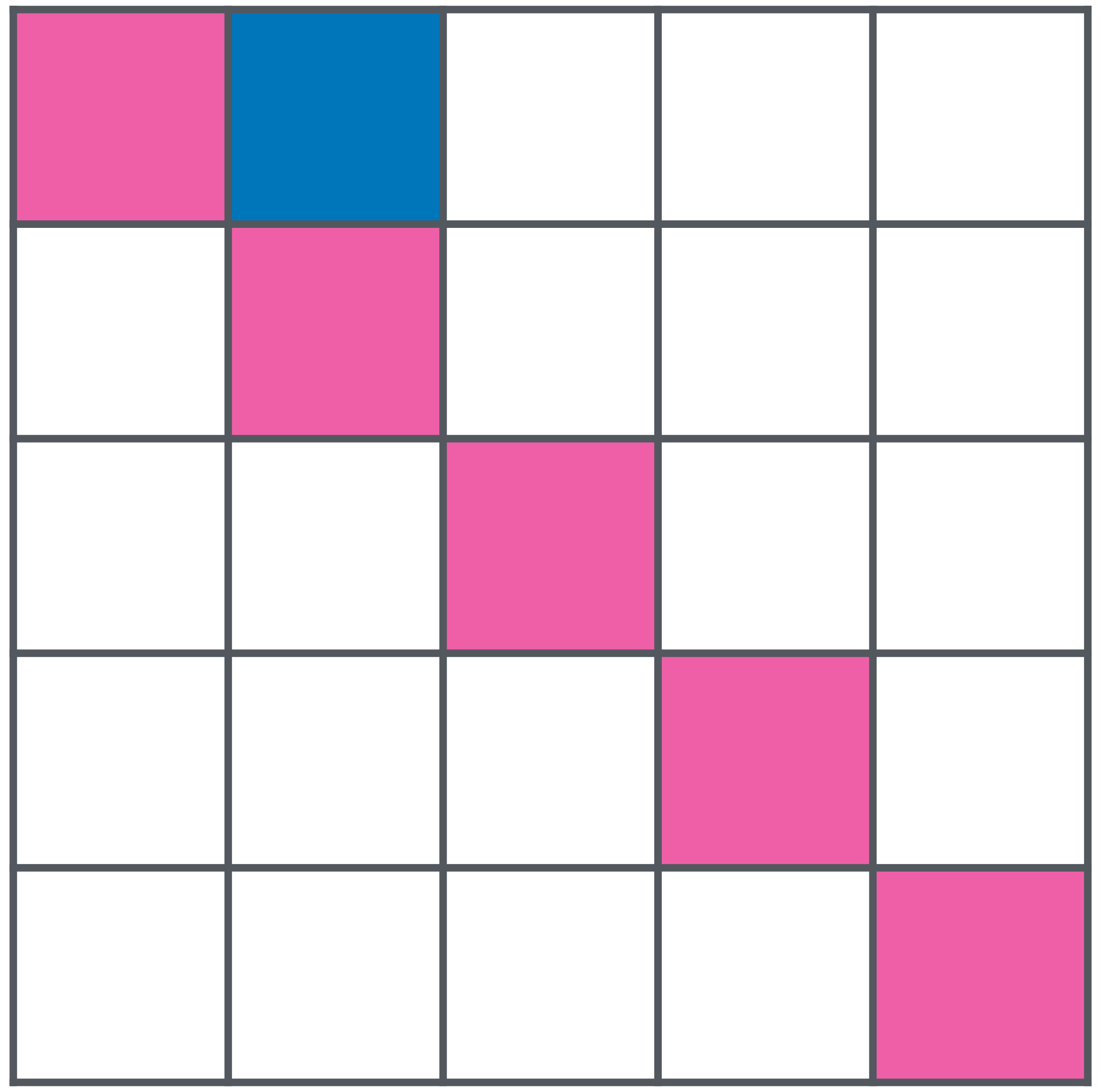
$A =$ 

Zaheer et al. 2017

Examples

- General permutations $H = S_n$

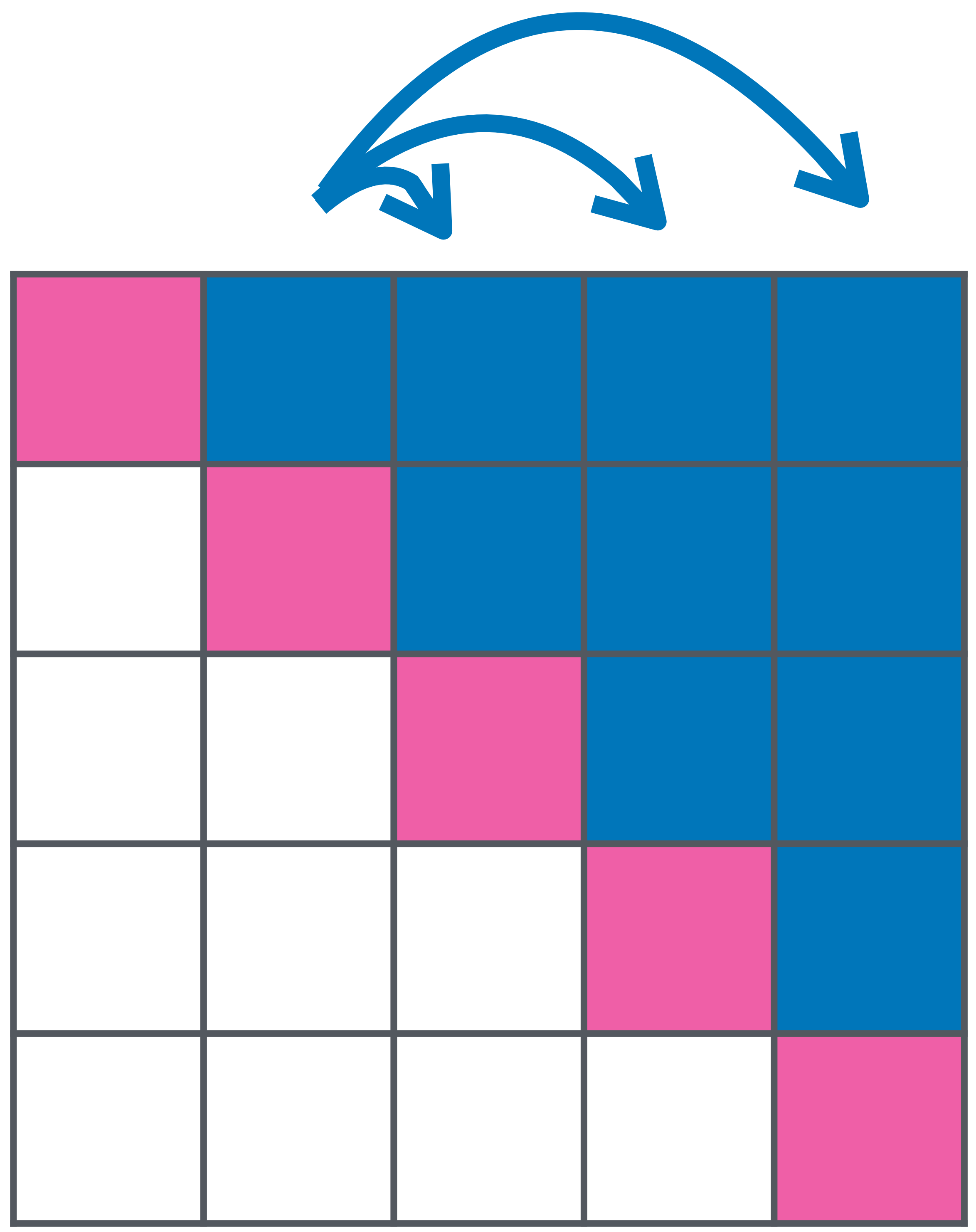
$A =$



Zaheer et al. 2017

Examples

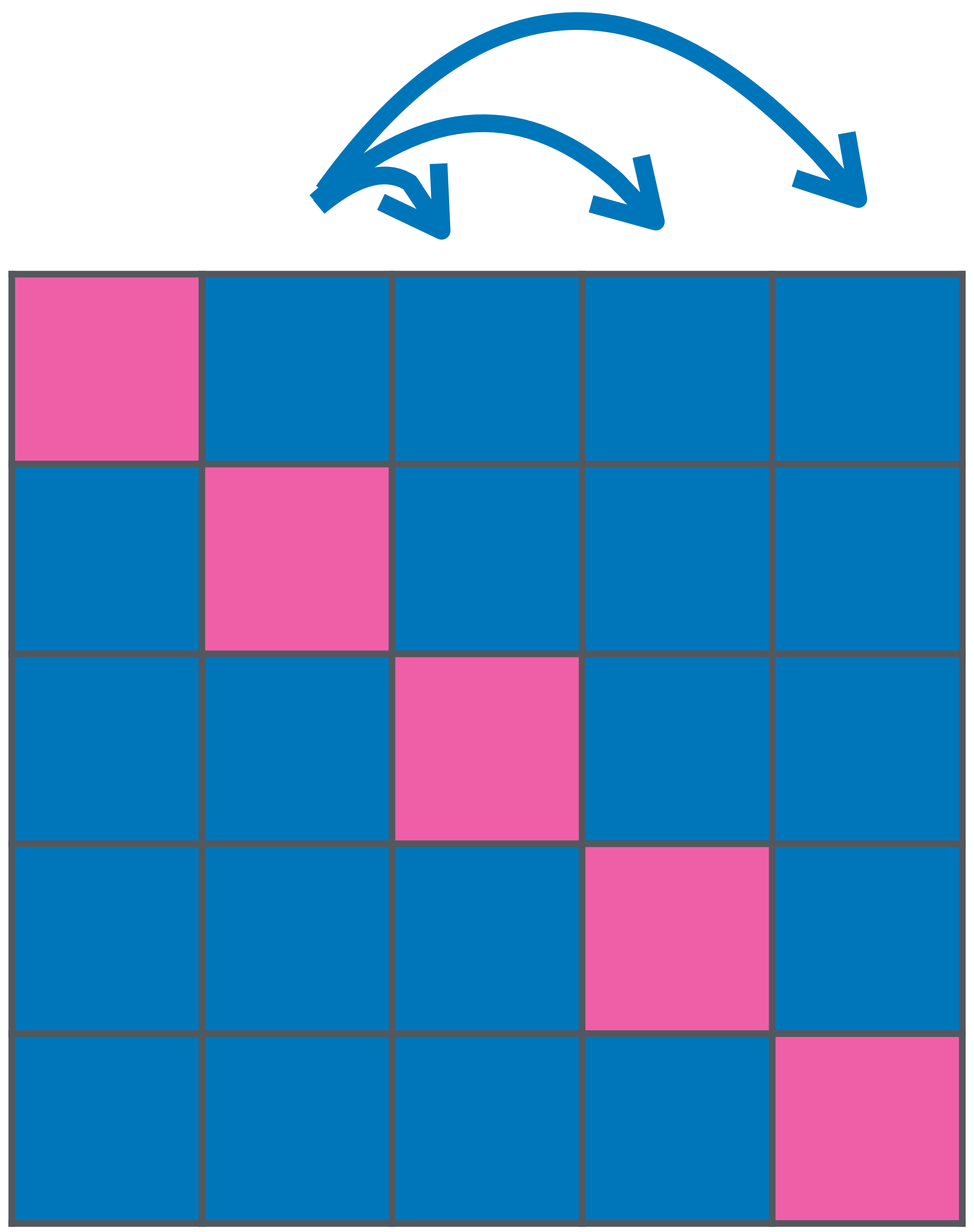
- General permutations $H = S_n$

$A =$ 

Zaheer et al. 2017

Examples

- General permutations $H = S_n$
- 2 parameters!

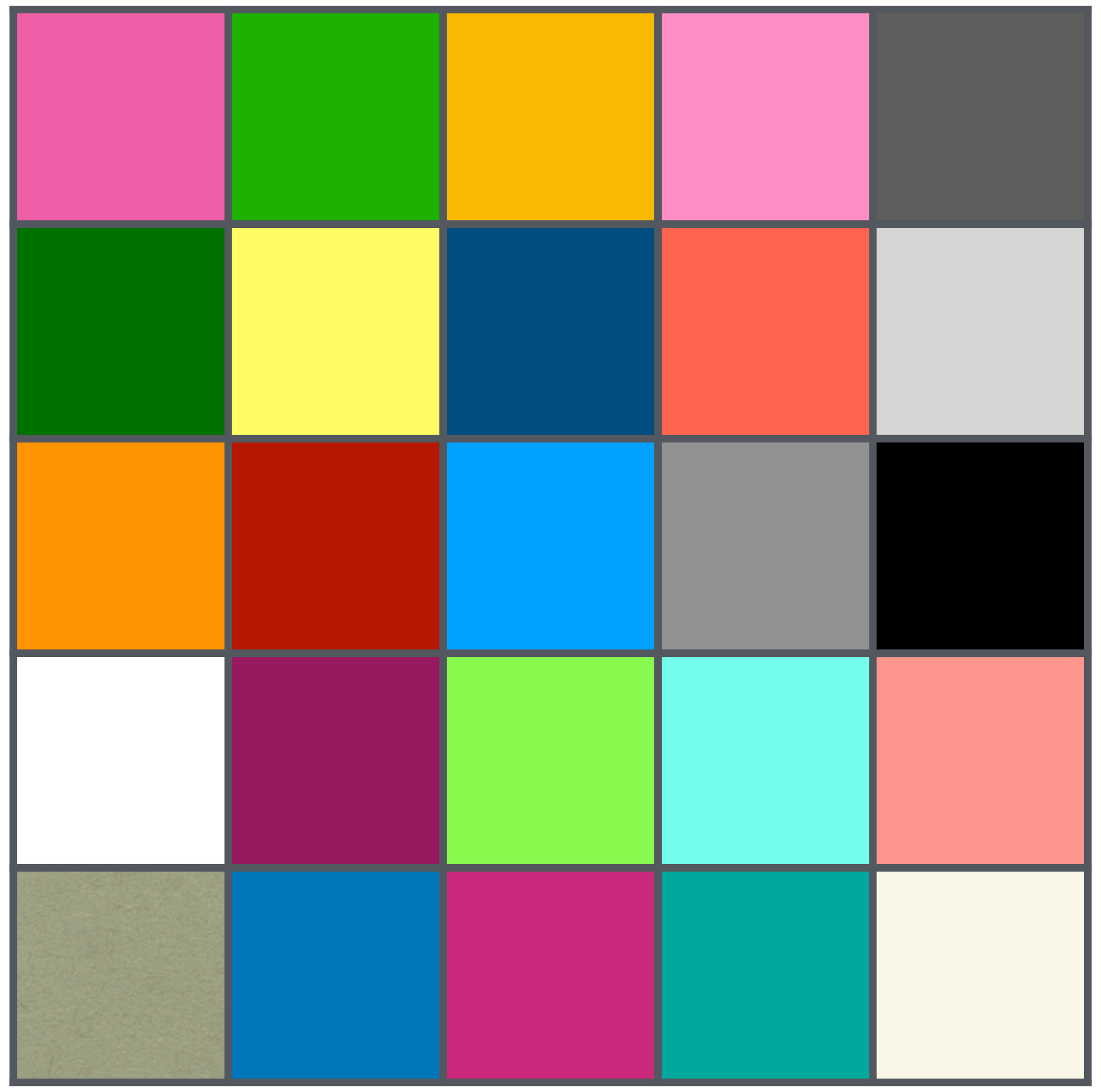
$A =$ 

Zaheer et al. 2017

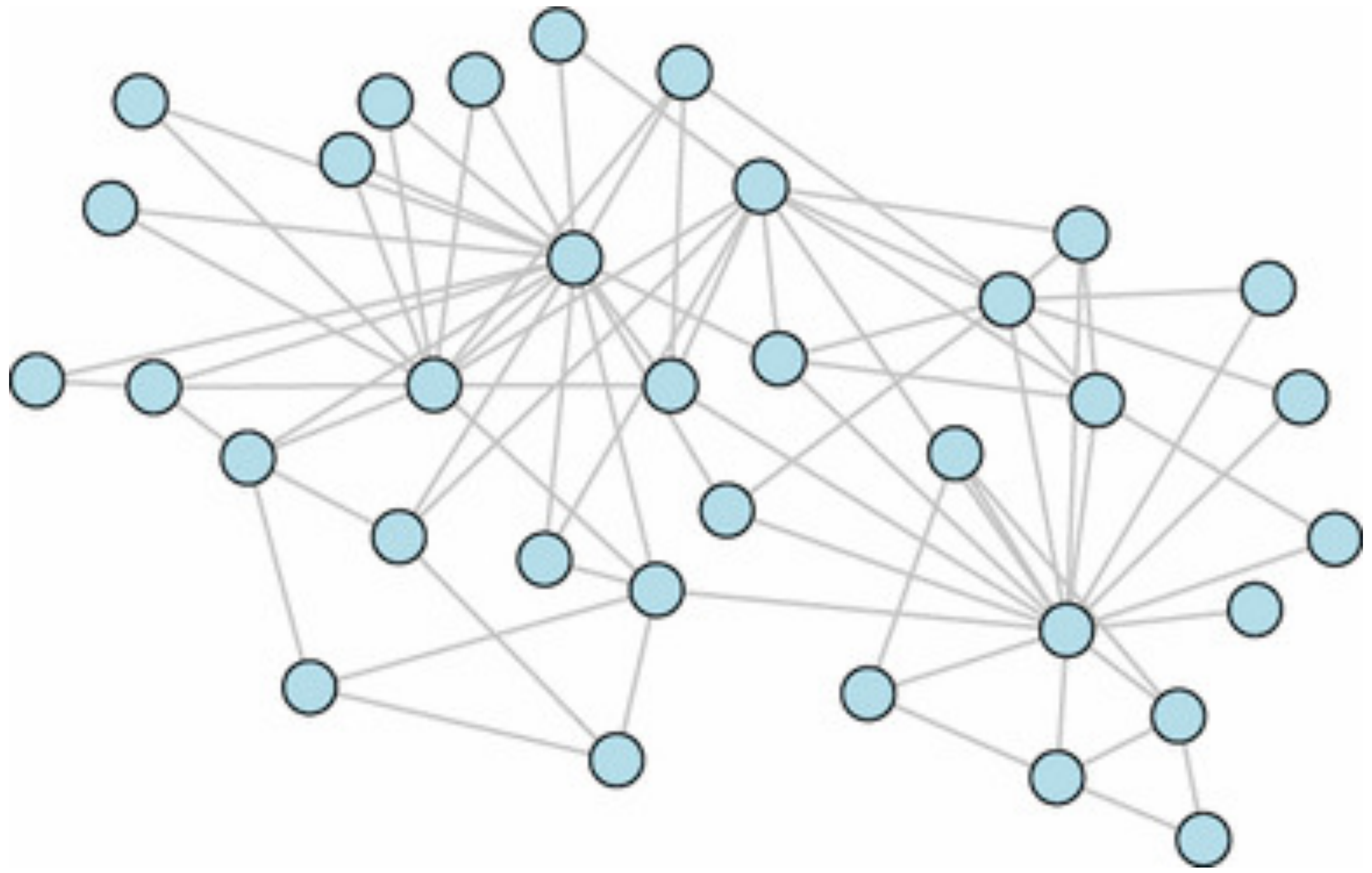
Examples

- $H = \{id\}$
- n^2 parameters!
- No parameter sharing

$A =$

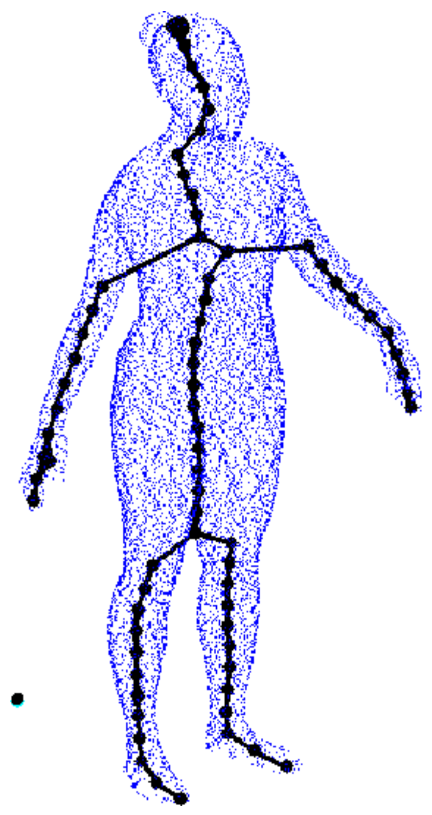


Parameter-sharing for learning graphs

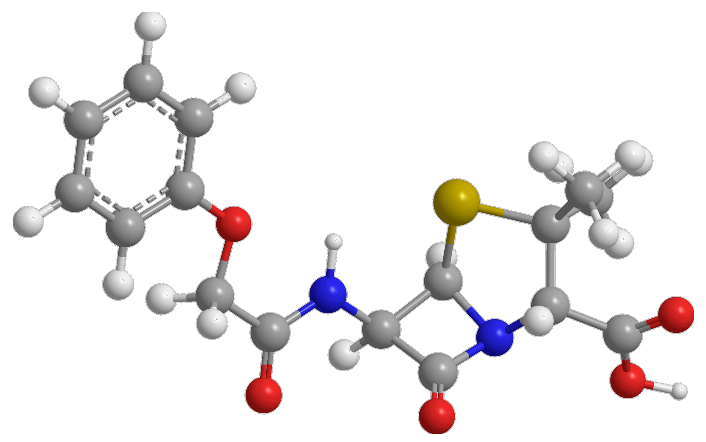


Learning graphs

- 3D Shapes

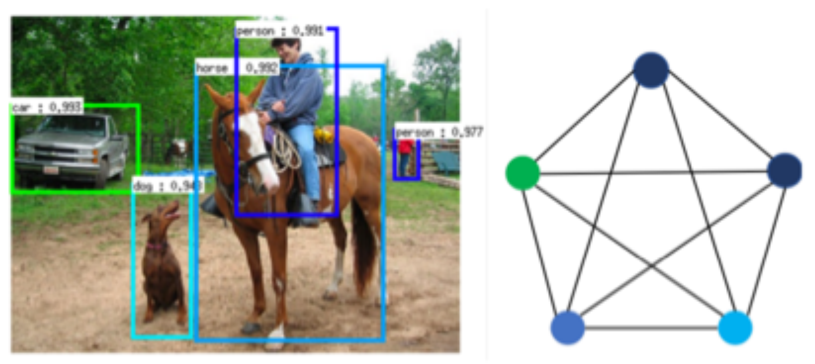


- Molecules and chemical compounds

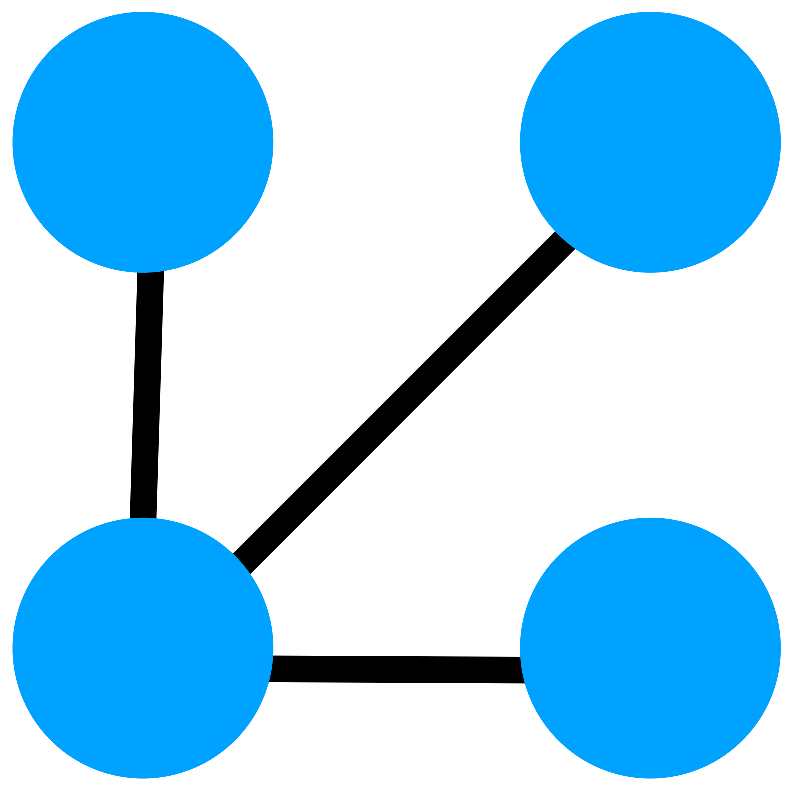


- Social Networks

- Scenes in images

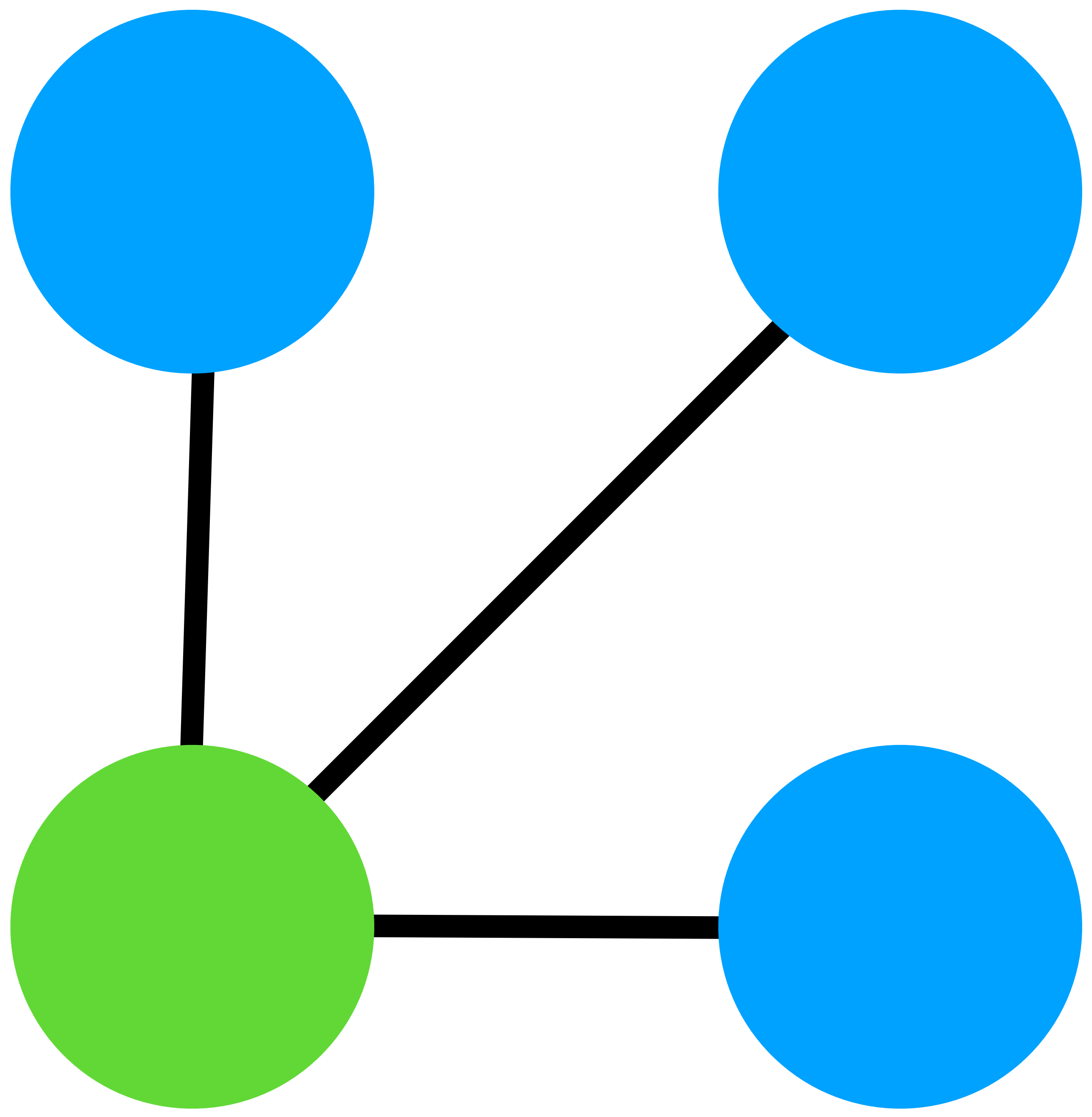


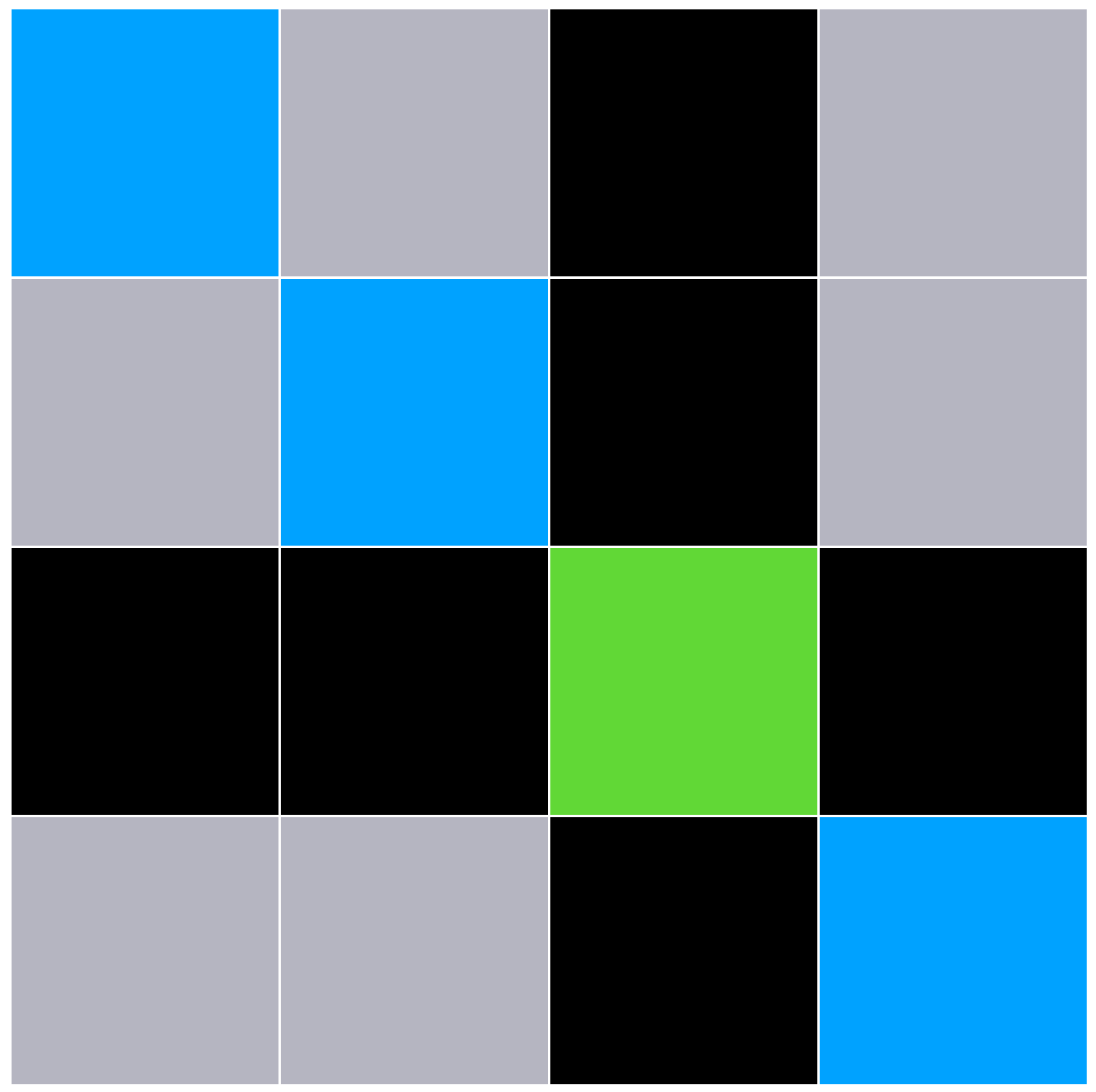
Supervised learning for graphs

$f\Bigg($  $\Bigg) \approx$ ‘label’

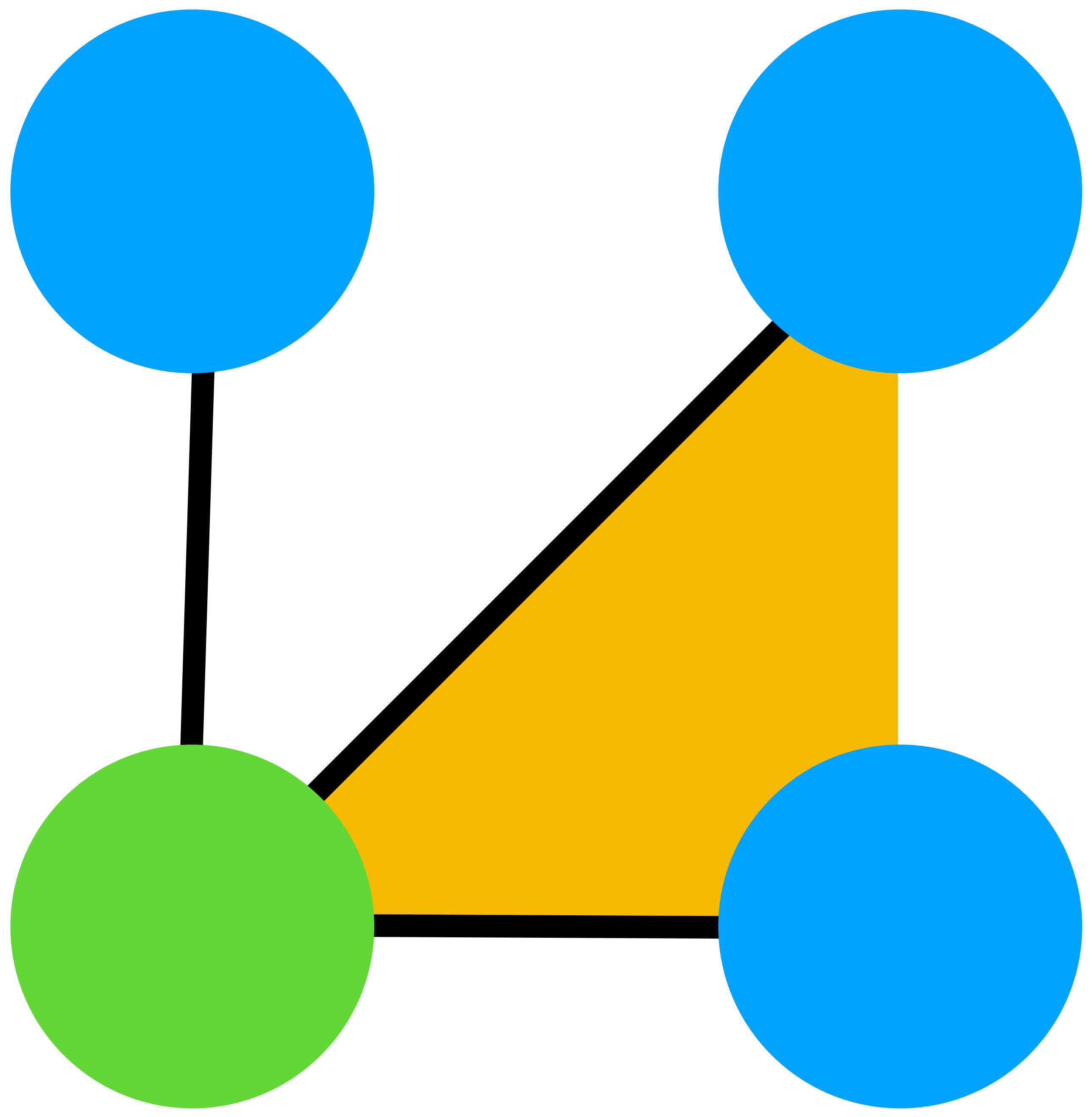
graphs as matrices

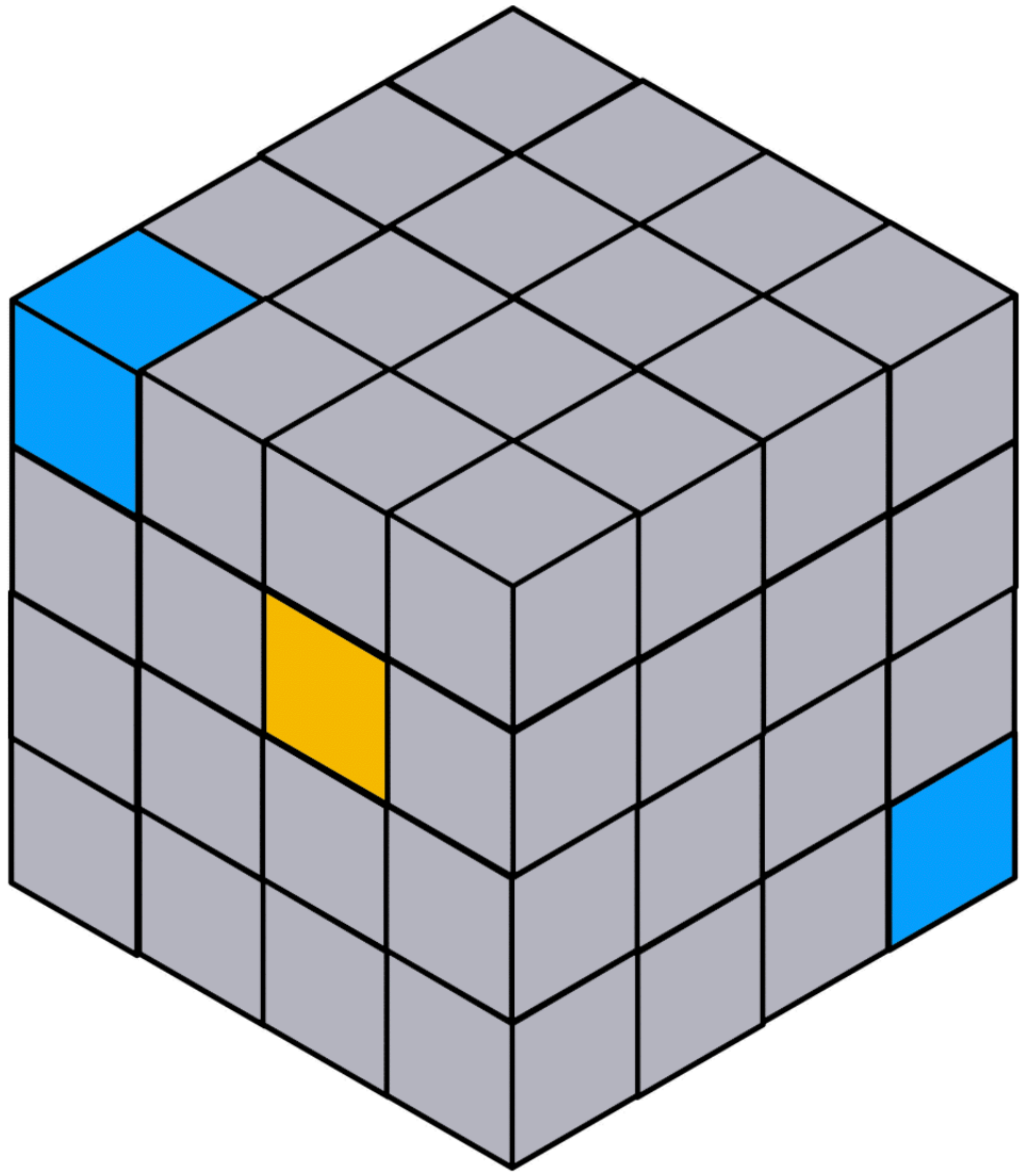
- Graphs:



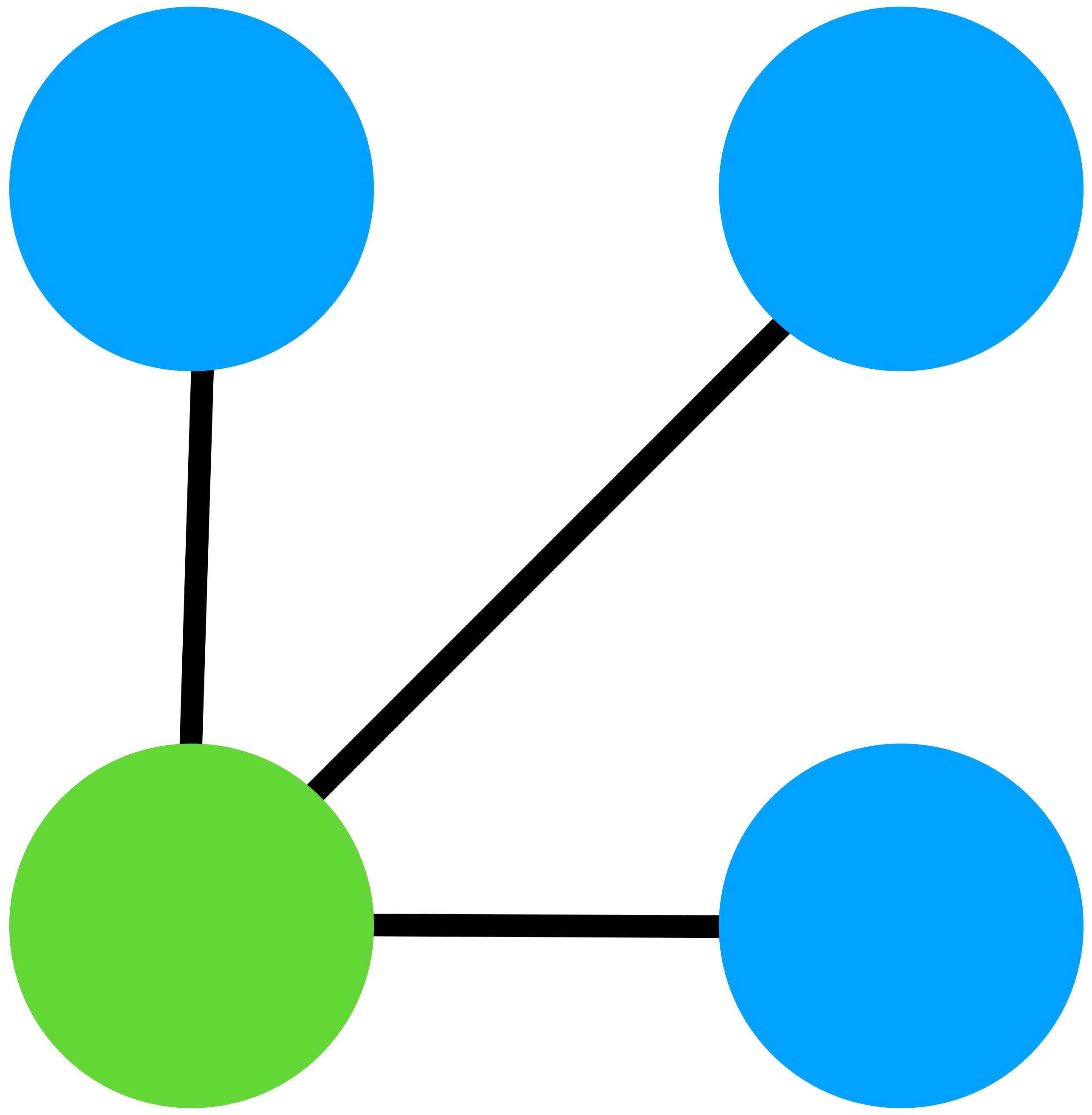


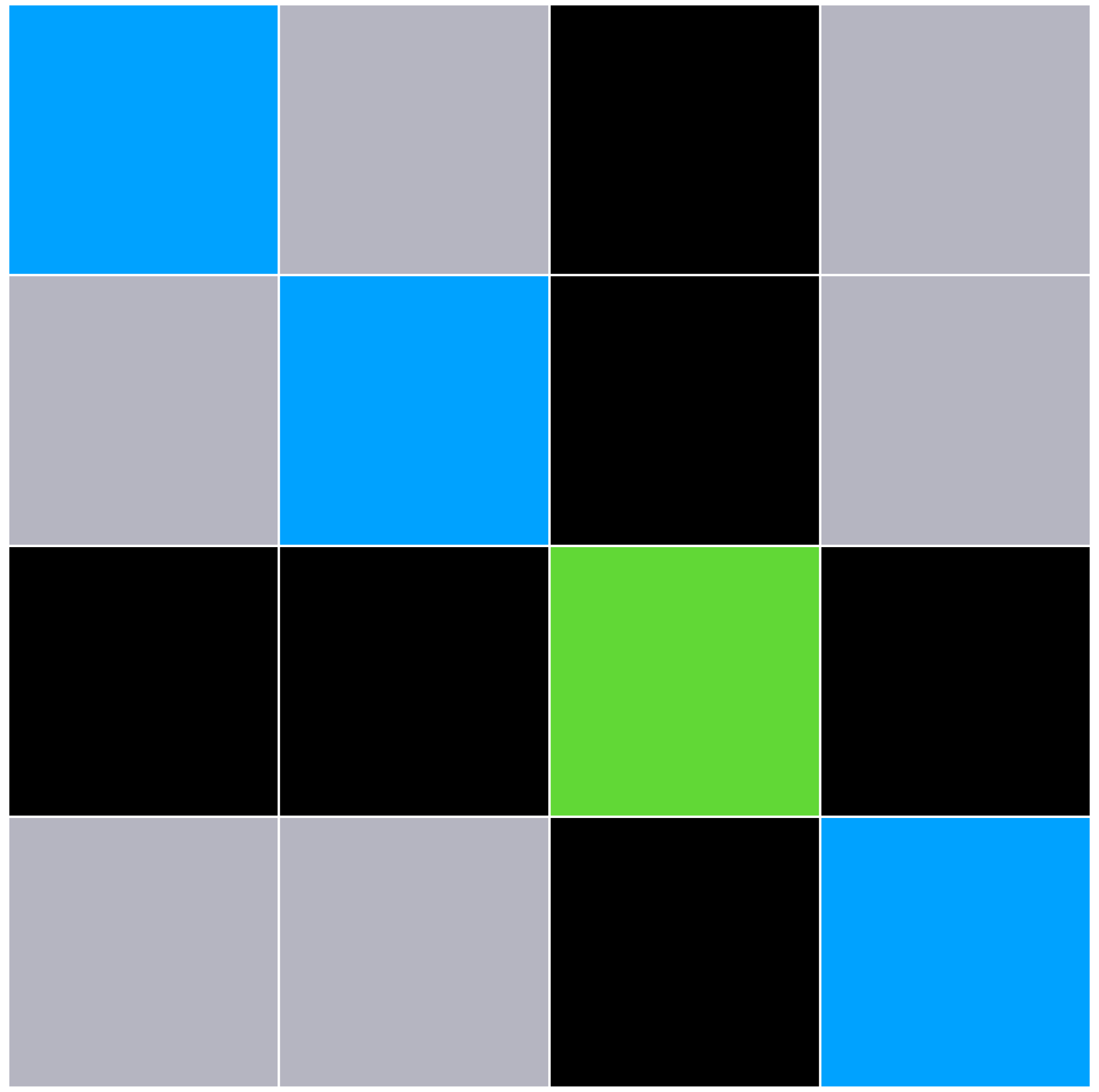
graphs and hyper graphs as tensors



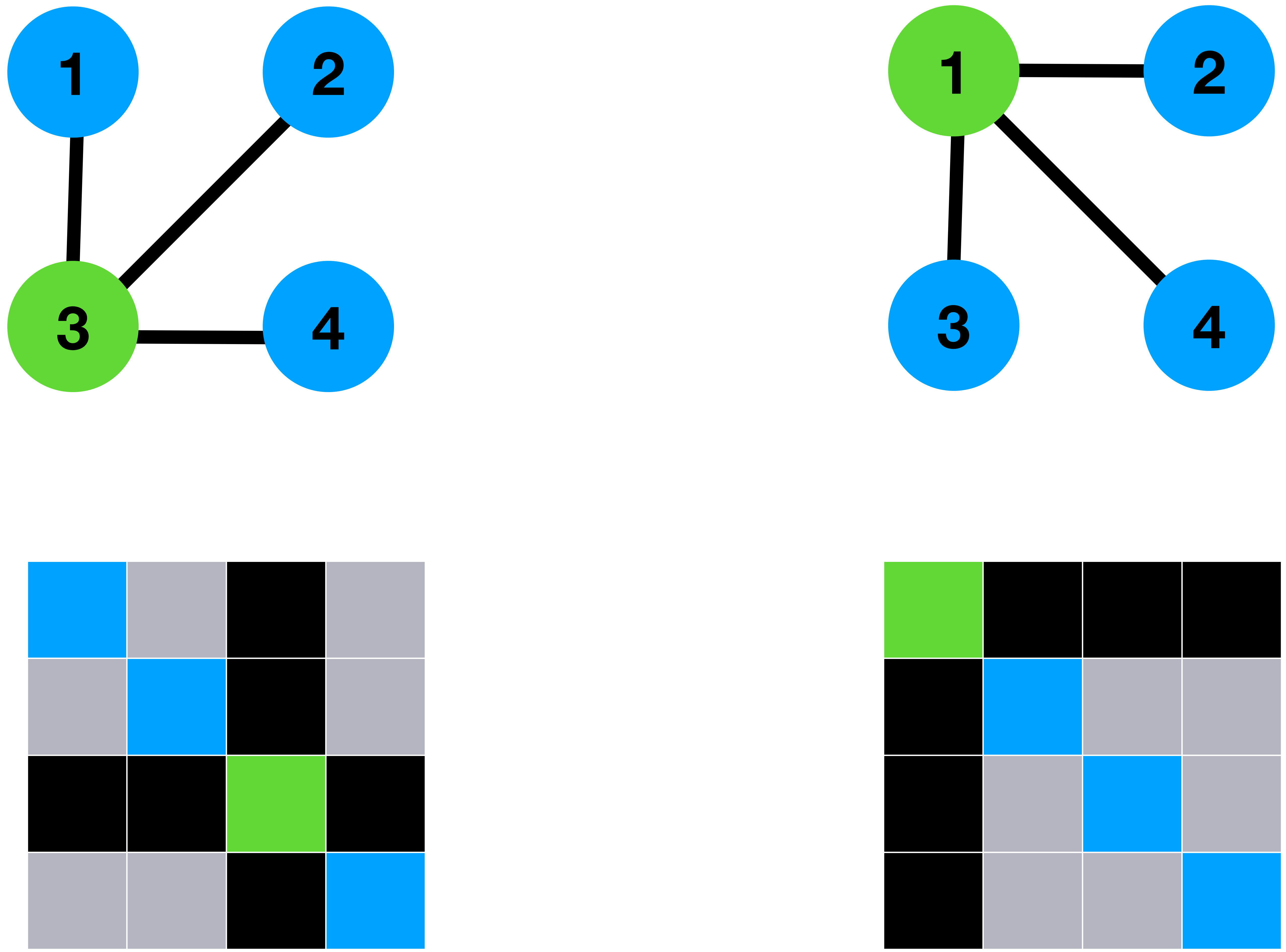


Graph symmetries

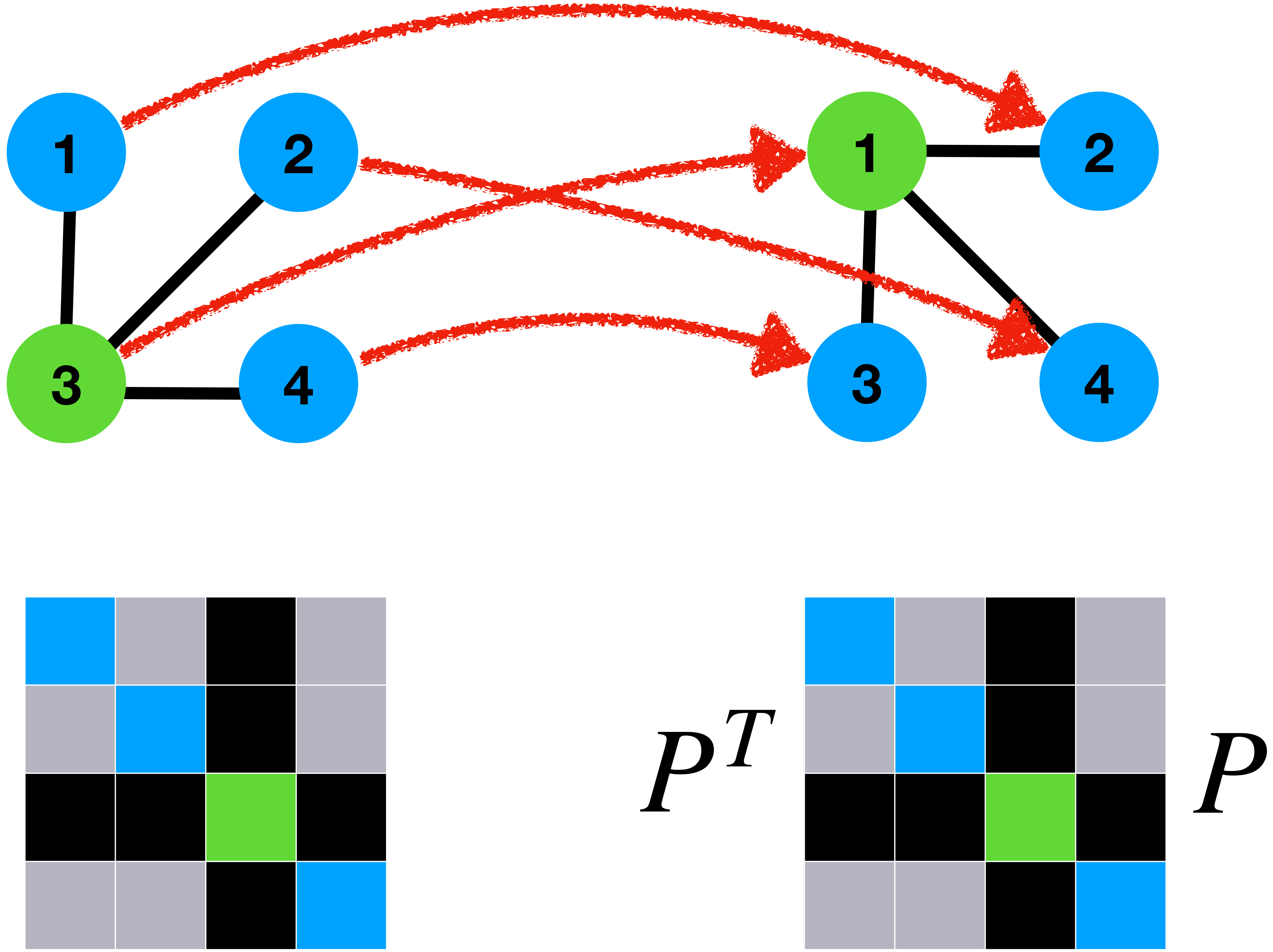




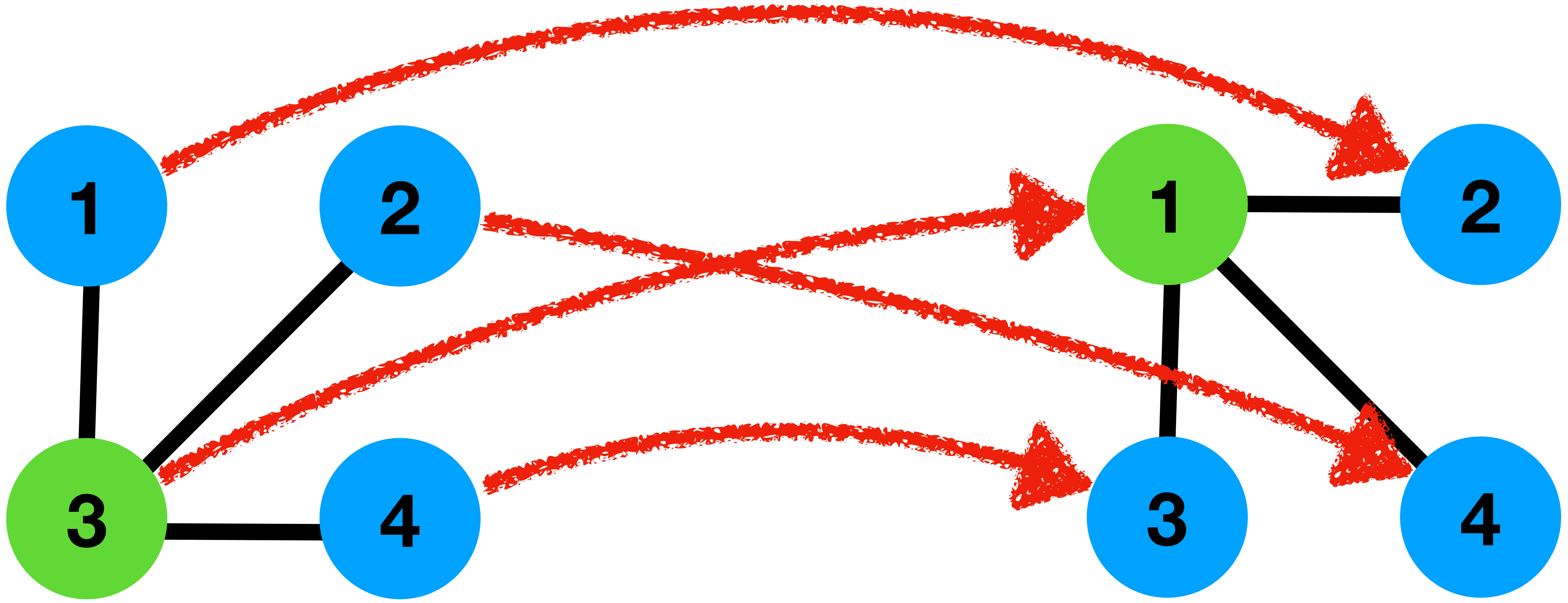
Graph symmetries

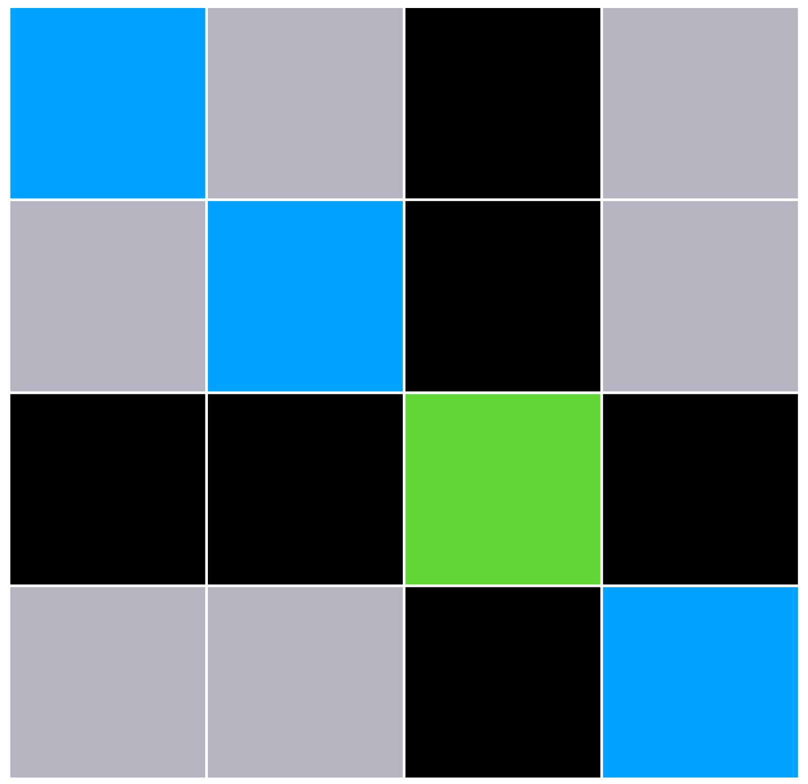


Graph symmetries

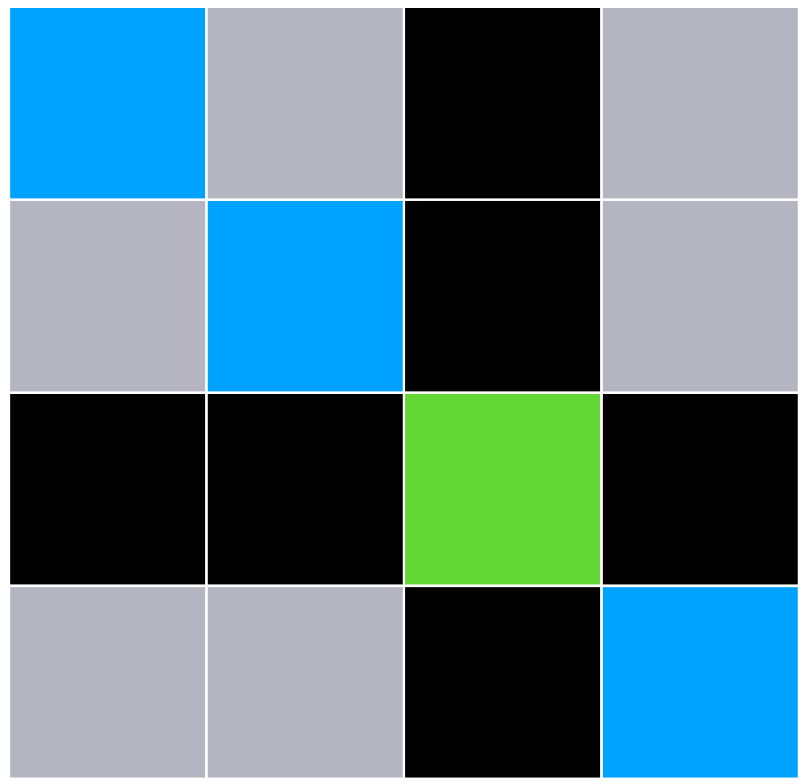


Graph symmetries



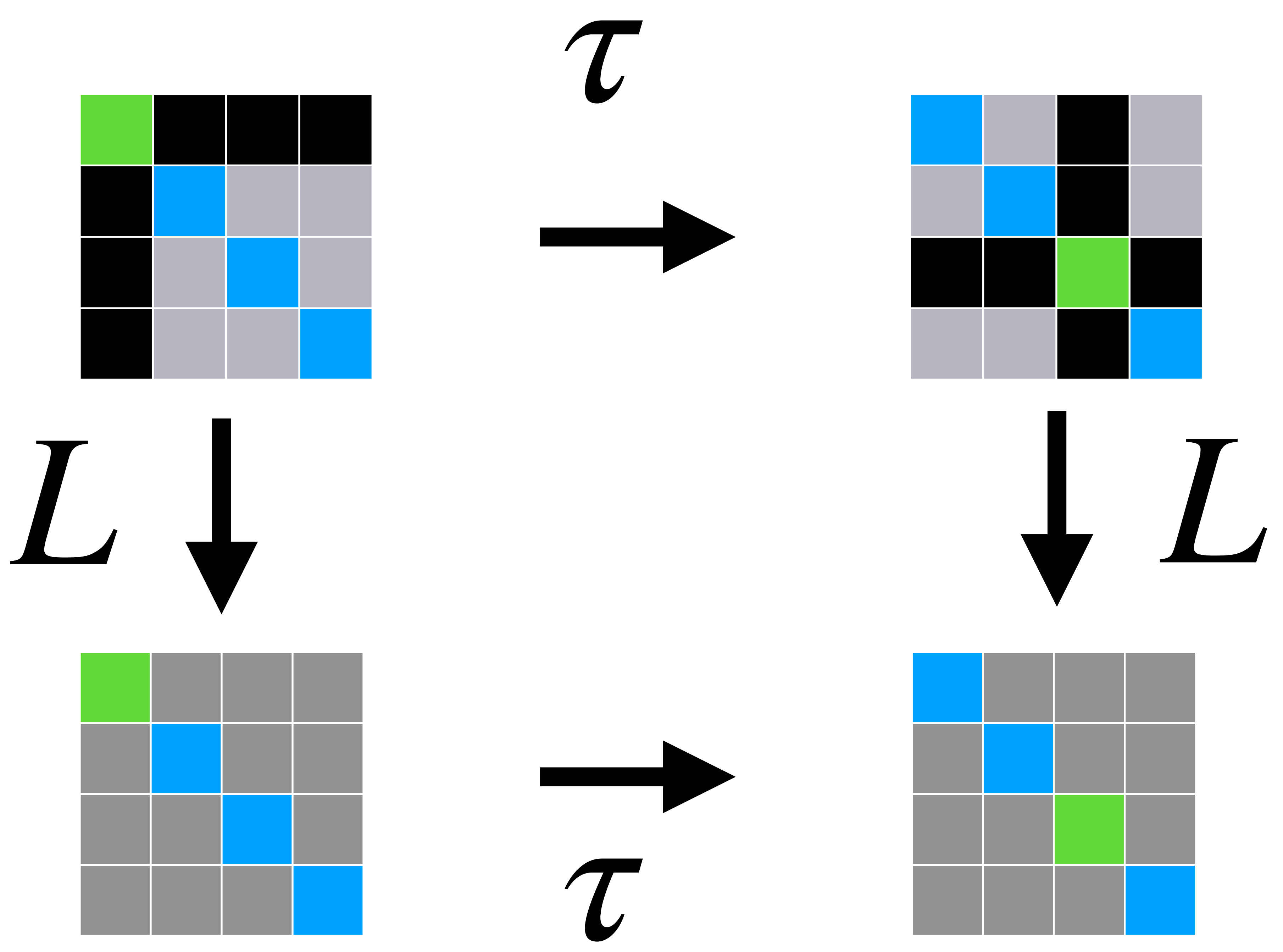


$\tau \cdot$



Equivariance in the graph case

- $H = S_n \leq S_{n^2}$



$$\tau \cdot L(\mathbf{X}) = L(\tau \cdot \mathbf{X})$$

Parameter-sharing

- 15 parameters
- Independent of graphs size
- Here $n = 5$

n^2

n^2

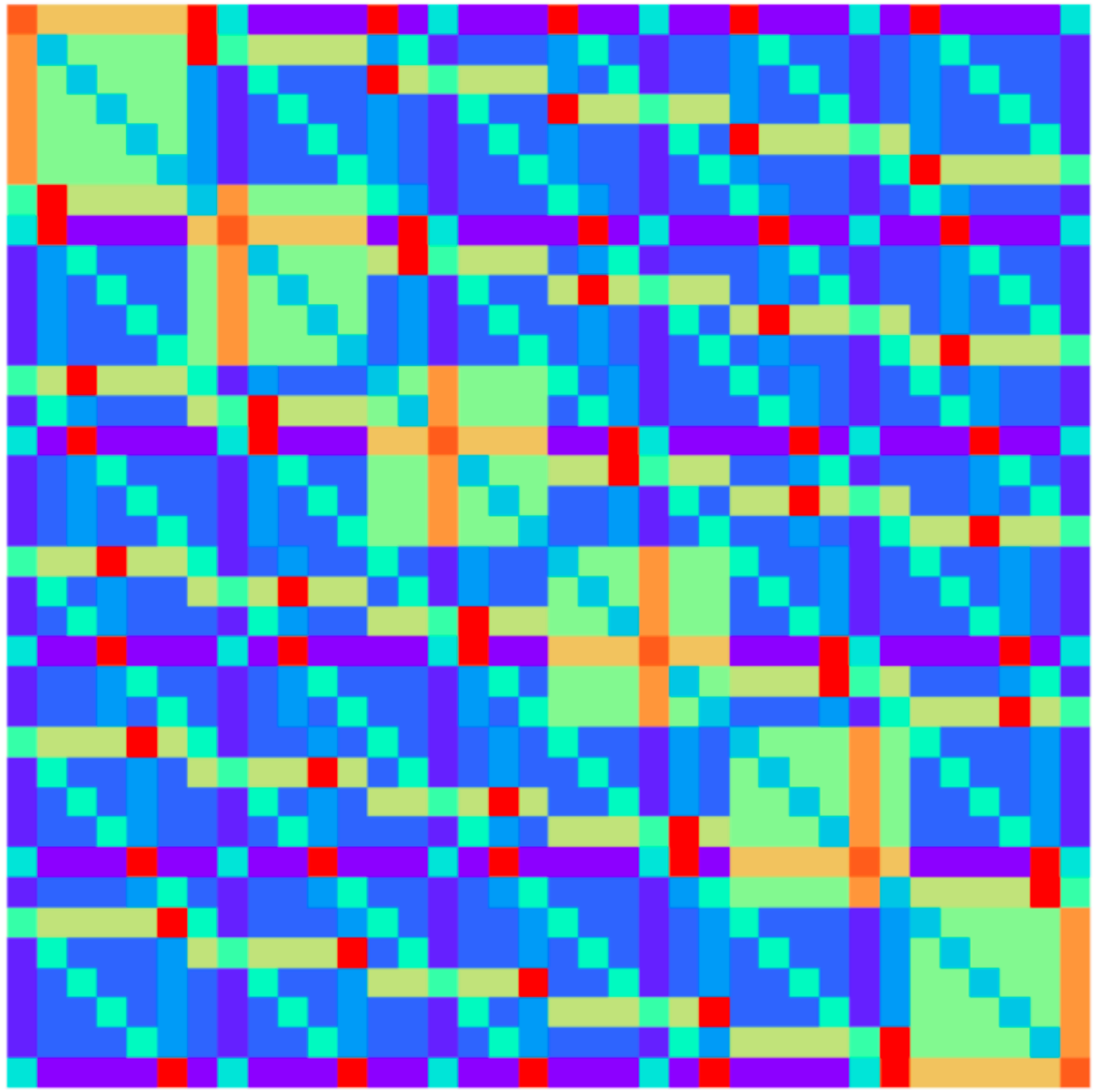
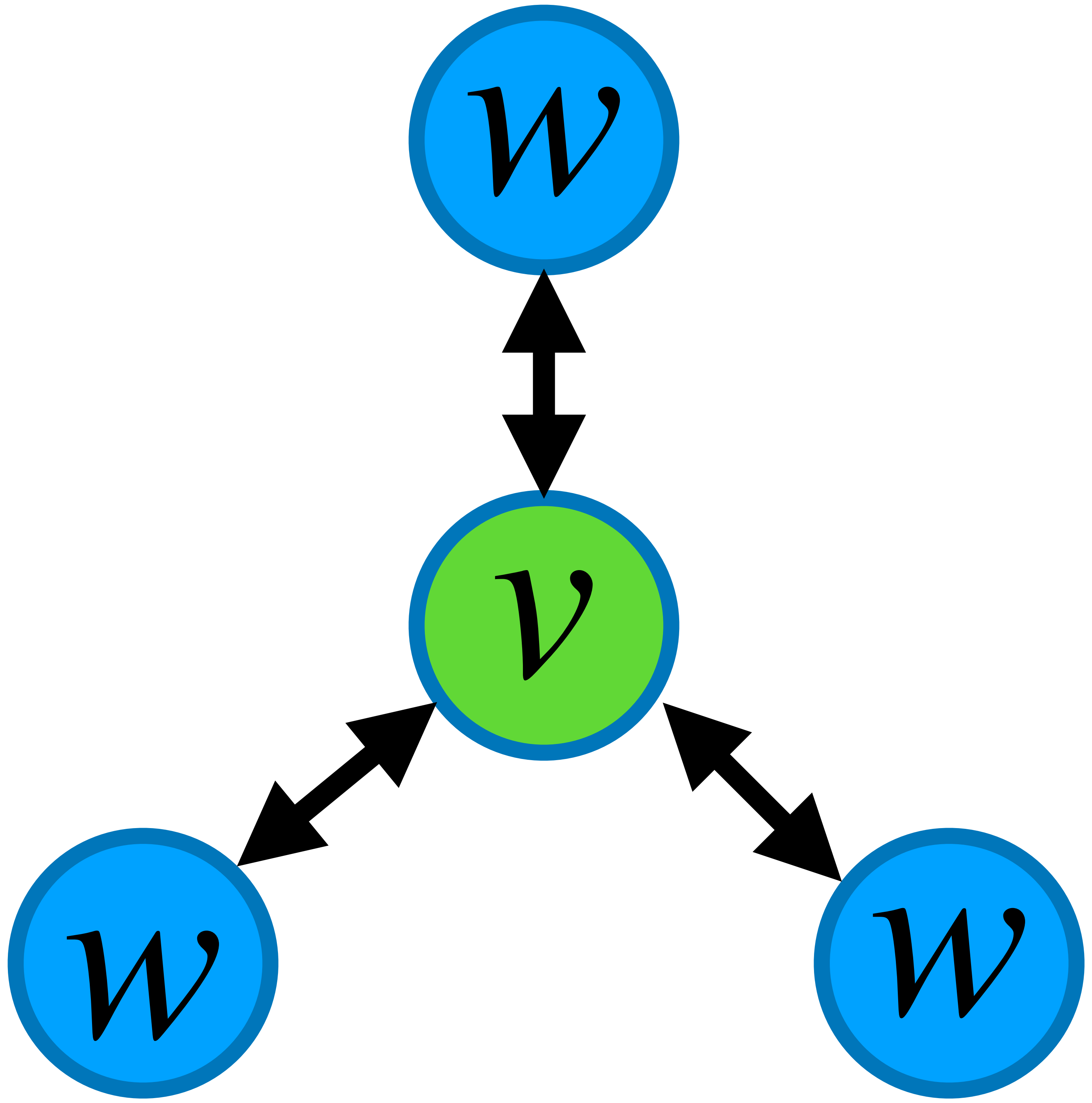


Figure taken from Siamak Ravanbakhsh

Expressive power

- **Theorem:** Can represent message-passing neural networks to arbitrary precision

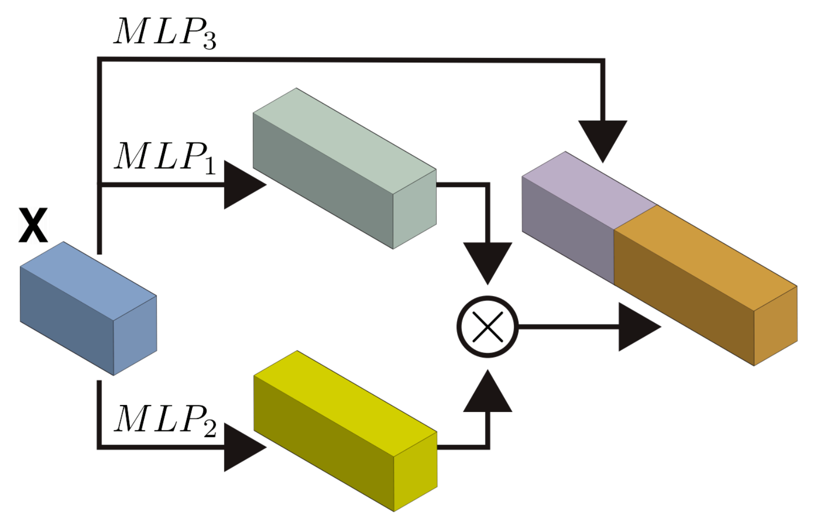


$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

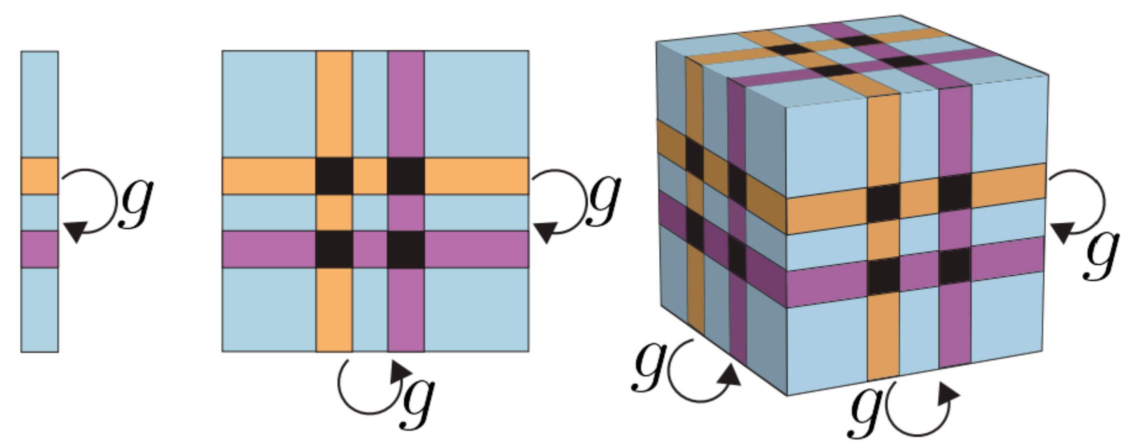
More expressive graph networks

- Polynomial layers improve expressivity



M. et al., NeurIPS 2019

- High-order tensors improve expressivity



M. et al., ICML 2019, NeurIPS 2019

Parameter-sharing for learning sets of symmetric elements

Input

Output

M. et al., ICML 2020

Set Symmetry

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

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

Input set

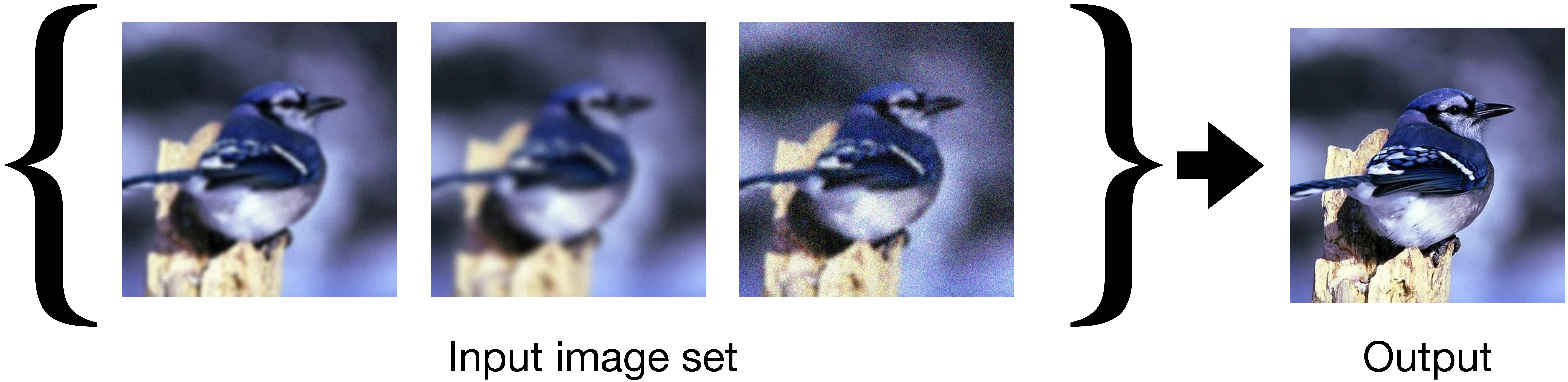
Set+Elements symmetry

Both the set and its elements have symmetries.

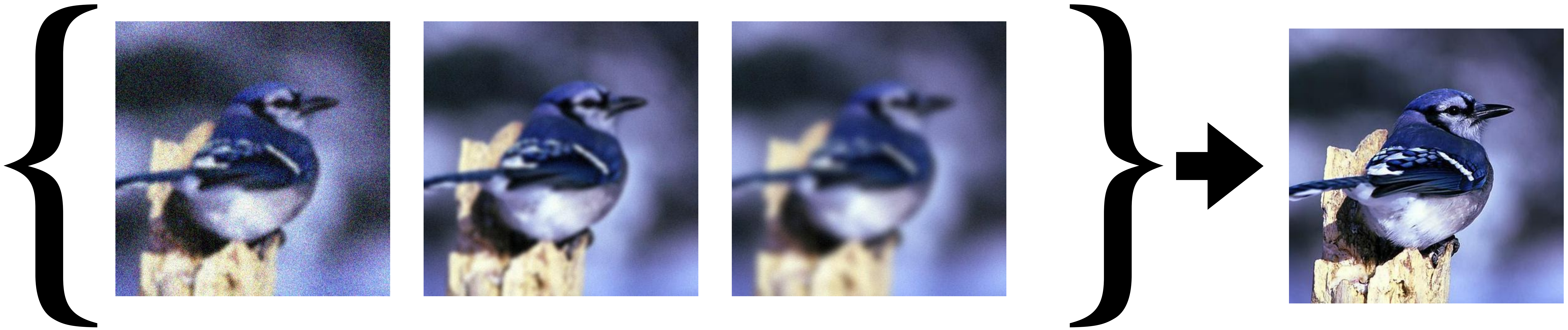
Input set

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

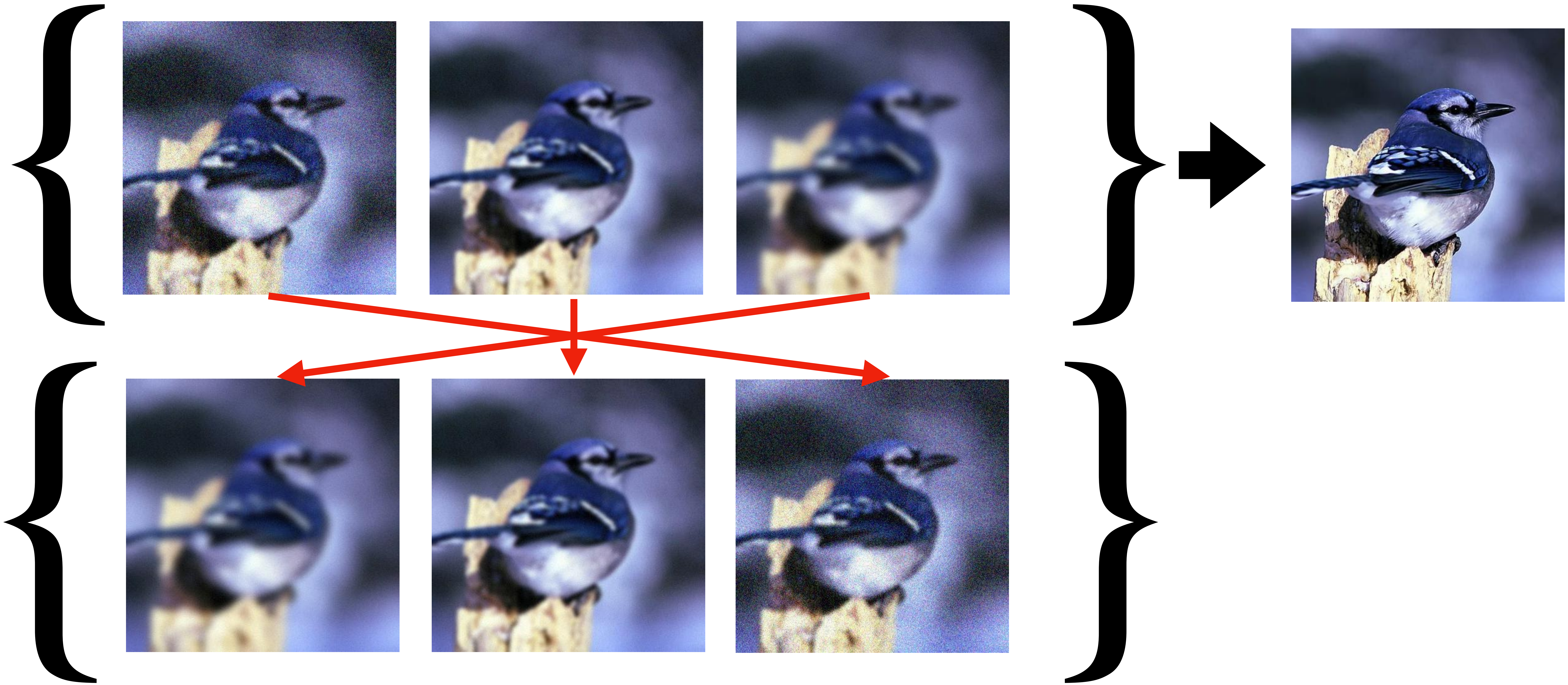
Deep Symmetric sets



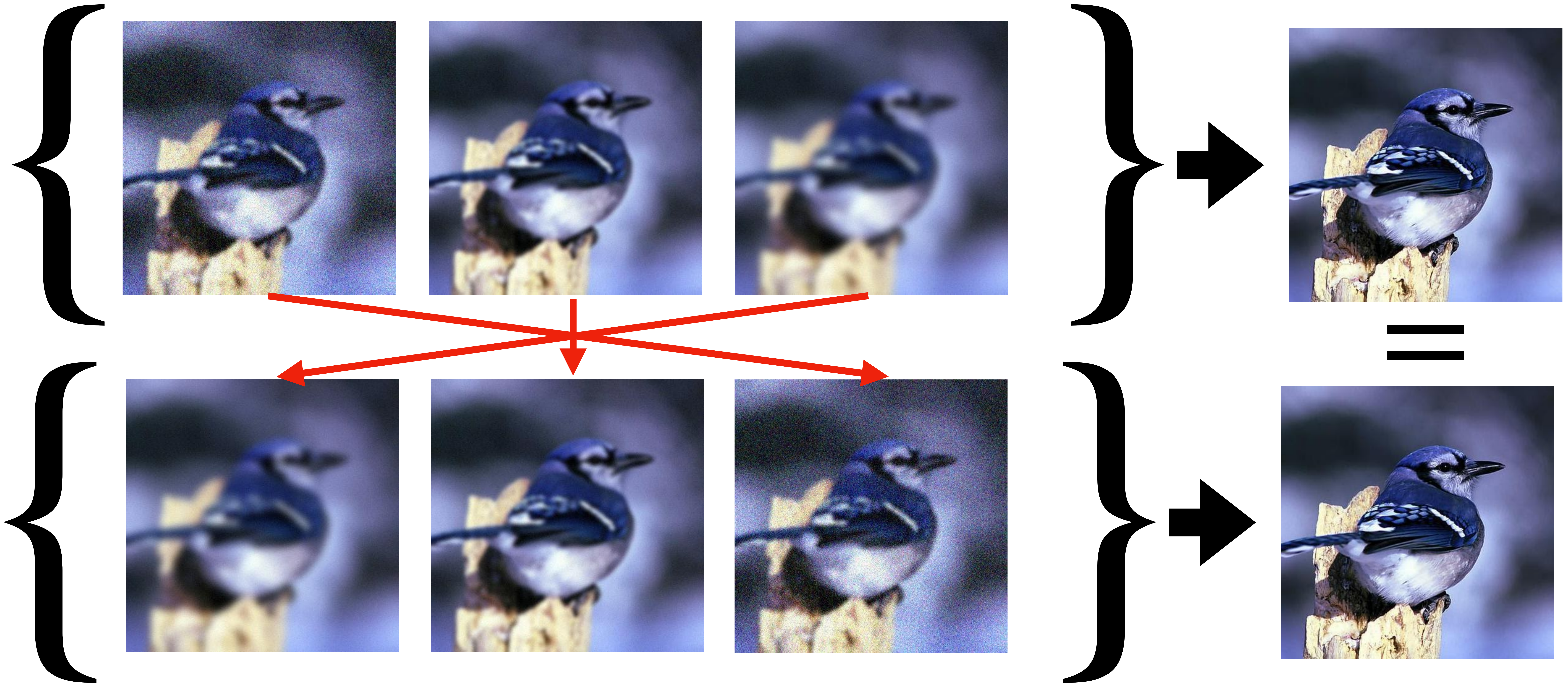
Set symmetry:
Order invariance/equivariance



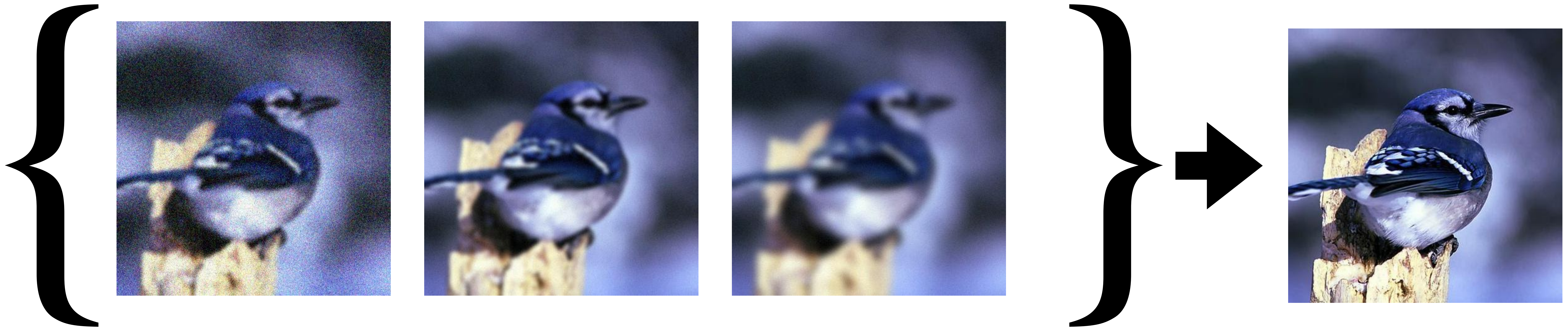
Set symmetry:
Order invariance/equivariance



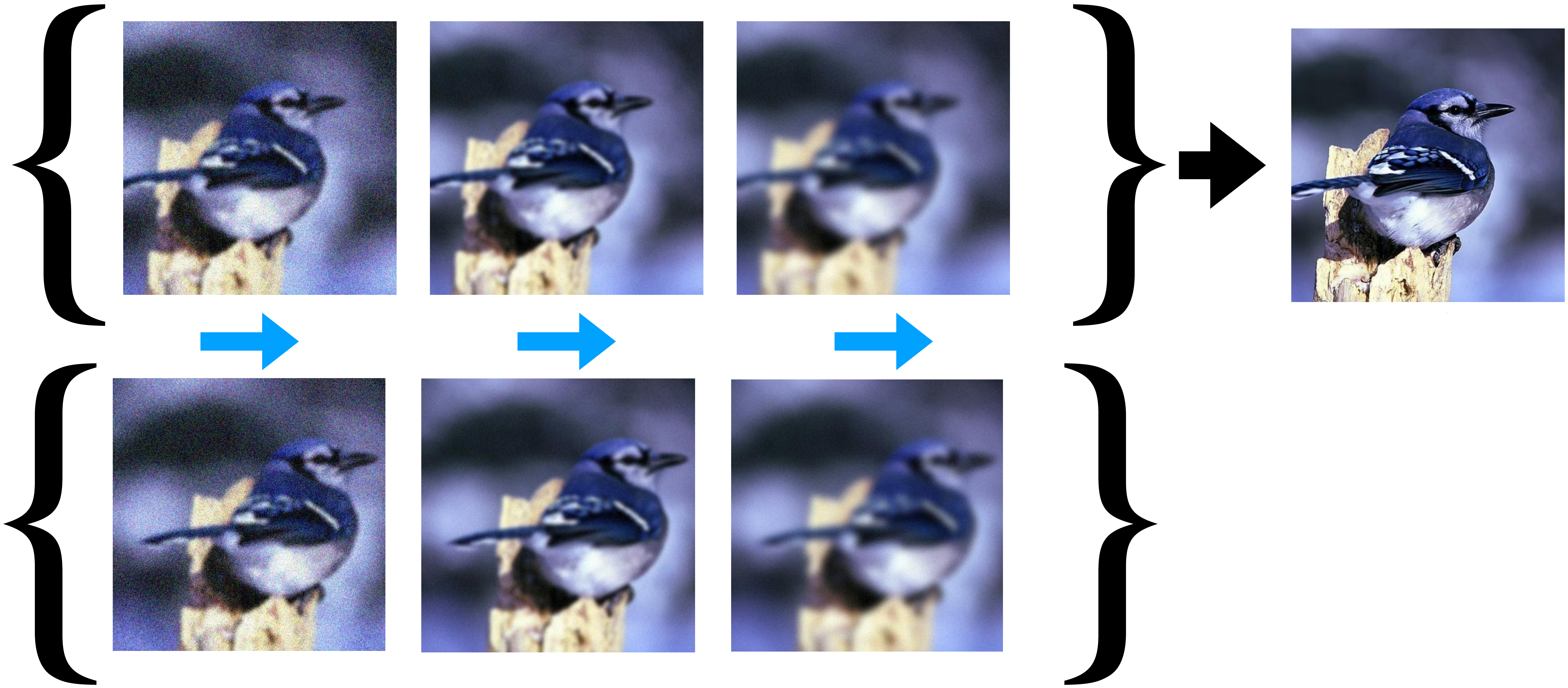
Set symmetry:
Order invariance/equivariance



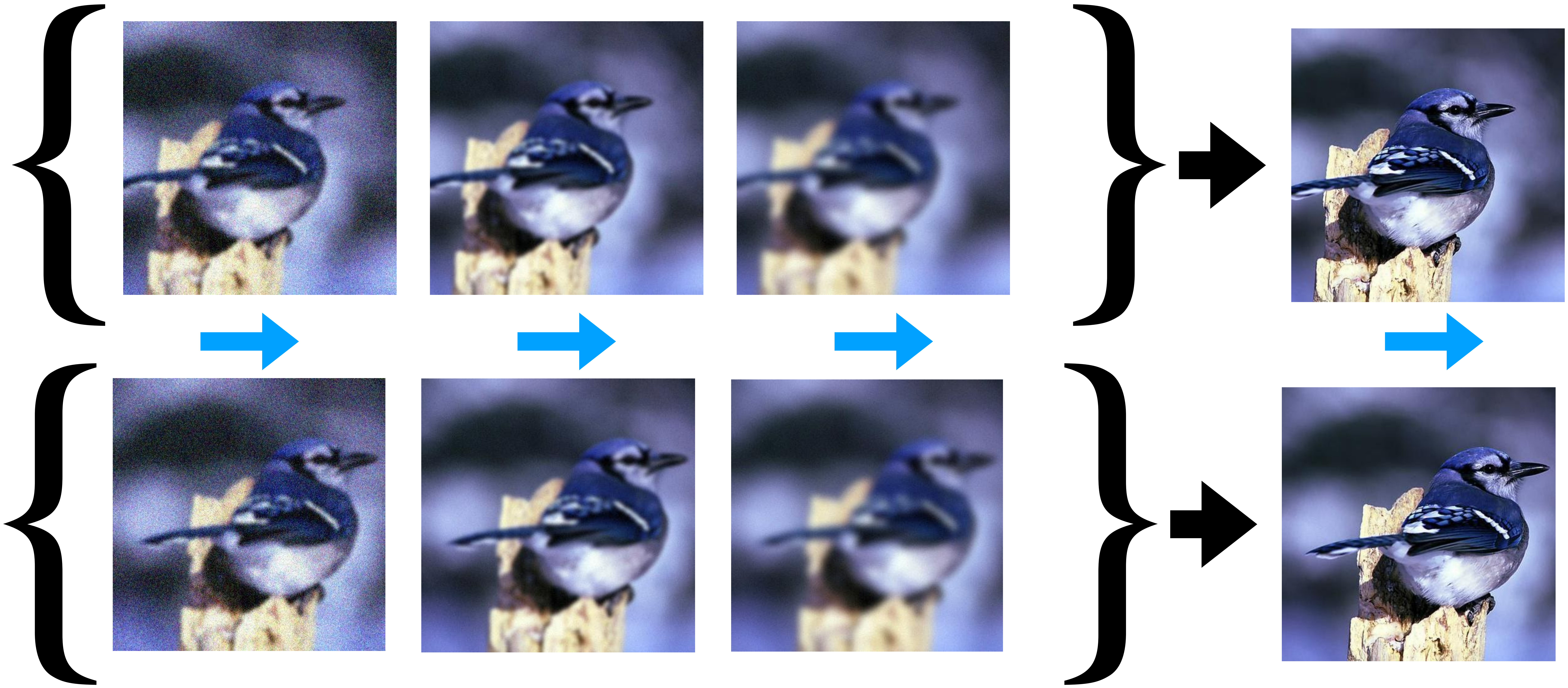
Element symmetry:
Translation invariance/equivariance



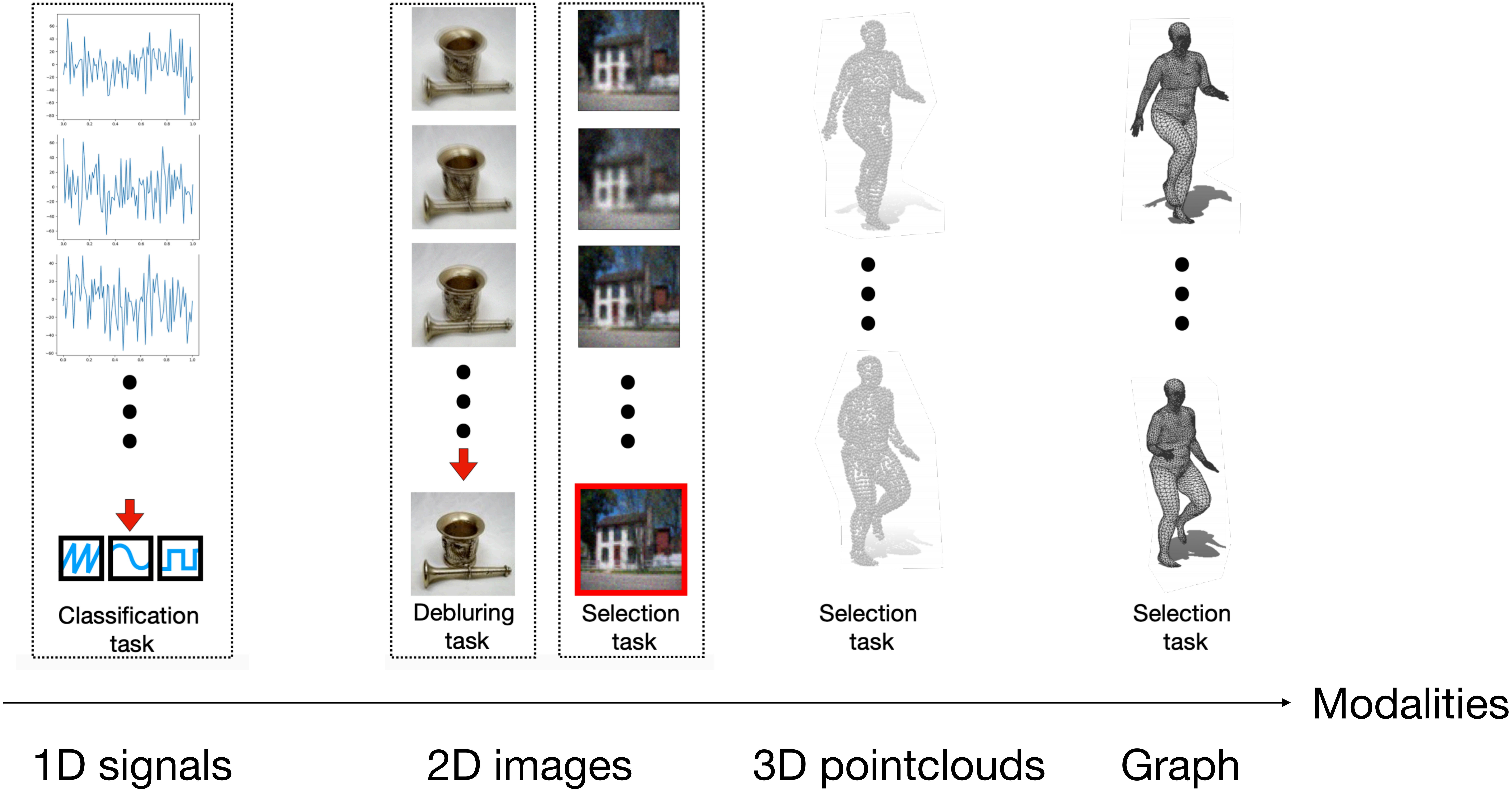
Element symmetry:
Translation invariance/equivariance



Element symmetry: Translation invariance/equivariance



Applications

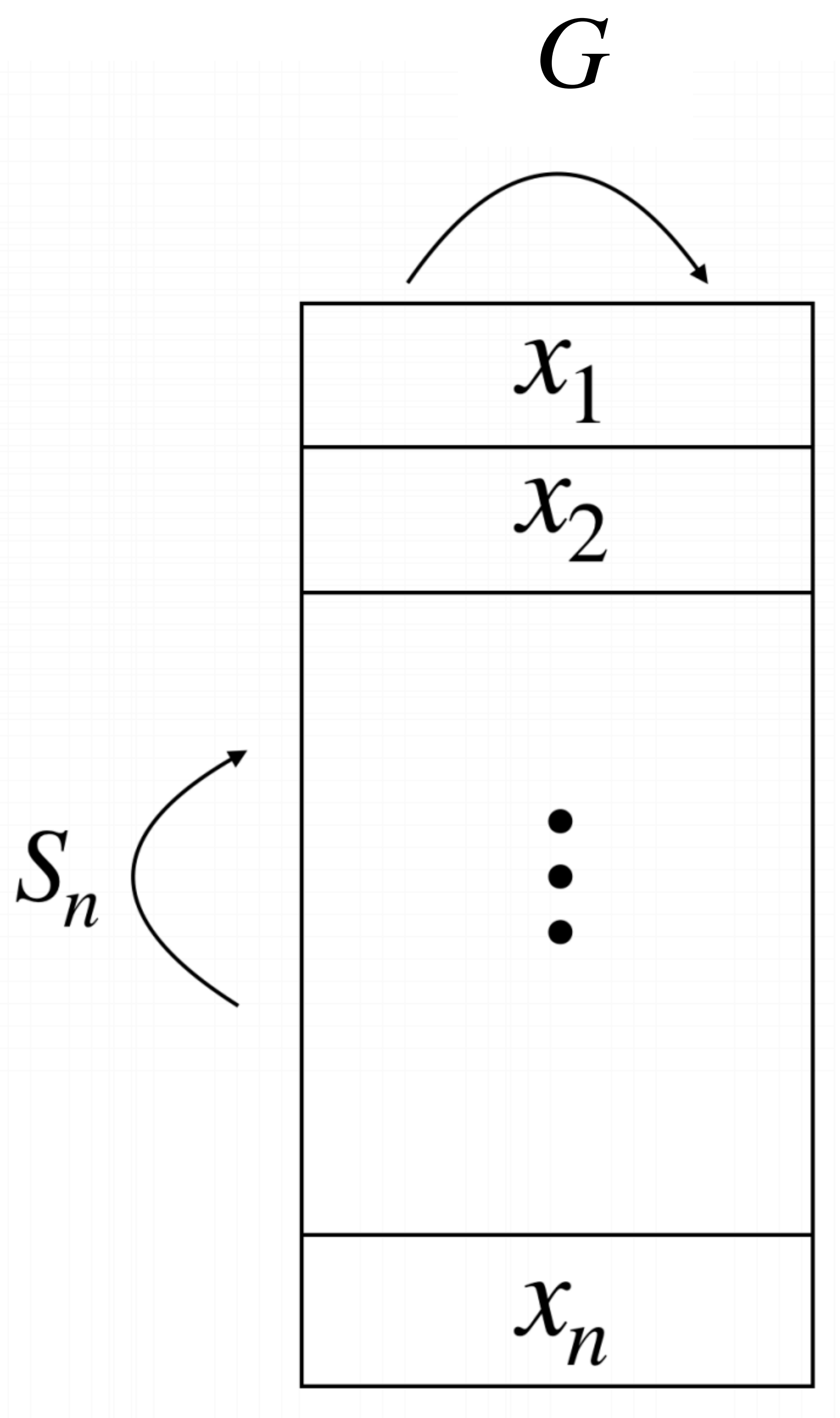


Setup

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

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

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



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

$$L(X)_i = L_1^G(x_i) + \sum_{j=1}^{n} L_2^G(x_j)$$

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

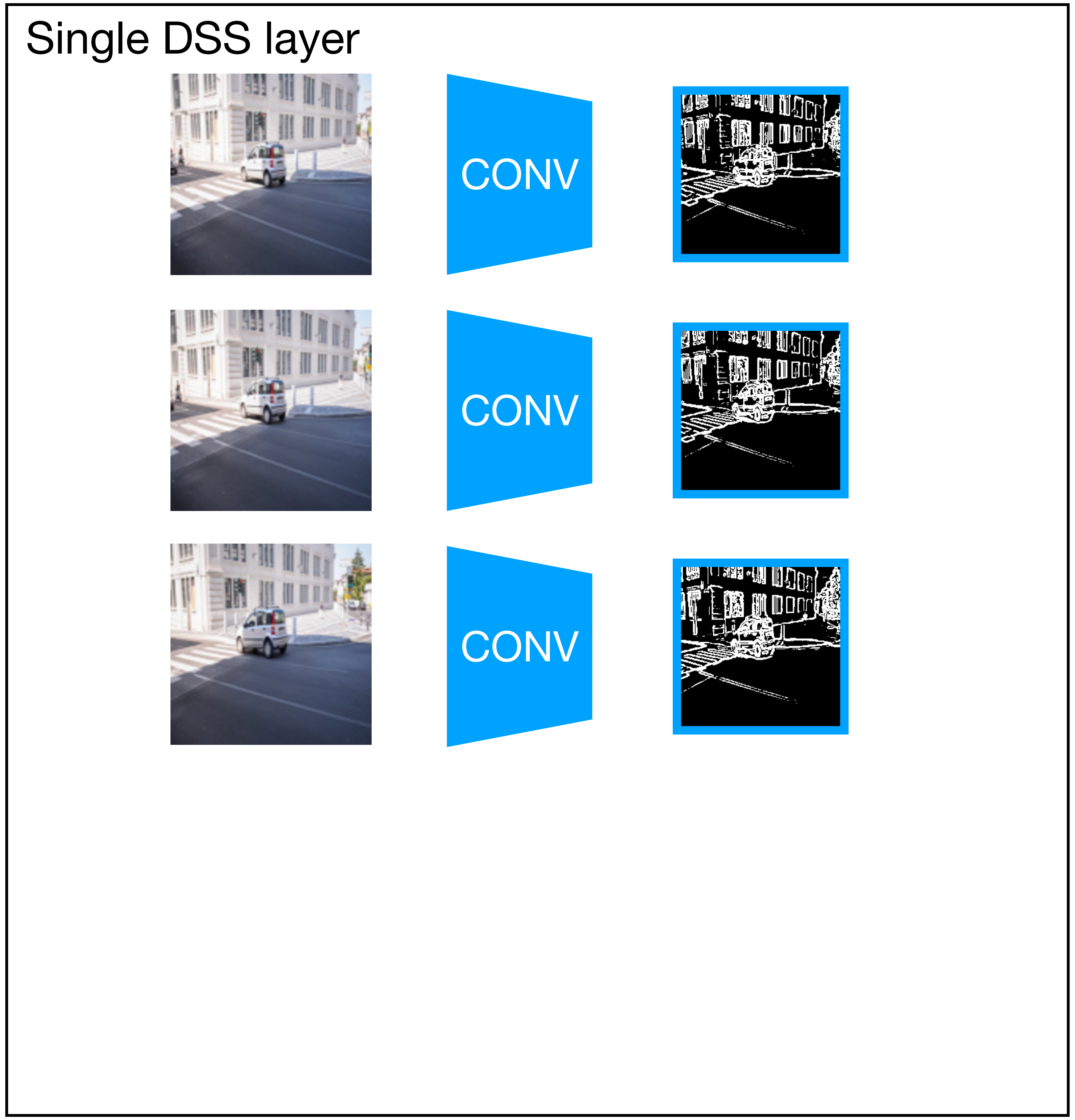
Single DSS layer

DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

G-equivariant layers are convolutions

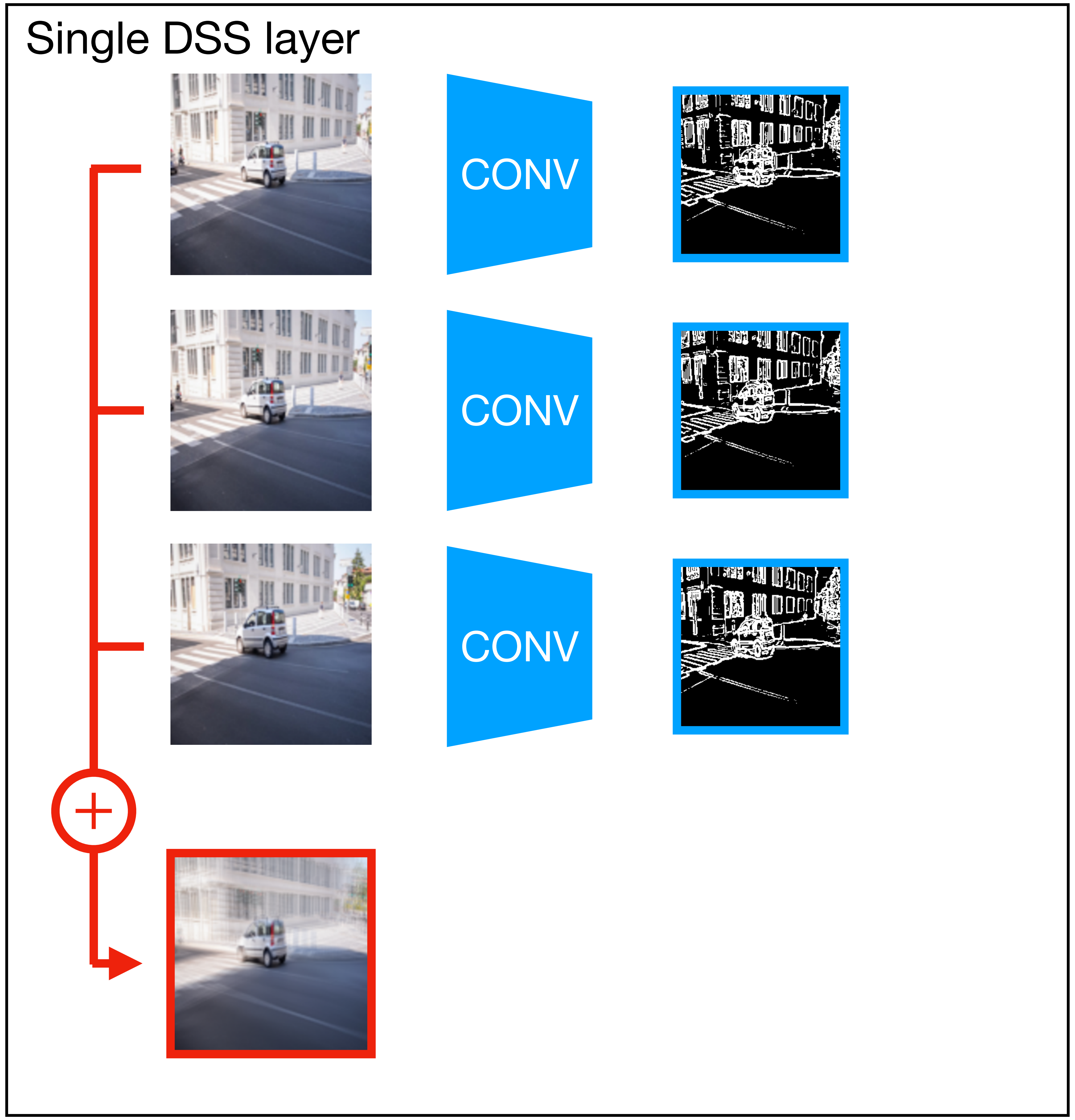


DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

G-equivariant layers are convolutions

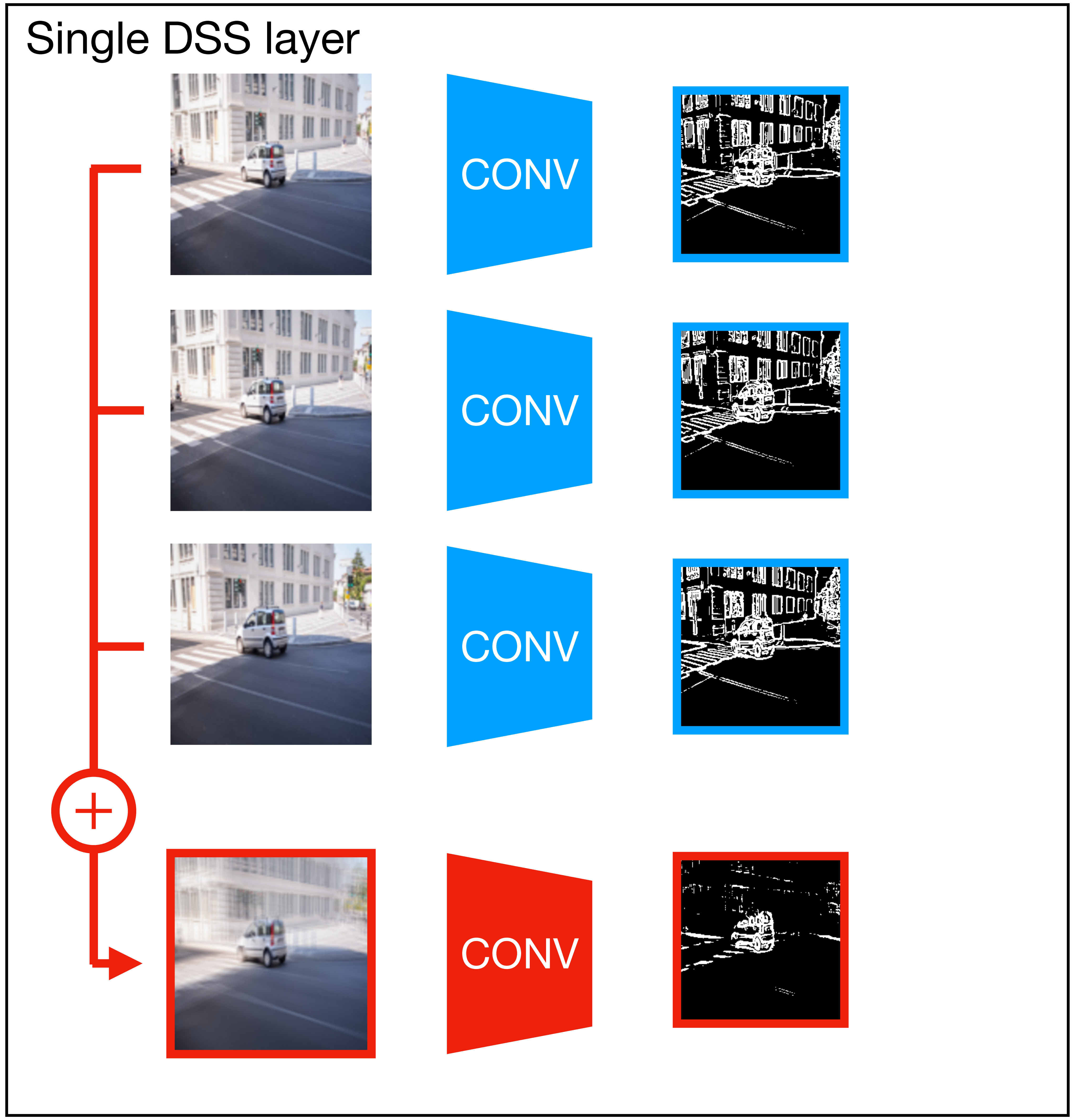


DSS for images

$x_1, \ldots, x_n$ are images

G is the group of $2D$ circular translations

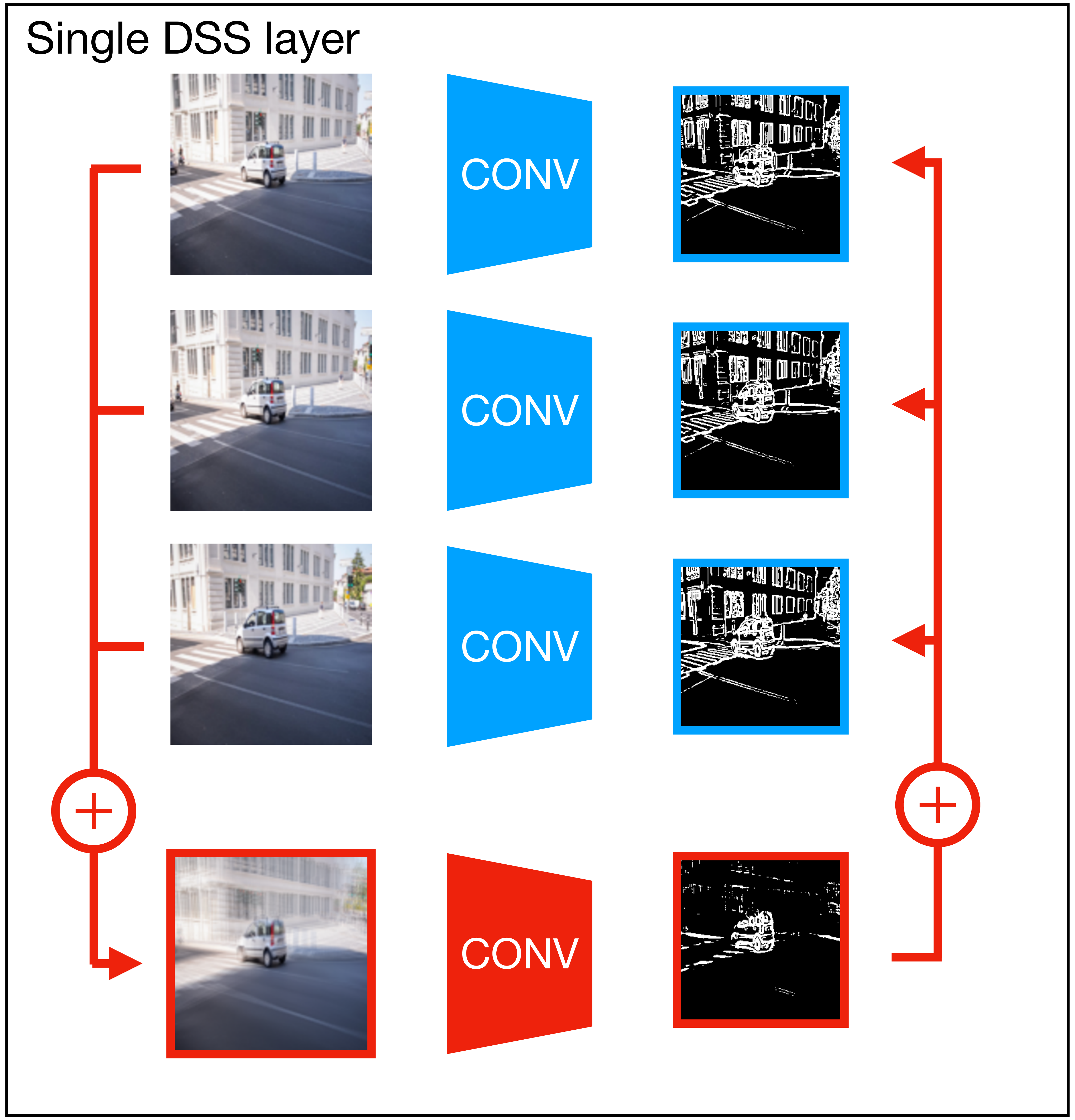
G-equivariant layers are convolutions



DSS for images

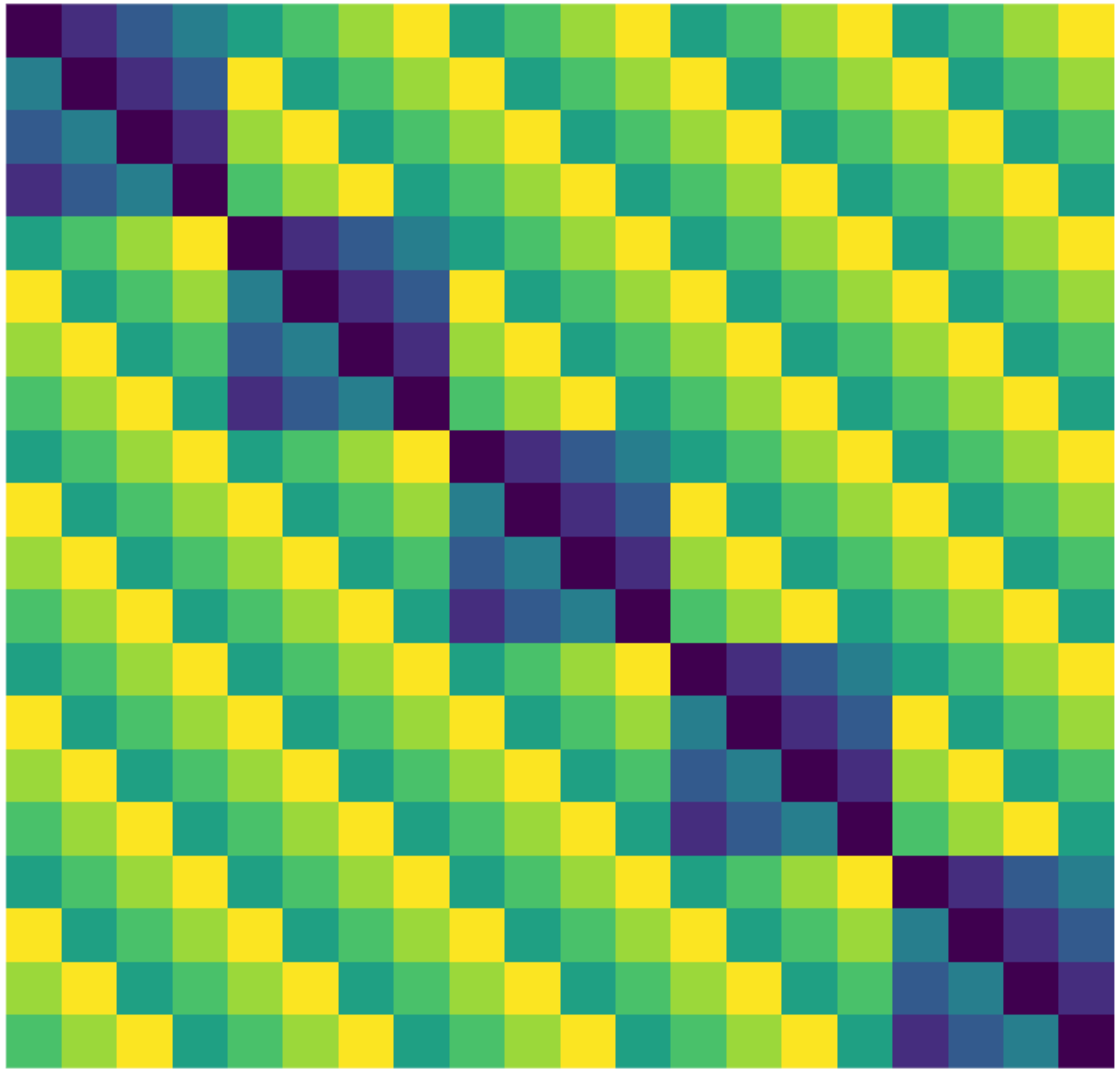
Siamese part

Information sharing part



Parameter sharing scheme

- $n = 5, d = 4, H = C_4$



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.

Conclusions

- Architectures for structured objects can benefit from taking into account **underlying symmetries**
- Boils down to a **parameter-sharing scheme** for permutation actions
- Pay attention to **expressivity**

The end

Relevant papers:

- Invariant and Equivariant Graph Networks. M. et al., *ICLR 2019*
- On the Universality of Invariant Networks. M. et al., *ICML 2019*
- Provably Powerful Graph Networks. M. et al., *NeurIPS 2019*
- Approximation Power of Invariant Graph Networks. M. et al., *NeurIPS 2019 GRLW*
- On Learning Sets of Symmetric Elements. M. et al., *ICML 2020*, Outstanding paper award

Collaborators: Yaron Lipman, Heli Ben-Hamu, Nadav Shamir, Hadar Serviansky, Nimrod Segol, Ethan Fetaya, Gal Chechik, Or Litany.

- Also see multiple very related papers papers by **Siamak Ravanbakhsh**