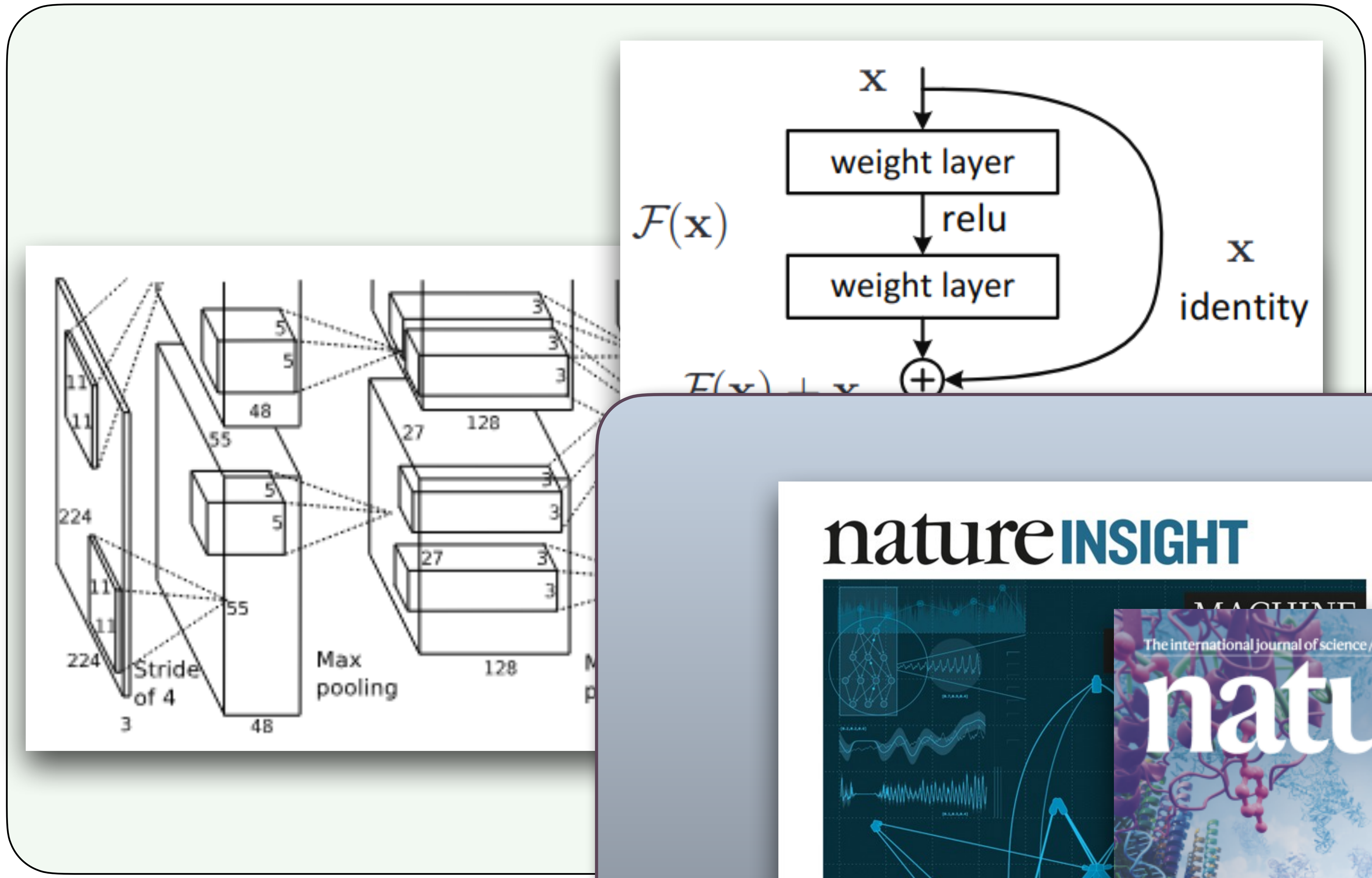
Scale Equivariant Graph Metanetworks

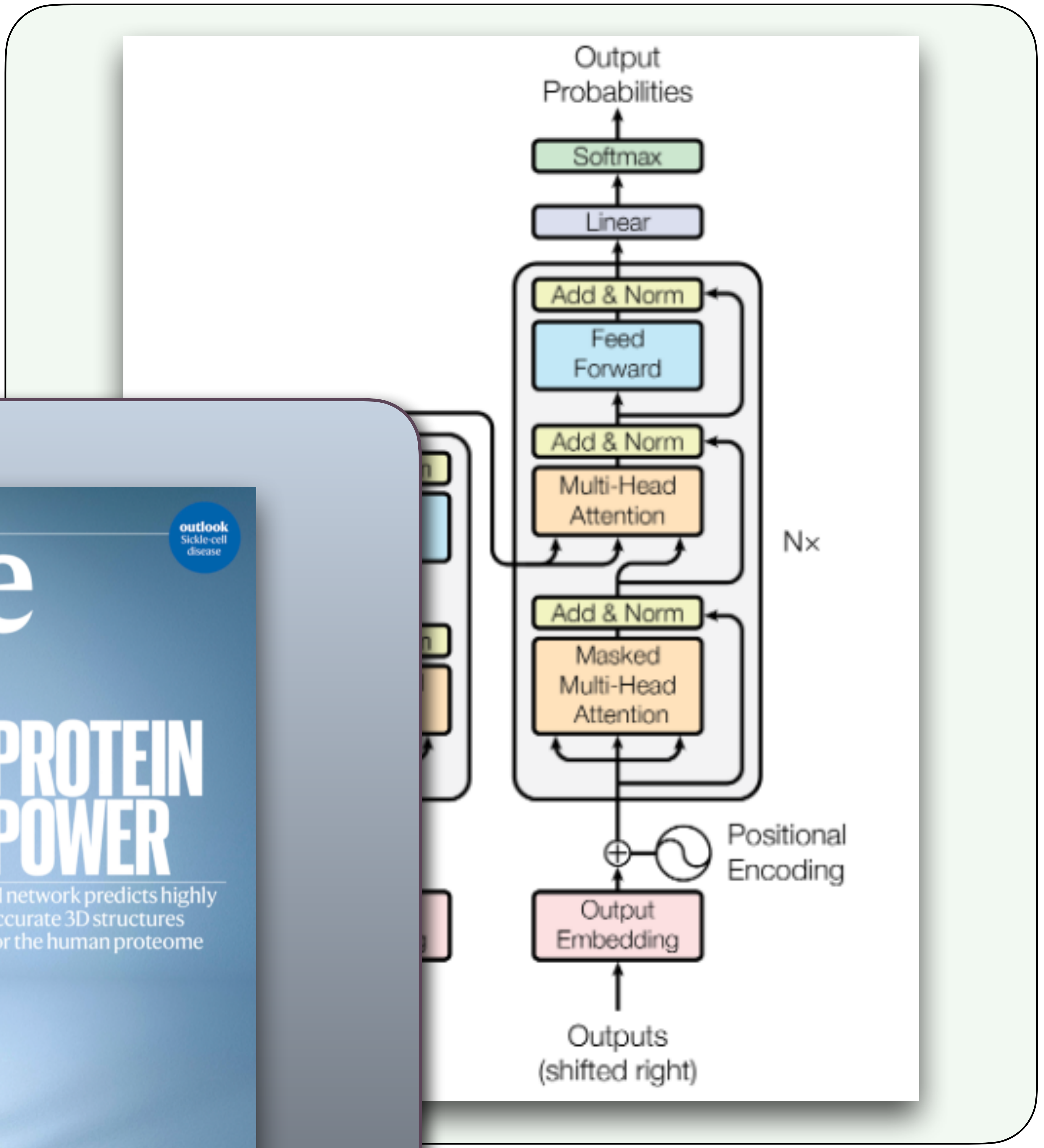
Ioannis Kalogeropoulos, Giorgos Bouritsas* and Yannis Panagakis*

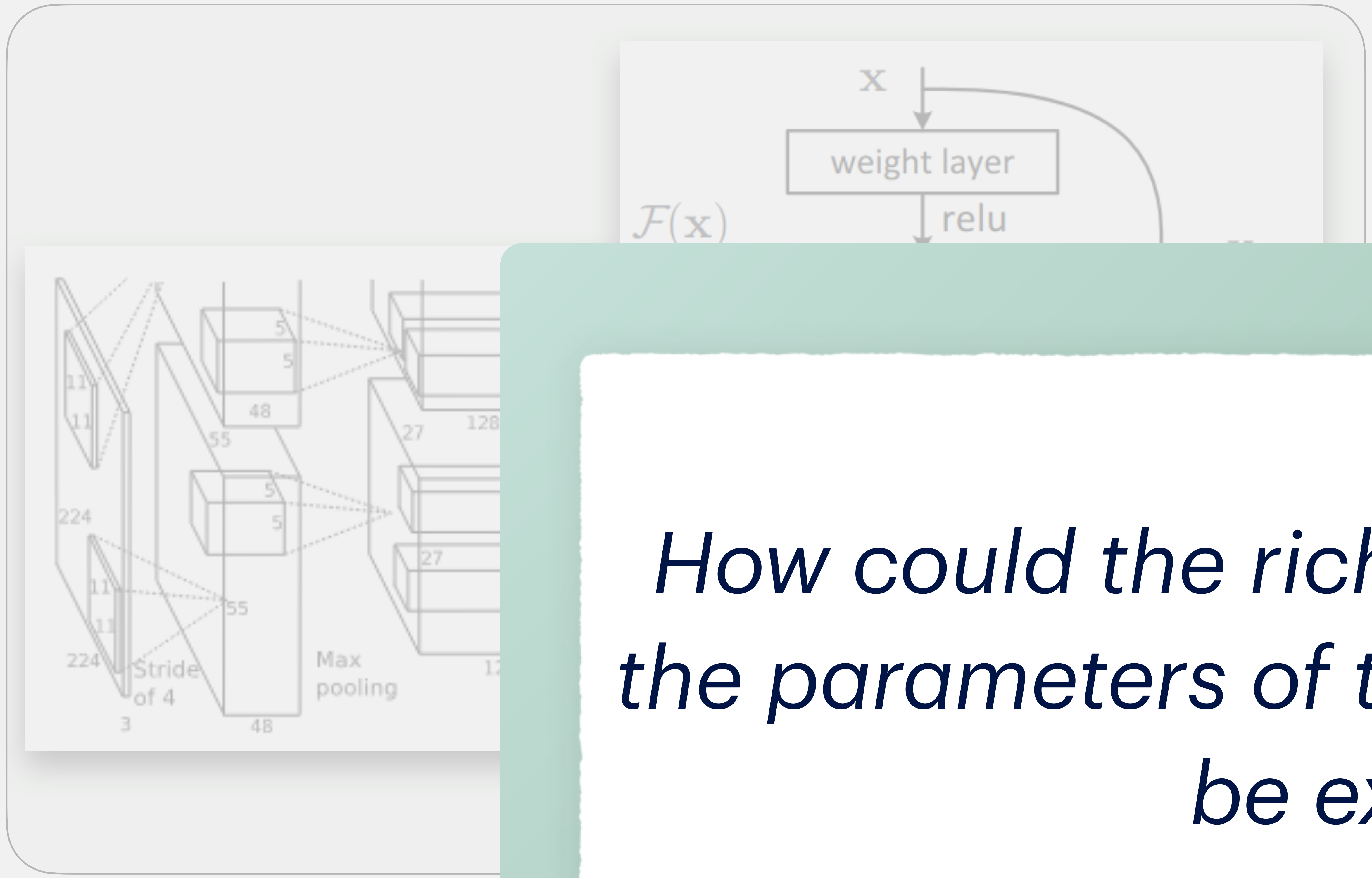
HELLENIC REPUBLIC
National and Kapodistrian
University of Athens
EST. 1837

ARCHIMEDES

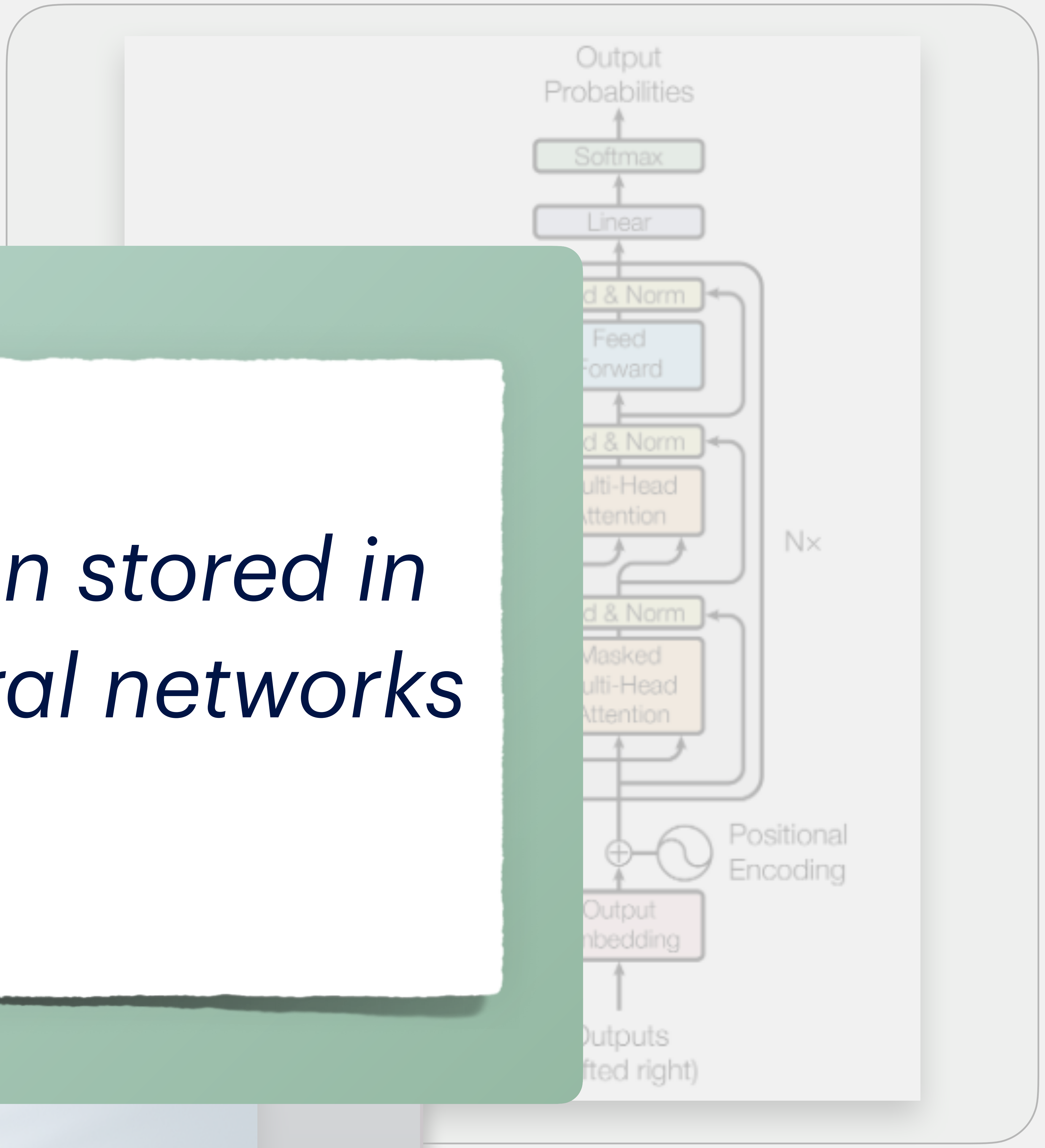

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
224
224
Stride of 4
3
11
11
55
55
5
5
48
48
Max pooling
27
27
128
128
3
3

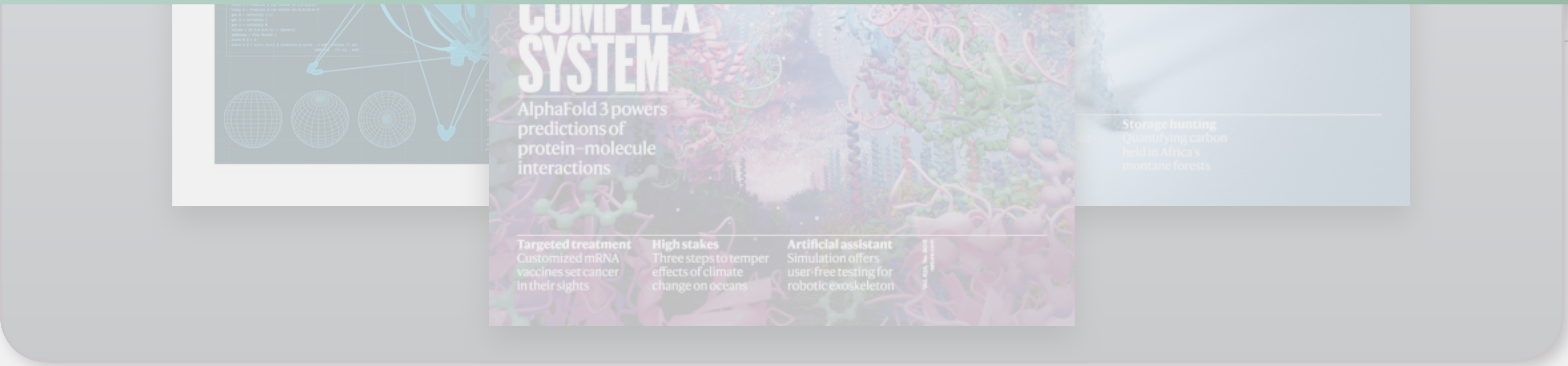

nature INSIGHT
The international journal of science / 26 August 2021
nature
outlook
PROTEIN POWER
AI network predicts highly accurate 3D structures for the human proteome
Storage hunting
Quantifying carbon held in Africa's montane forests
The international journal of science / 13 June 2024
nature
COMPLEX SYSTEM
AlphaFold 3 powers predictions of protein–molecule interactions
Targeted treatment
Customized mRNA vaccines set cancer in their sights
High stakes
Three steps to temper effects of climate change on oceans
Artificial assistant
Simulation offers user-free testing for robotic exoskeleton


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Output Embedding
Outputs (shifted right)



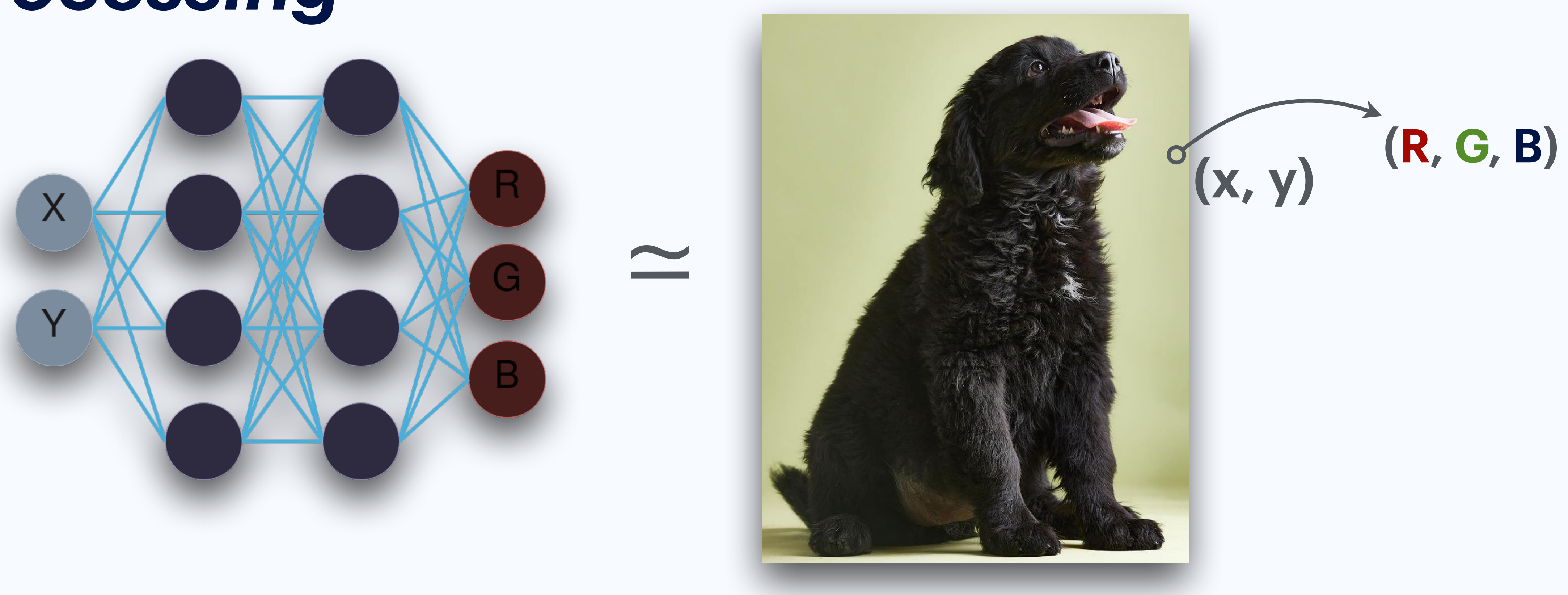
How could the rich information stored in the parameters of trained neural networks be exploited?





Why?

INR / NeRF Processing

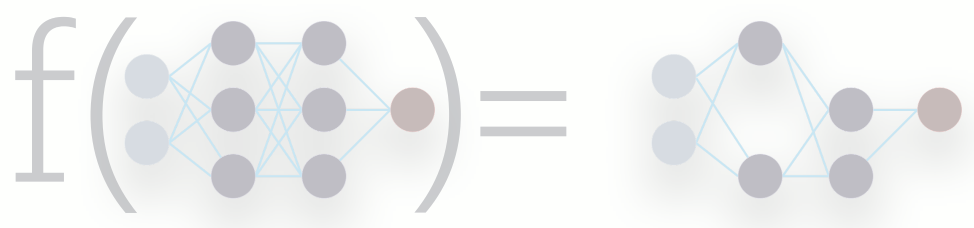


- Classification
- Editing
- 3D generation

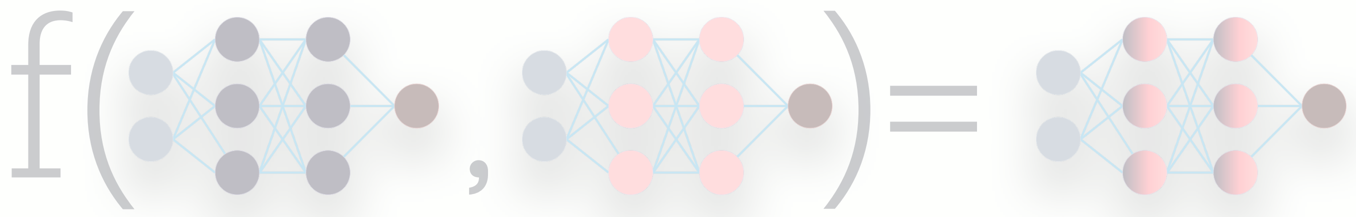
*Potential *unified framework* to handle different signals.

NN Editing

- Pruning



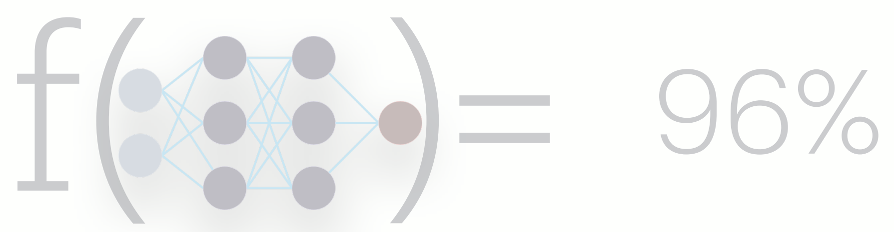
- Merging



- Domain adaptation

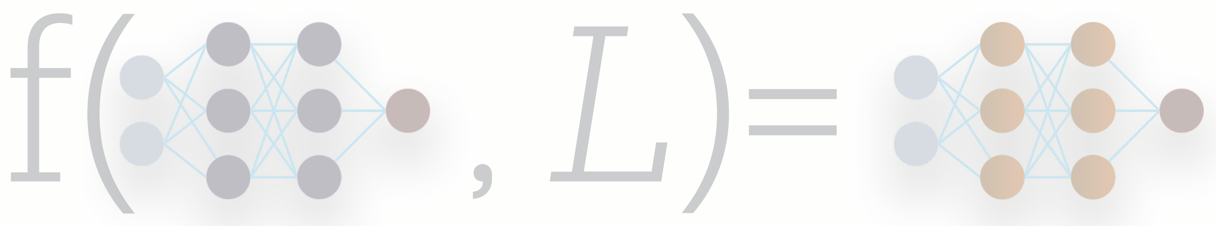
Analysis/Interpretation

- Generalization prediction

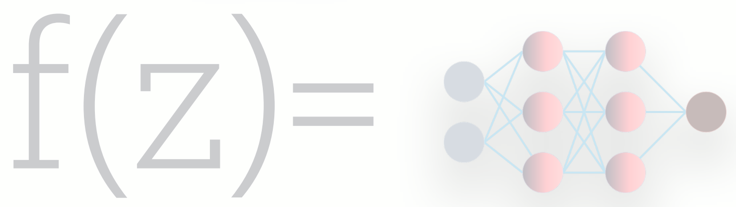


NN Synthesis

- Optimization

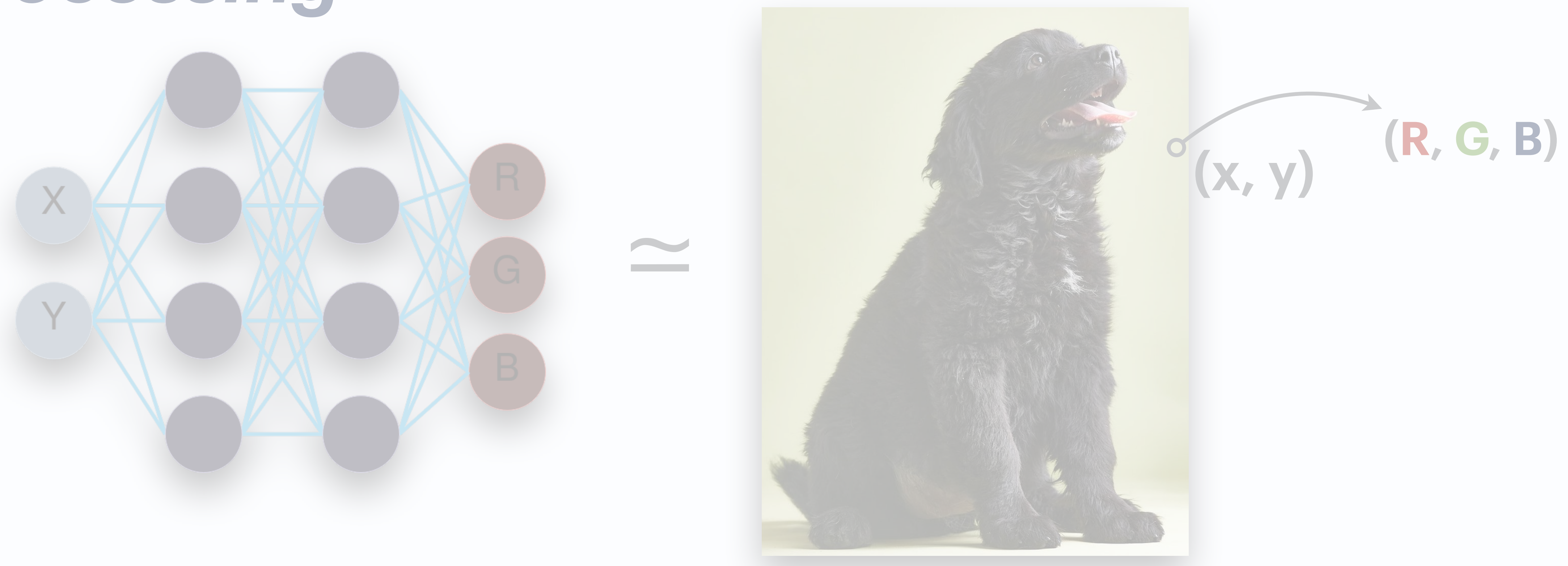


- Parameter generation



Why?

INR / NeRF Processing

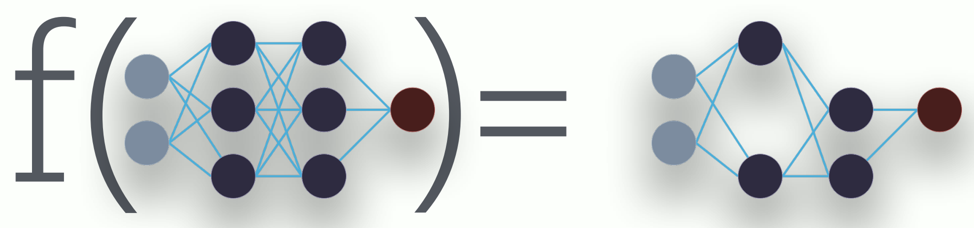


- Classification
- Editing
- 3D generation

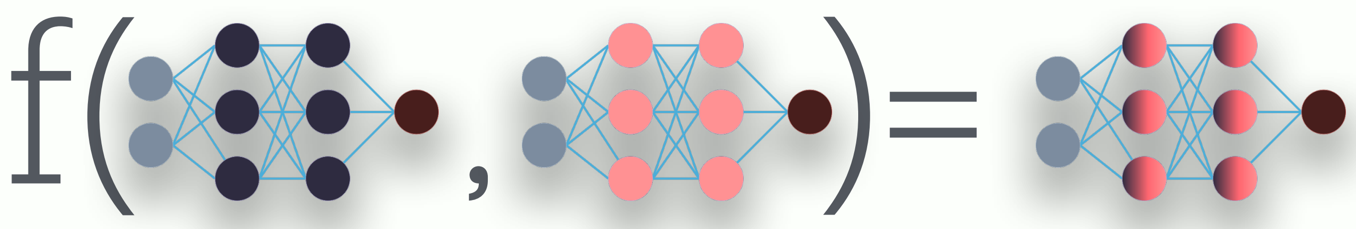
*Potential *unified framework* to handle different signals.

NN Editing

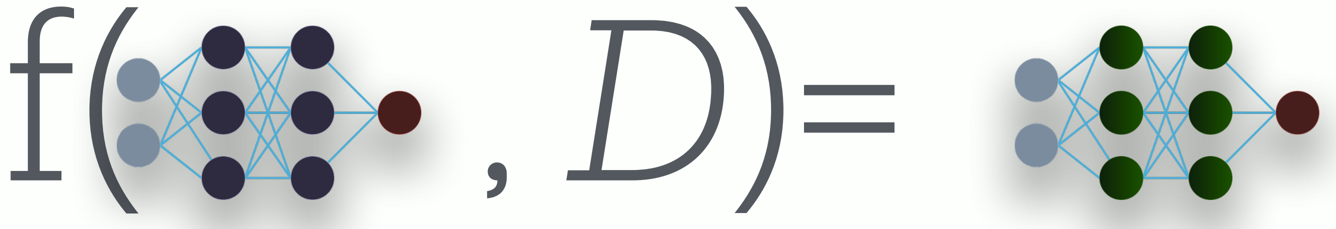
- Pruning



- Merging

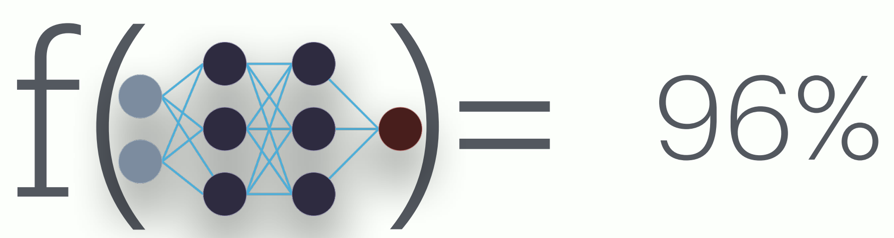


- Domain adaptation



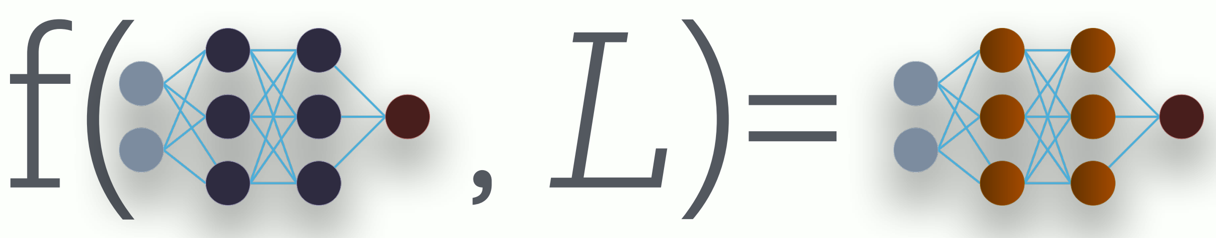
Analysis/Interpretation

- Generalization prediction

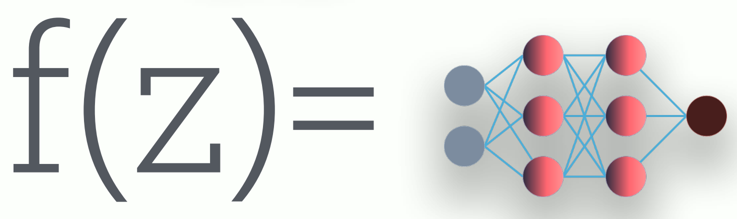


NN Synthesis

- Optimization



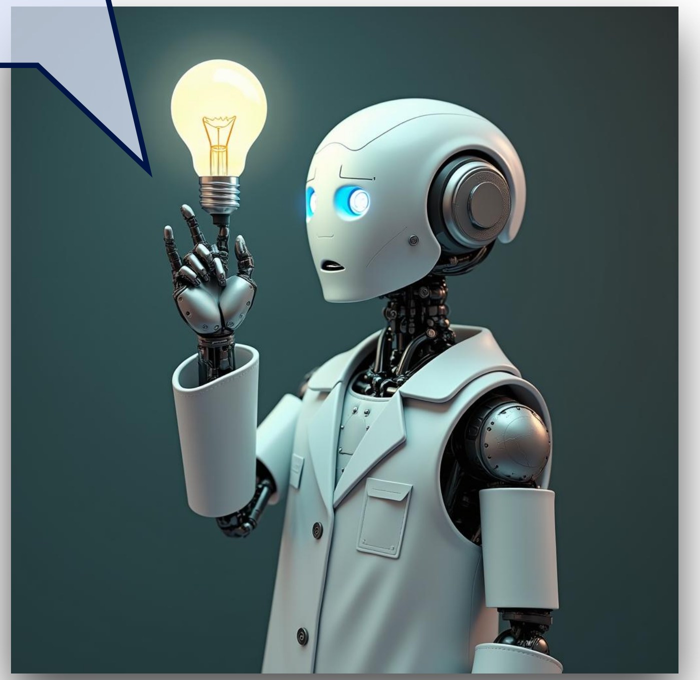
- Parameter generation

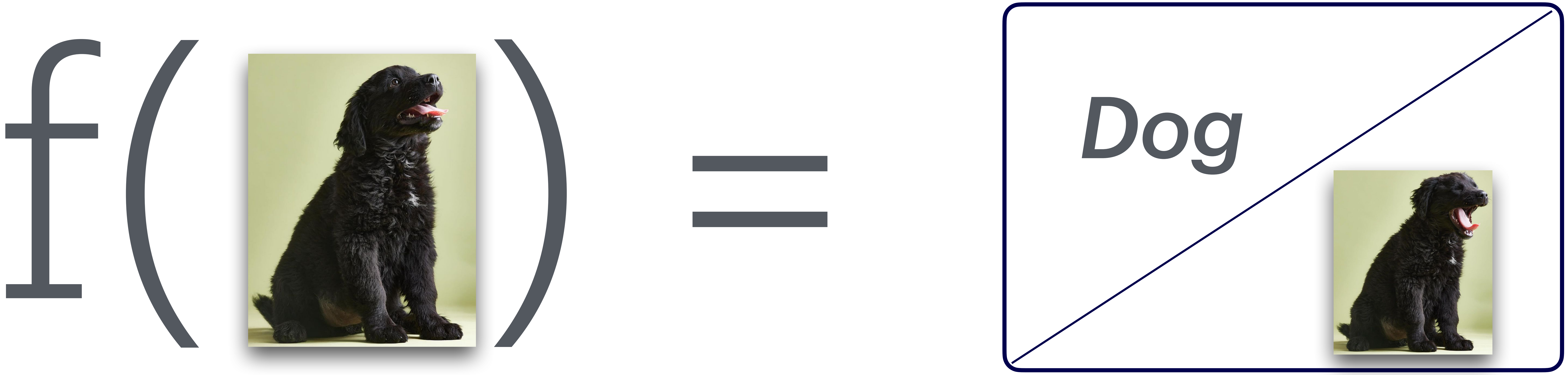


How can we process and extract insights solely from the parameters of NNs?

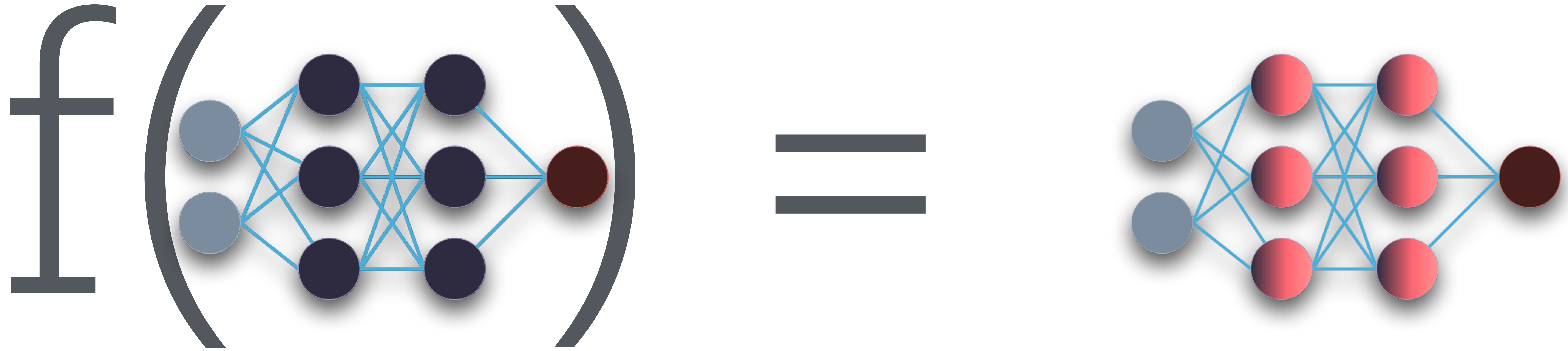
How can we process and extract insights solely from the parameters of NNs?

Devise architectures that learn to process other neural architectures!



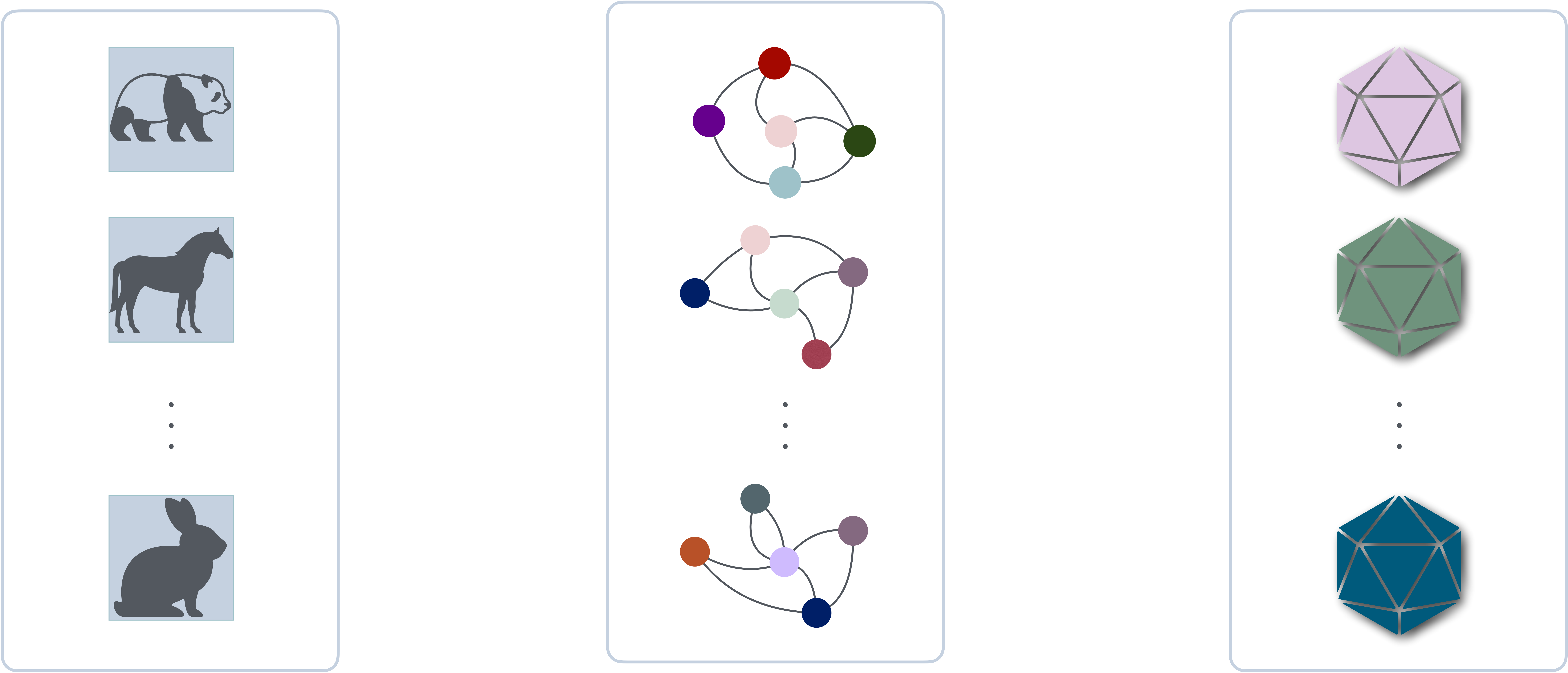


*aka Learning higher-order functions

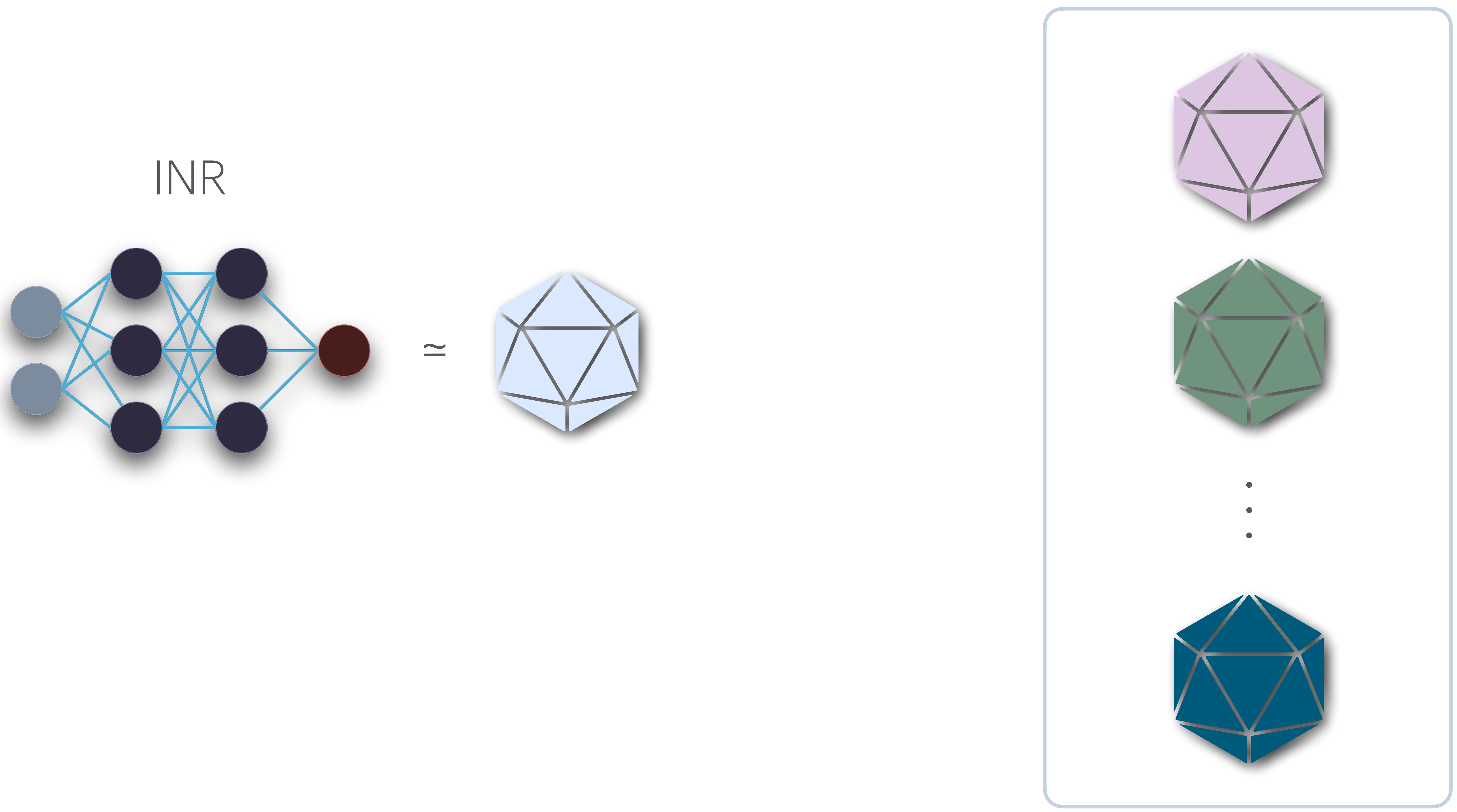


*aka Learning higher-order functions

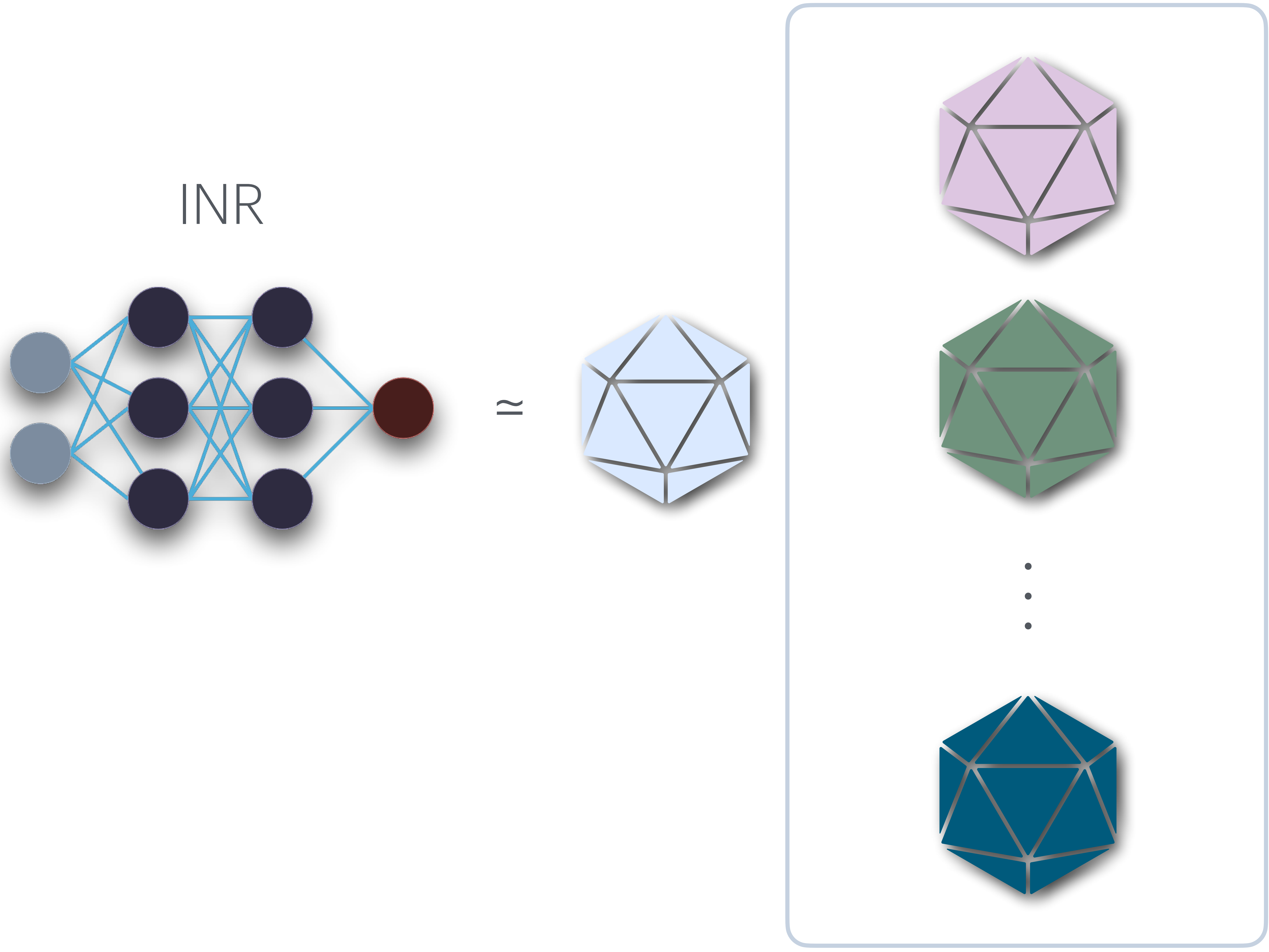
So far: Datasets of signals

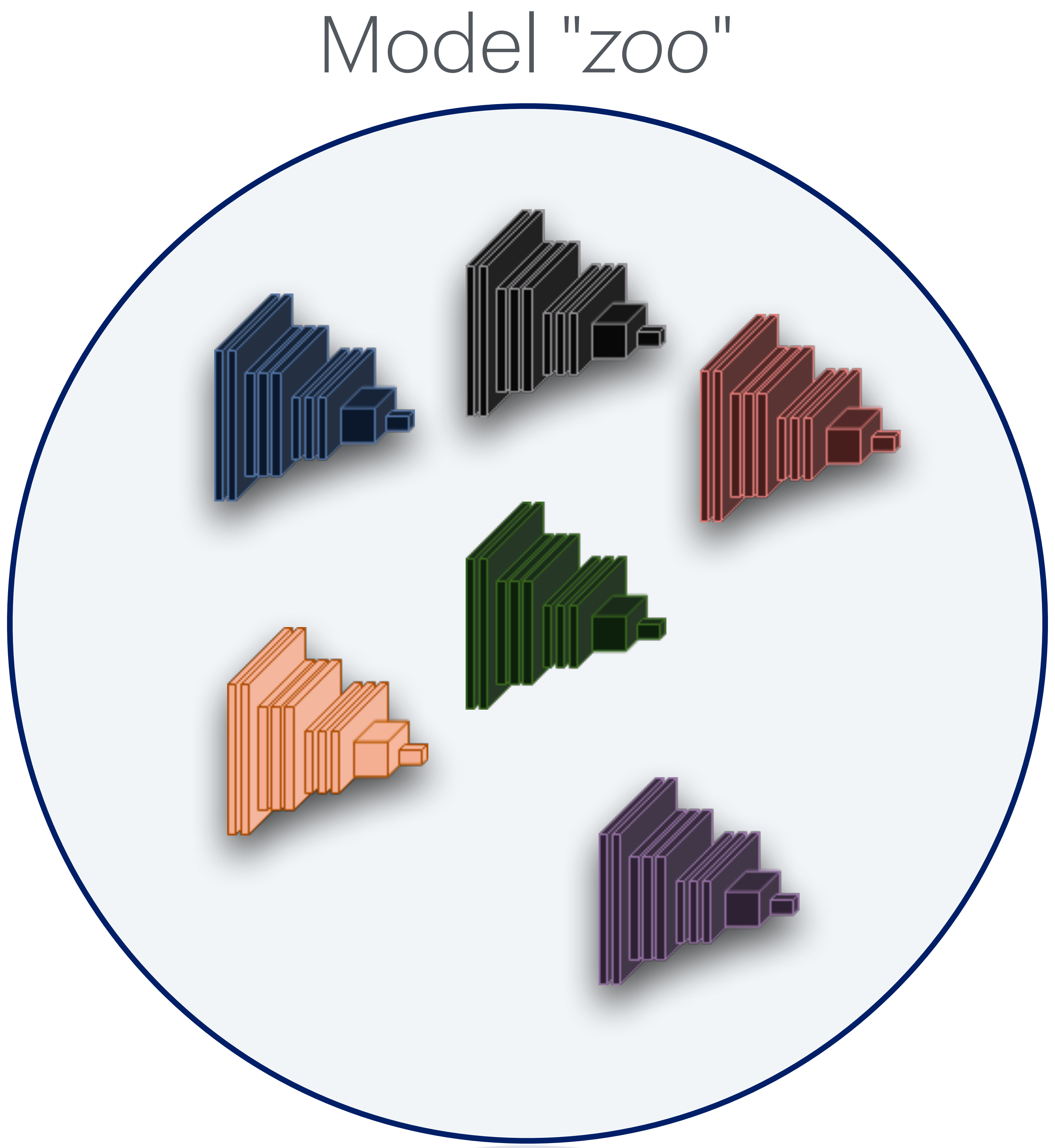


So far: Datasets of signals



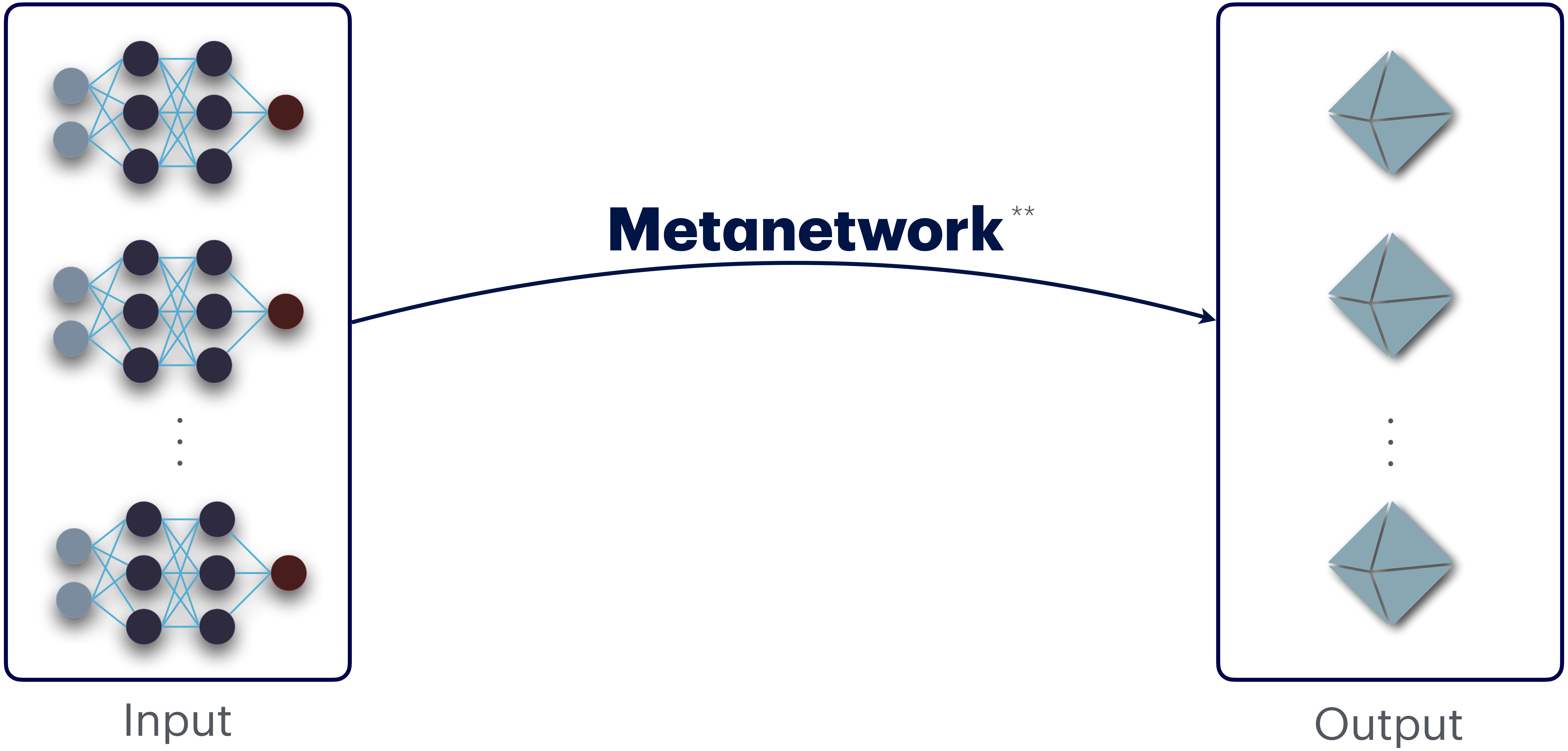
So far: Datasets of signals





New paradigm: Datasets of NNs *

~~Datasets of signals~~



* Dupont, Emilien, et al., ICML 2022
**Lim, Derek, et al., ICLR 2024

Previous approaches

1. ***Ignoring* structure***:

- **Flatten** weights
- Jointly fitting INR embeddings ***with meta-learning techniques***
- etc.

* Unterthiner, T. et al. 2020, De Luigi, L.,et al., ICLR (2023), Dupont, Emilien, et al., ICML 2022

Previous approaches

1. ***Ignoring* structure*:**

- **Flatten** weights
- Jointly fitting INR embeddings ***with meta-learning techniques***
- etc.

2. ***Equivariant*** - Structure aware**:

Non local

- Construct linear equivariant layers to permutation symmetries.
- Intricate weight-sharing patterns.
- Cannot process varying architectures.

* Unterthiner, T. et al. 2020, De Luigi, L.,et al., ICLR (2023), Dupont, Emilien, et al., ICML 2022

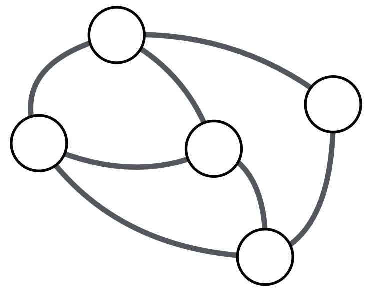
** Navon, A., et al. ICML 2023, Zhou, A., et al. NeurIPS 2024, Lim, D., et al. ICLR 2024, Kofinas, M., et al. ICLR 2024.

Previous approaches

1. ***Ignoring* structure***:

- **Flatten** weights • Jointly fitting INR embeddings ***with meta-learning techniques*** • etc.

2. *Equivariant* - Structure aware**:

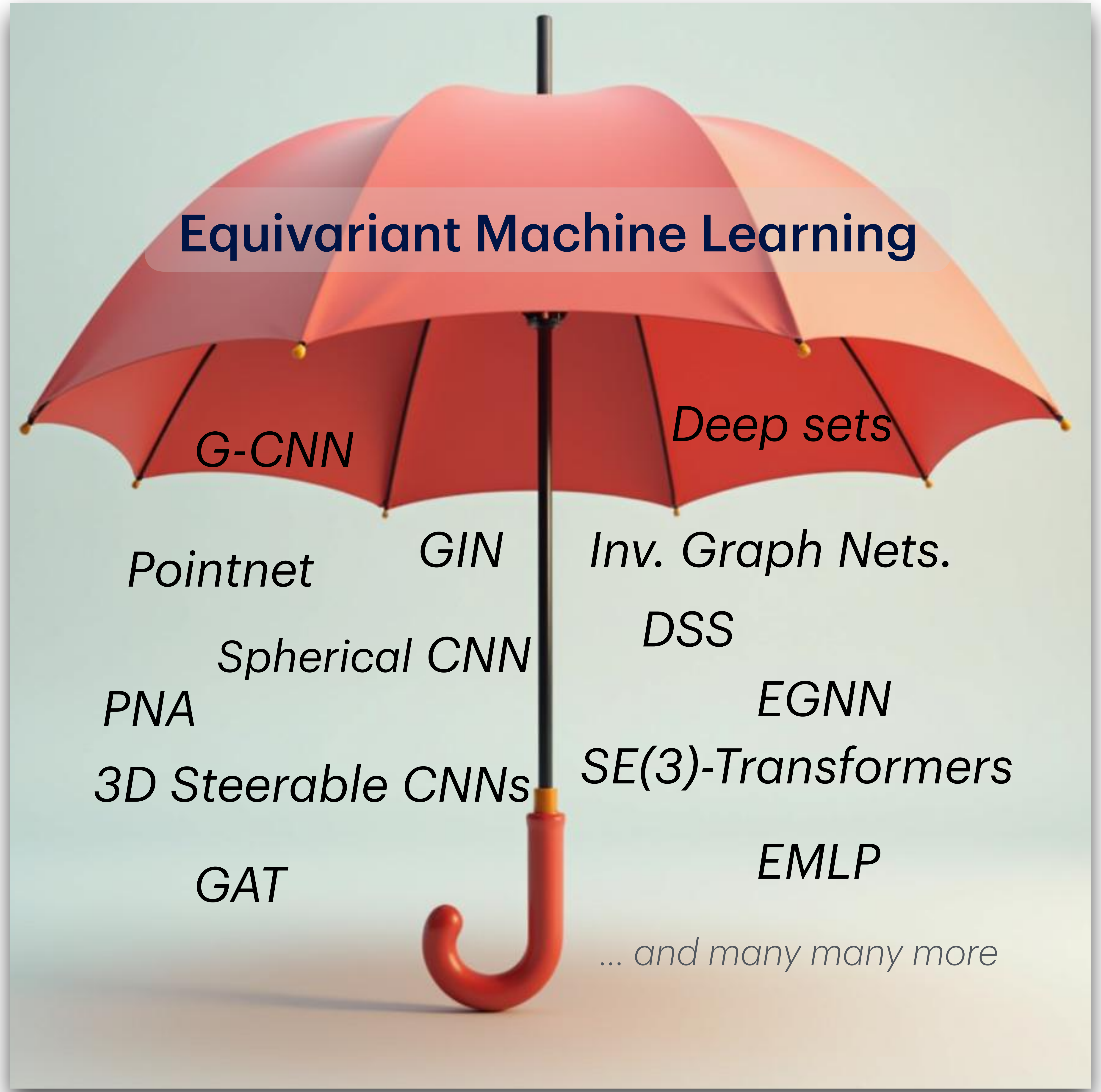
Non local	Graph-based
• Construct linear equivariant layers to permutation symmetries. • Intricate weight-sharing patterns. • Cannot process varying architectures.	• Treat NNs as graphs.  • Process them with GNNs. • Can process varying architectures.

* Unterthiner, T. et al. 2020, De Luigi, L.,et al., ICLR (2023), Dupont, Emilien, et al., ICML 2022

** Navon, A., et al. ICML 2023, Zhou, A., et al. NeurIPS 2024, Lim, D., et al. ICLR 2024, Kofinas, M., et al. ICLR 2024.

Q: What makes NNs *different* from other modalities?

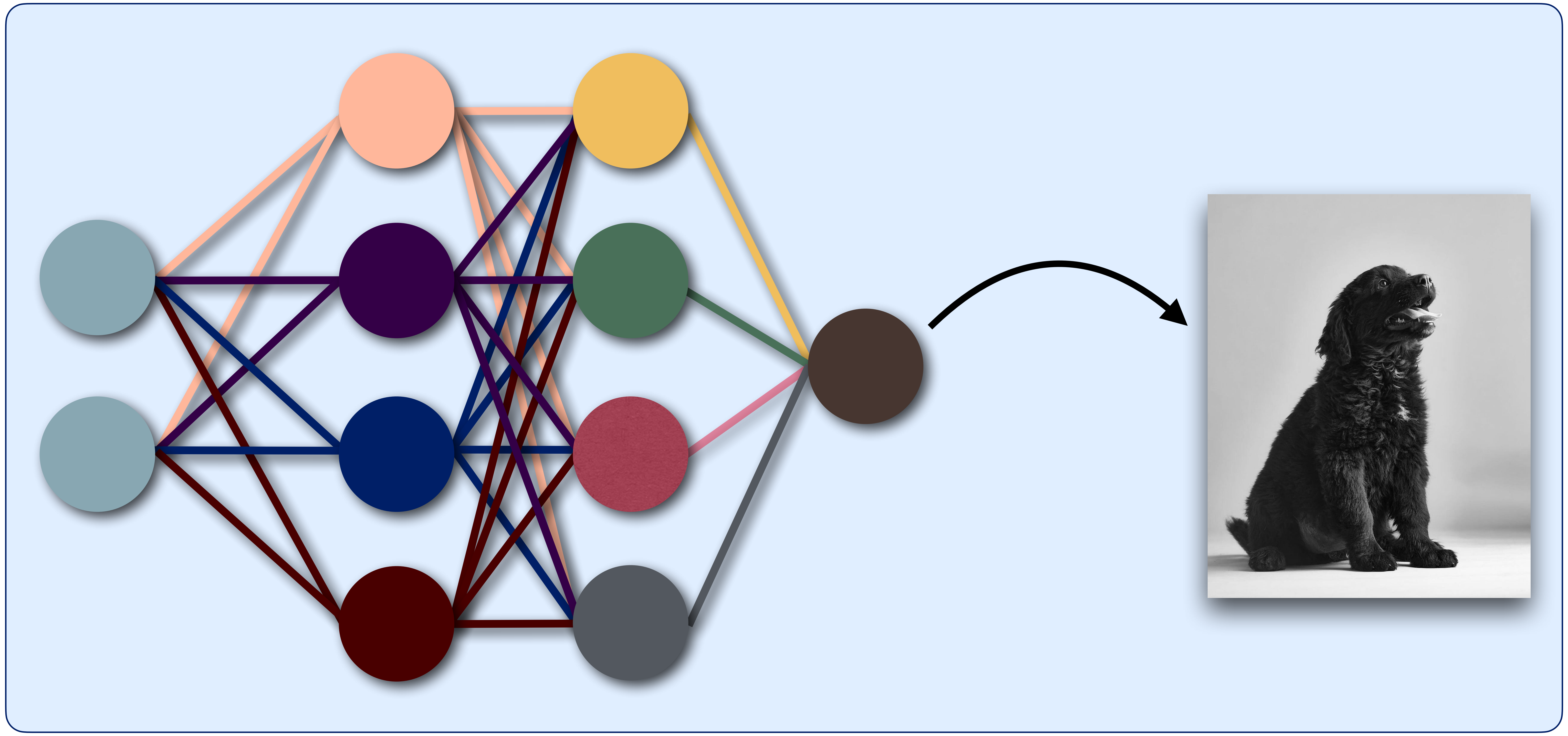
A: ***Symmetries.***



*Cohen, T., & Welling, M. ICML 2016, Zaheer, M., et al. NIPS 2017, Qi, Charles R., et al. CVPR 2017, Xu, K., et al. ICLR 2019, Maron, H., et al ICLR 2019, Cohen, T. S., et al. ICLR 2018, Maron, H., et al. ICML 2020, Finzi, M., et al. ICML 2021, Veličković, P., et al. ICLR 2018, Fuchs, F., et al., NeurIPS 2020, Satorras, V. G. et al ICML 2021, Weiler M., et al. NeurIPS 2018

NN symmetries - *Permutation*

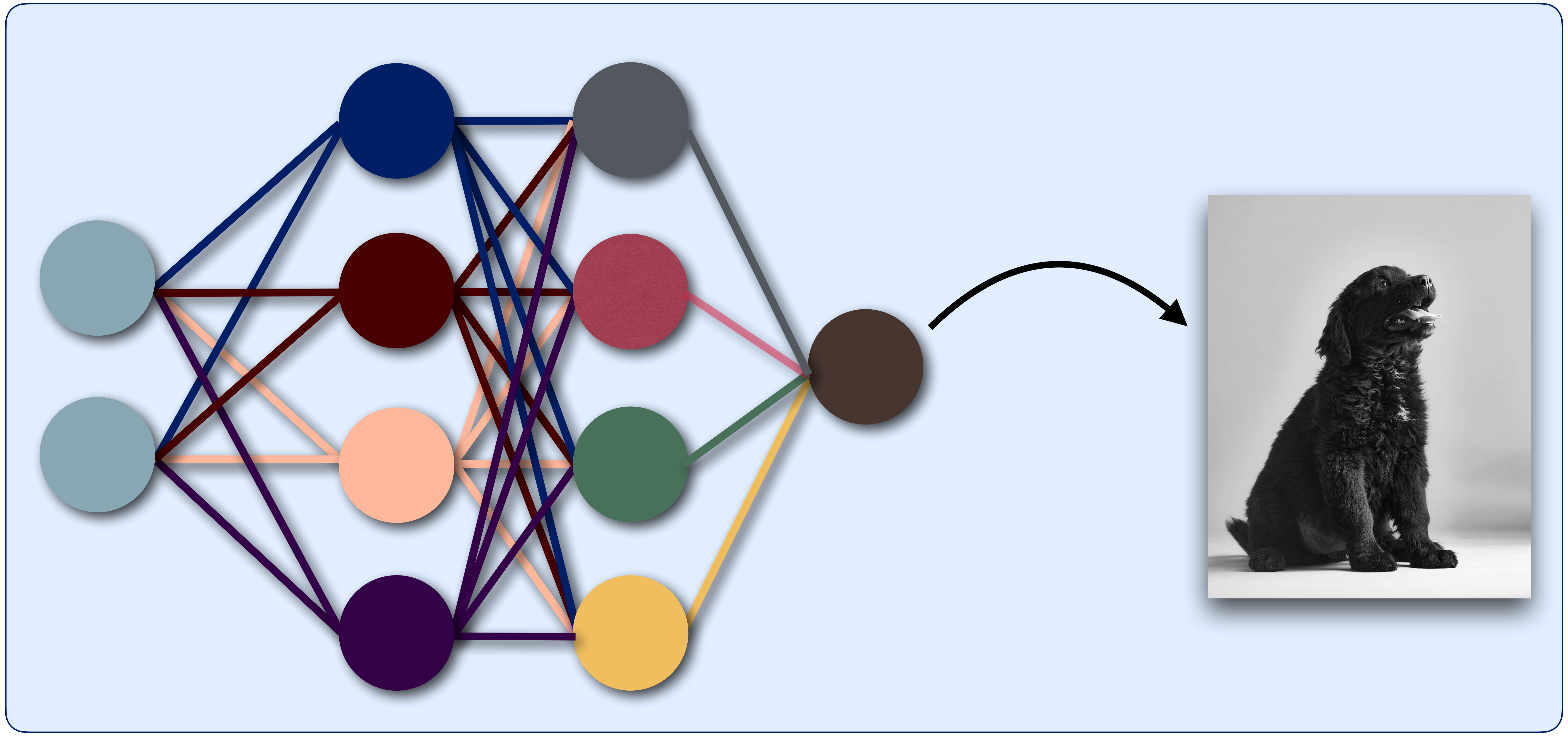
Hidden neurons do not possess any inherent ordering.



$$(\mathbf{P}_\ell \mathbf{W}_\ell \mathbf{P}_{\ell-1}^{-1}, \mathbf{P}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

NN symmetries - *Permutation*

Hidden neurons do not possess any inherent ordering.



$$(\mathbf{P}_\ell \mathbf{W}_\ell \mathbf{P}_{\ell-1}^{-1}, \mathbf{P}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

Previous works on neural network processing **account only for the *permutation* symmetries**.

Are these the only symmetries within neural networks?

NN symmetries - *Scaling*

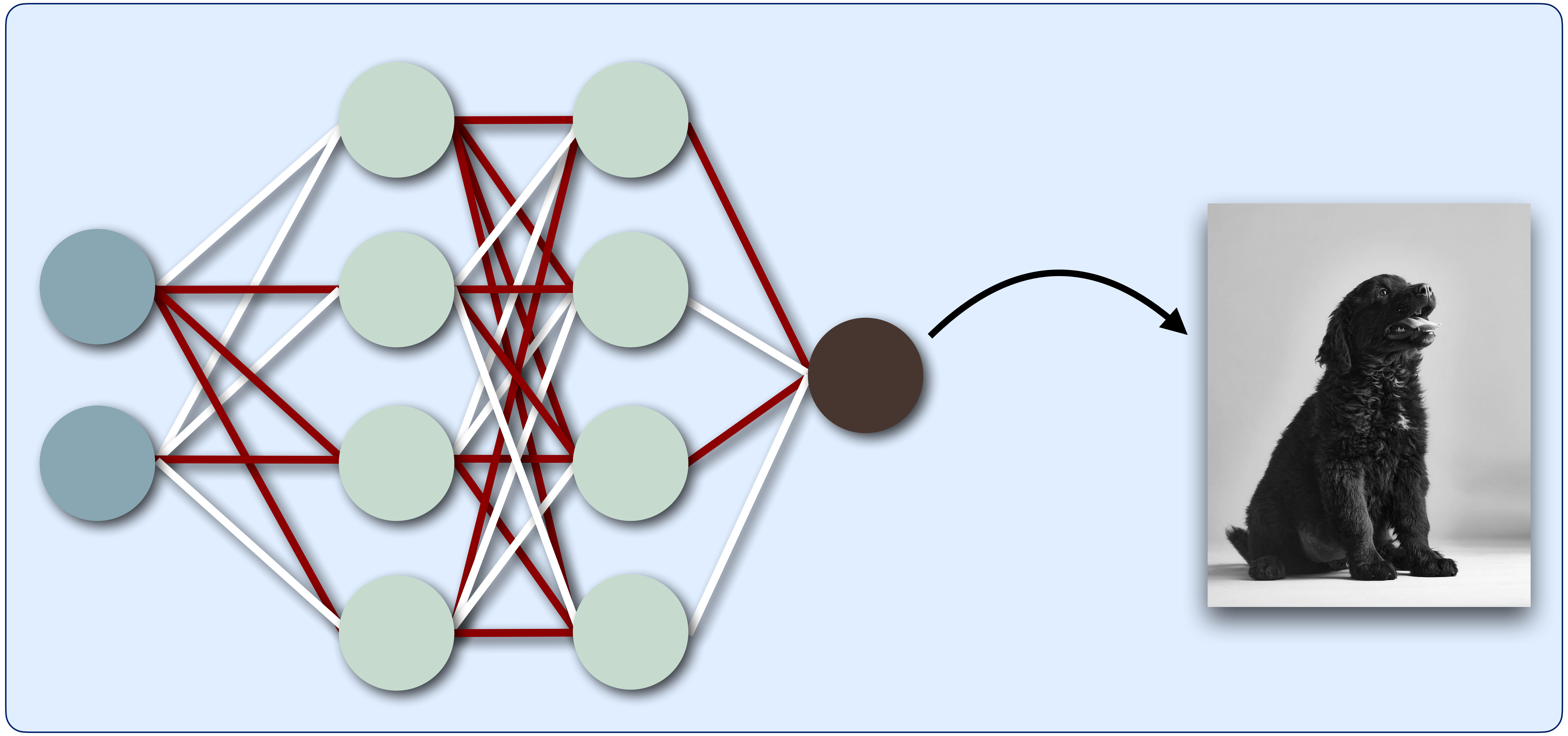
Activation functions have inherent symmetries bestowed to the NN.

Sign symmetry

(sine/tanh)

○ Positive ● Negative

sign flipping



$$(\mathbf{Q}_\ell \mathbf{W}_\ell \mathbf{Q}_{\ell-1}^{-1}, \mathbf{Q}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

NN symmetries - *Scaling*

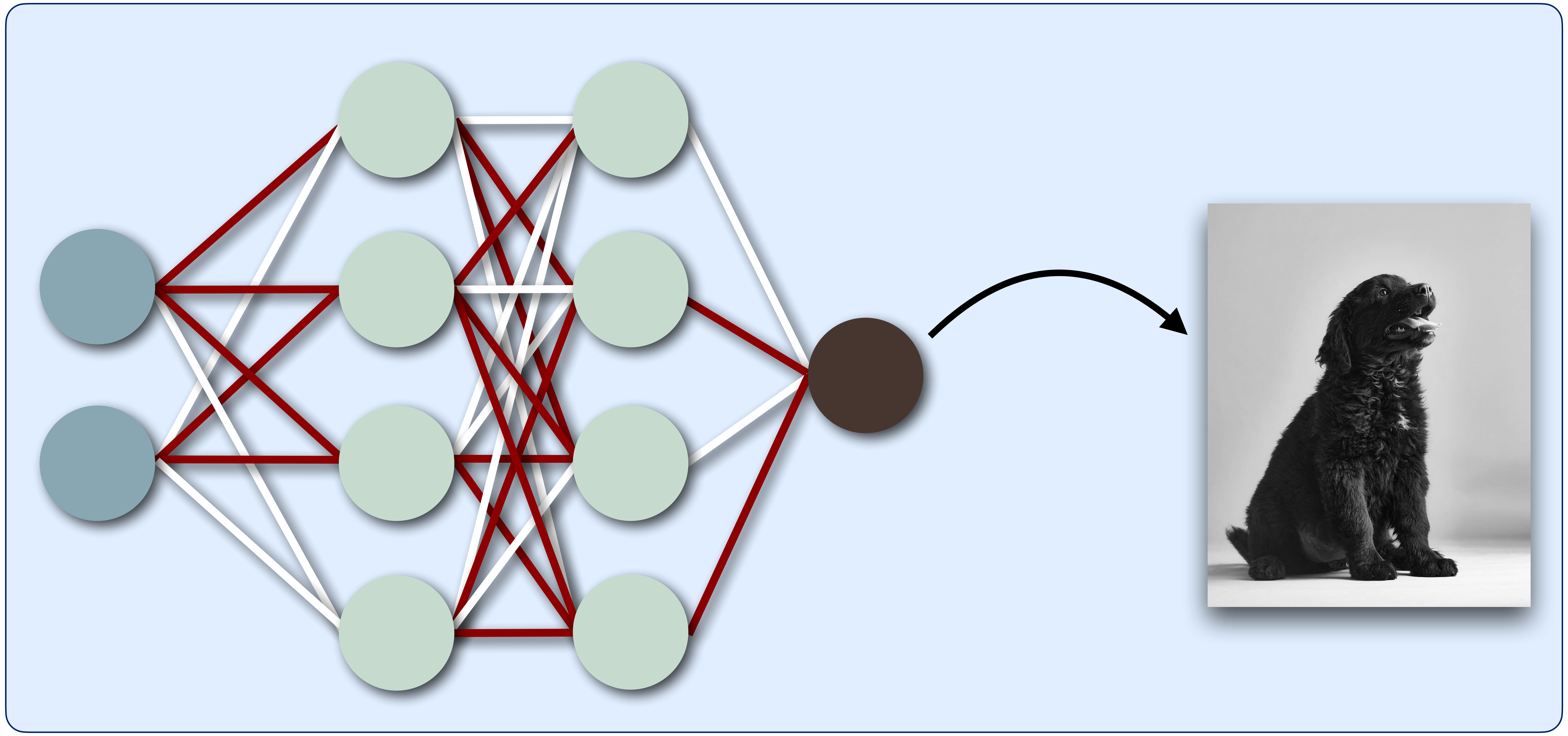
Activation functions have inherent symmetries bestowed to the NN.

Sign symmetry

(sine/tanh)

○ Positive ● Negative

sign flipping



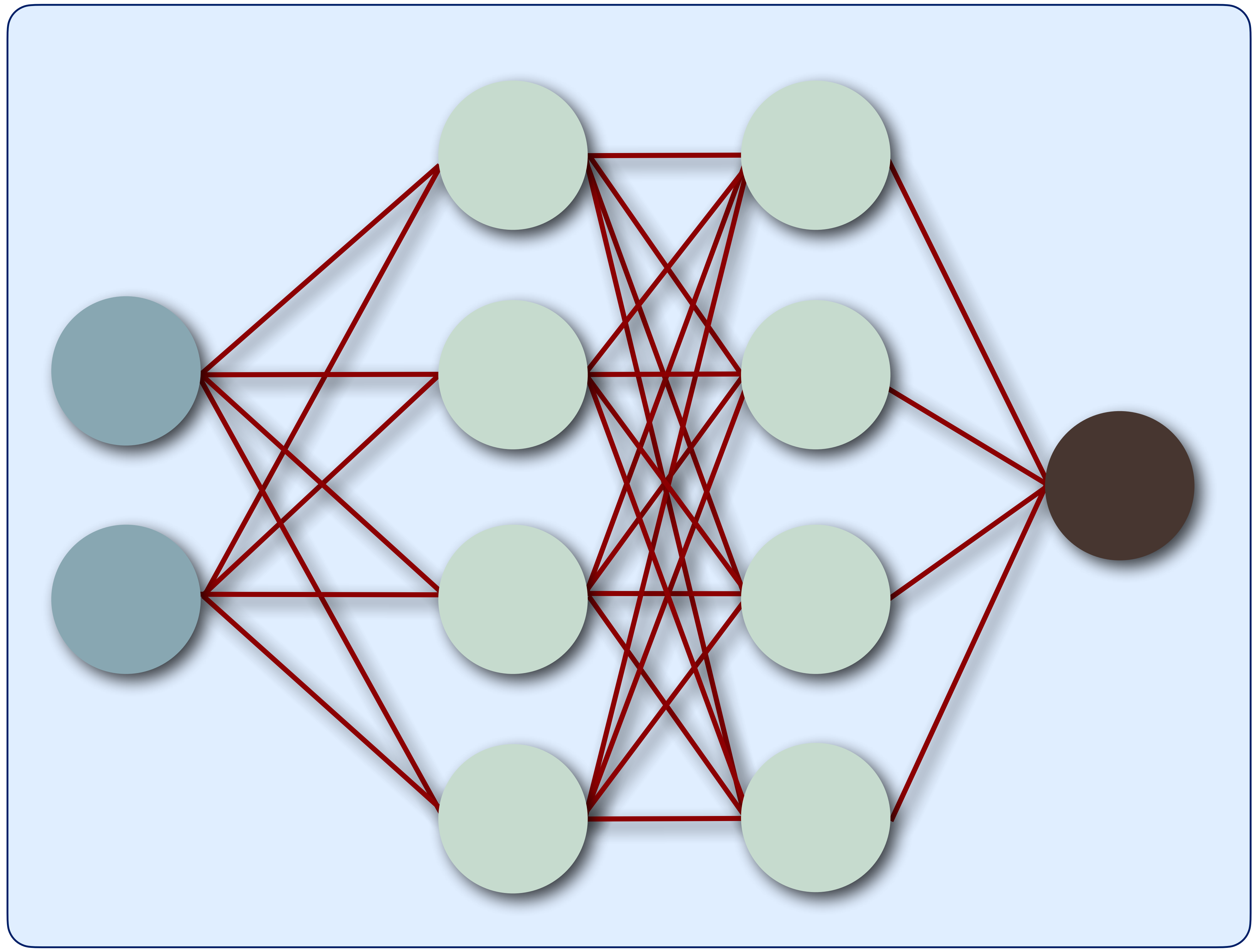
$$(\mathbf{Q}_\ell \mathbf{W}_\ell \mathbf{Q}_{\ell-1}^{-1}, \mathbf{Q}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

NN symmetries - *Scaling*

Activation functions have inherent symmetries bestowed to the NN.

Positive scale symmetry
(ReLU)

width: norm of scaling



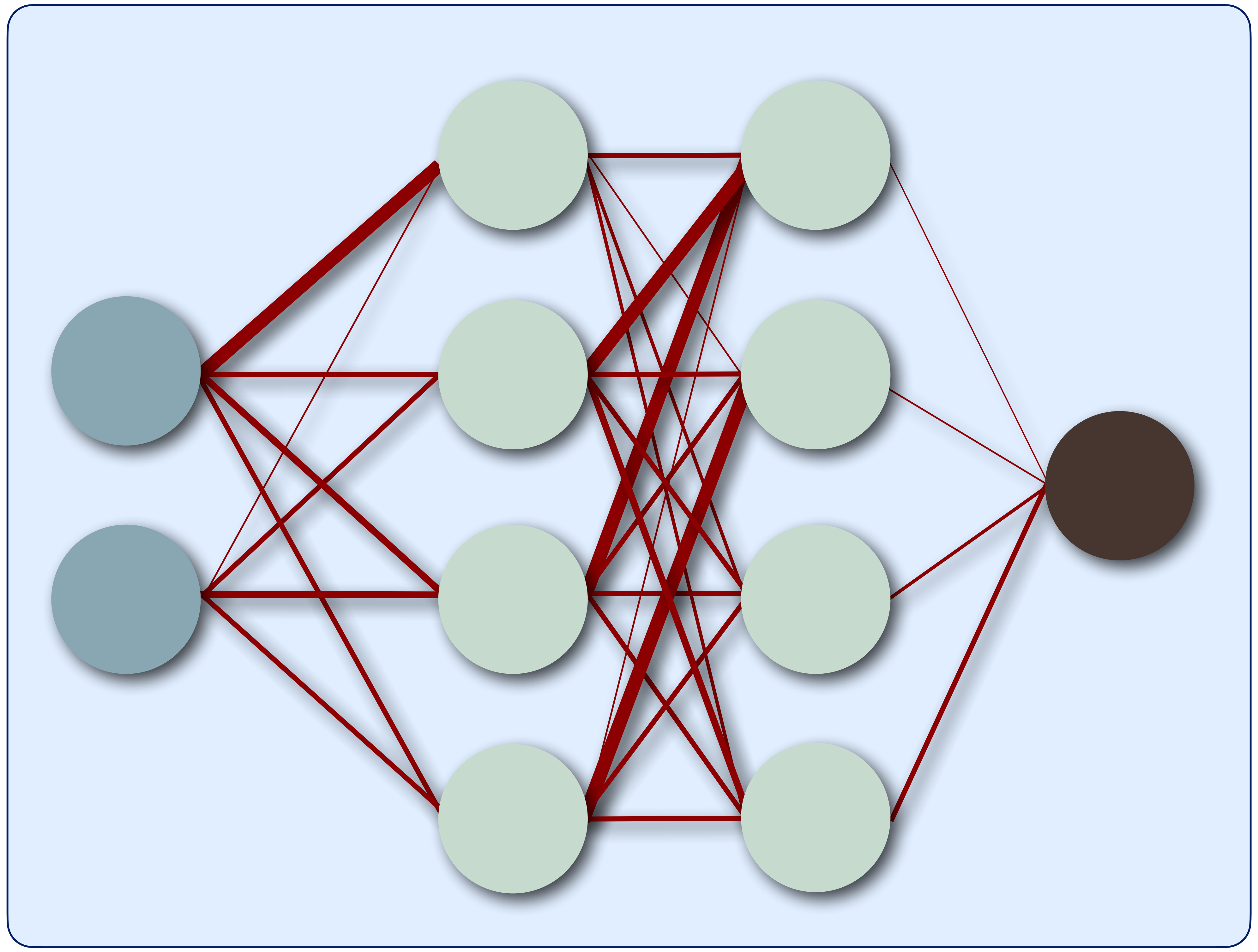
$$(\mathbf{Q}_\ell \mathbf{W}_\ell \mathbf{Q}_{\ell-1}^{-1}, \mathbf{Q}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

NN symmetries - *Scaling*

Activation functions have inherent symmetries bestowed to the NN.

Positive scale symmetry
(ReLU)

width: norm of scaling



$$(\mathbf{Q}_\ell \mathbf{W}_\ell \mathbf{Q}_{\ell-1}^{-1}, \mathbf{Q}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

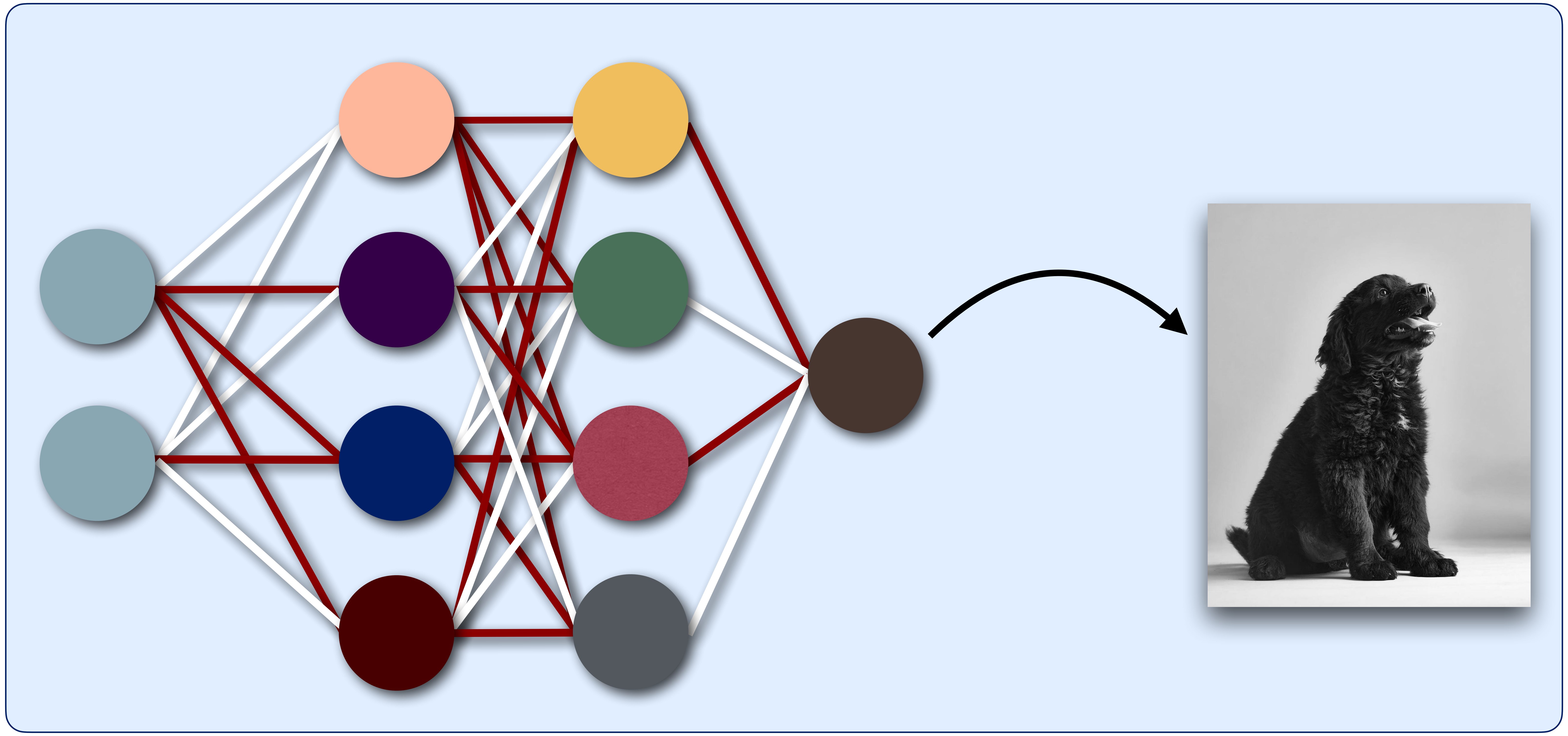
NN symmetries

Putting them all together

Sign symmetry
(sine/tanh)

○ Positive ● Negative

sign flipping



$$(\mathbf{P}_\ell \mathbf{Q}_\ell \mathbf{W}_\ell \mathbf{Q}_{\ell-1}^{-1} \mathbf{P}_{\ell-1}^{-1}, \mathbf{P}_\ell \mathbf{Q}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

NN symmetries

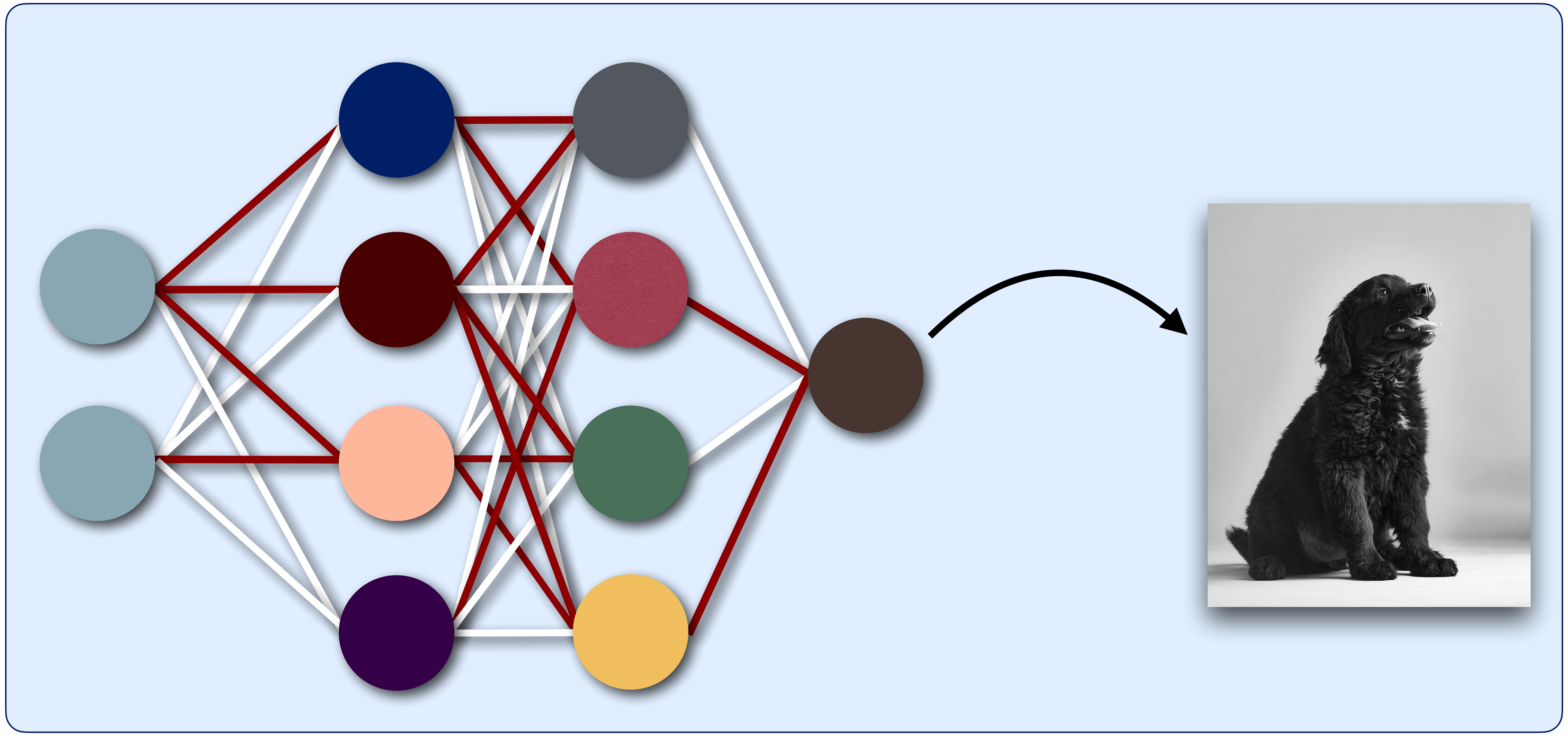
Putting them all together

Sign symmetry

(sine/tanh)

○ Positive ● Negative

sign flipping

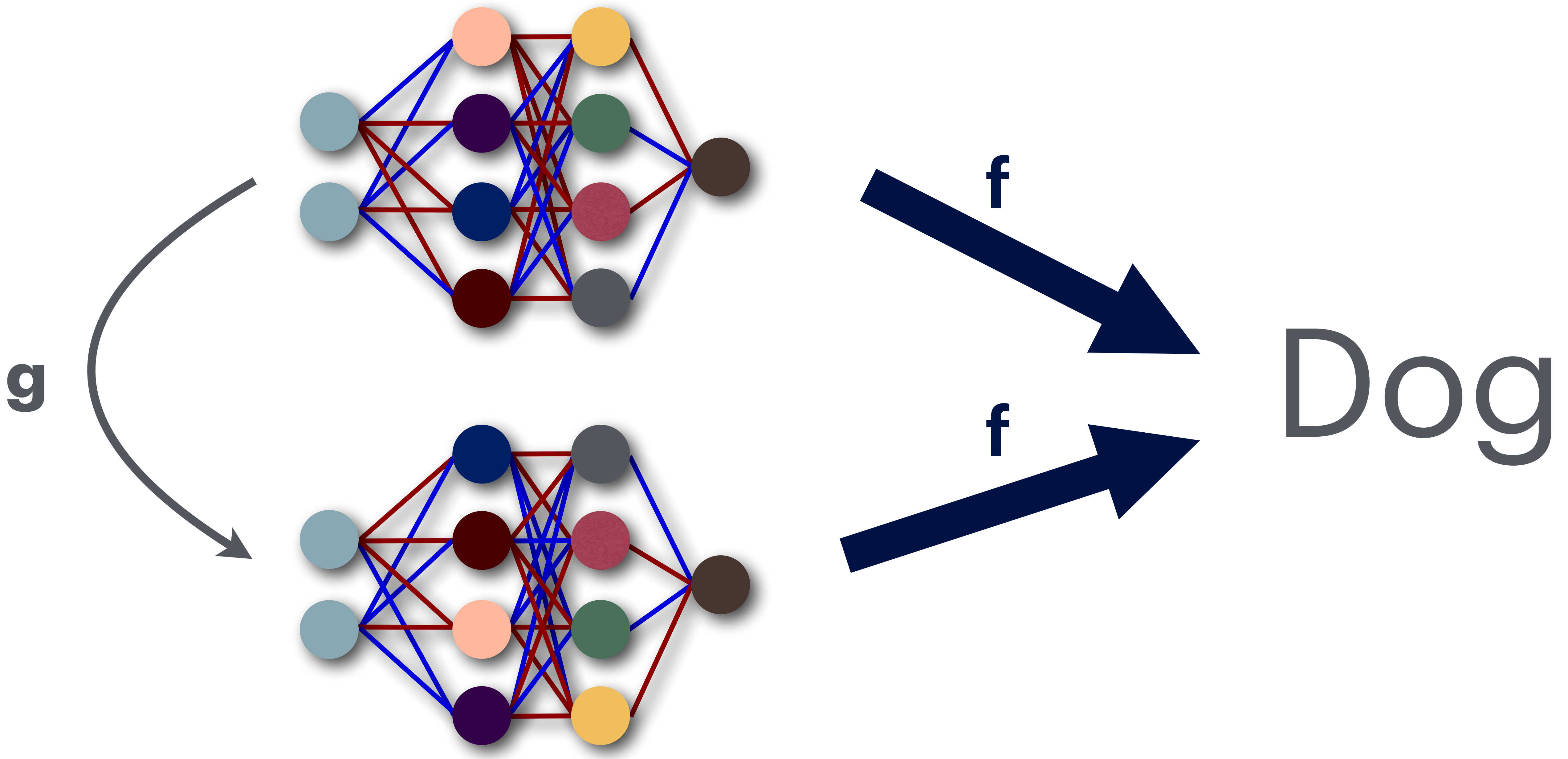


$$(\mathbf{P}_\ell \mathbf{Q}_\ell \mathbf{W}_\ell \mathbf{Q}_{\ell-1}^{-1} \mathbf{P}_{\ell-1}^{-1}, \mathbf{P}_\ell \mathbf{Q}_\ell \mathbf{b}_\ell)_{\ell=1}^{L} = \boldsymbol{\theta}' \simeq \boldsymbol{\theta} = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1}^{L}$$

Desired properties

- ***Invariant tasks:*** Our Metanetwork must be *invariant* to the ***permutation*** and ***scaling*** symmetries.

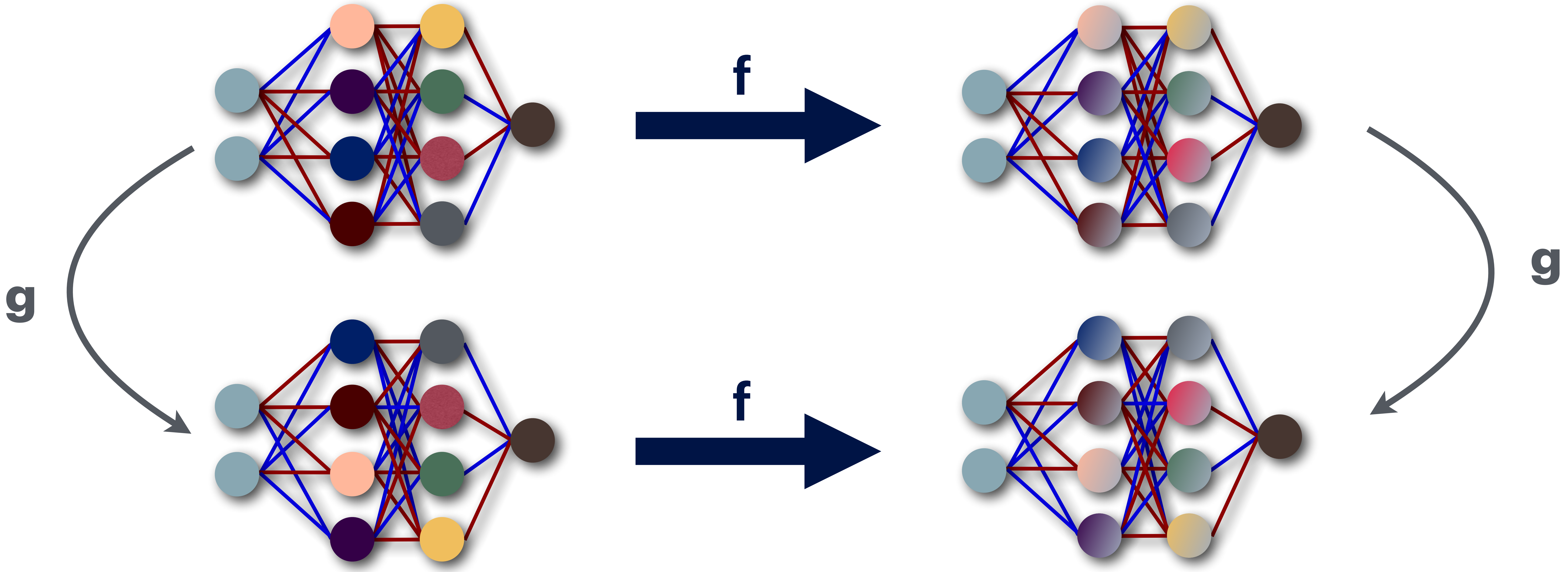
Map ***equivalent NNs*** to the ***same result***.



Desired properties

- ***Equivariant tasks:*** Our Metanetwork must be *equivariant* to the ***permutation*** and ***scaling*** symmetries.

Map ***equivalent NNs*** to ***equivalent NNs***.



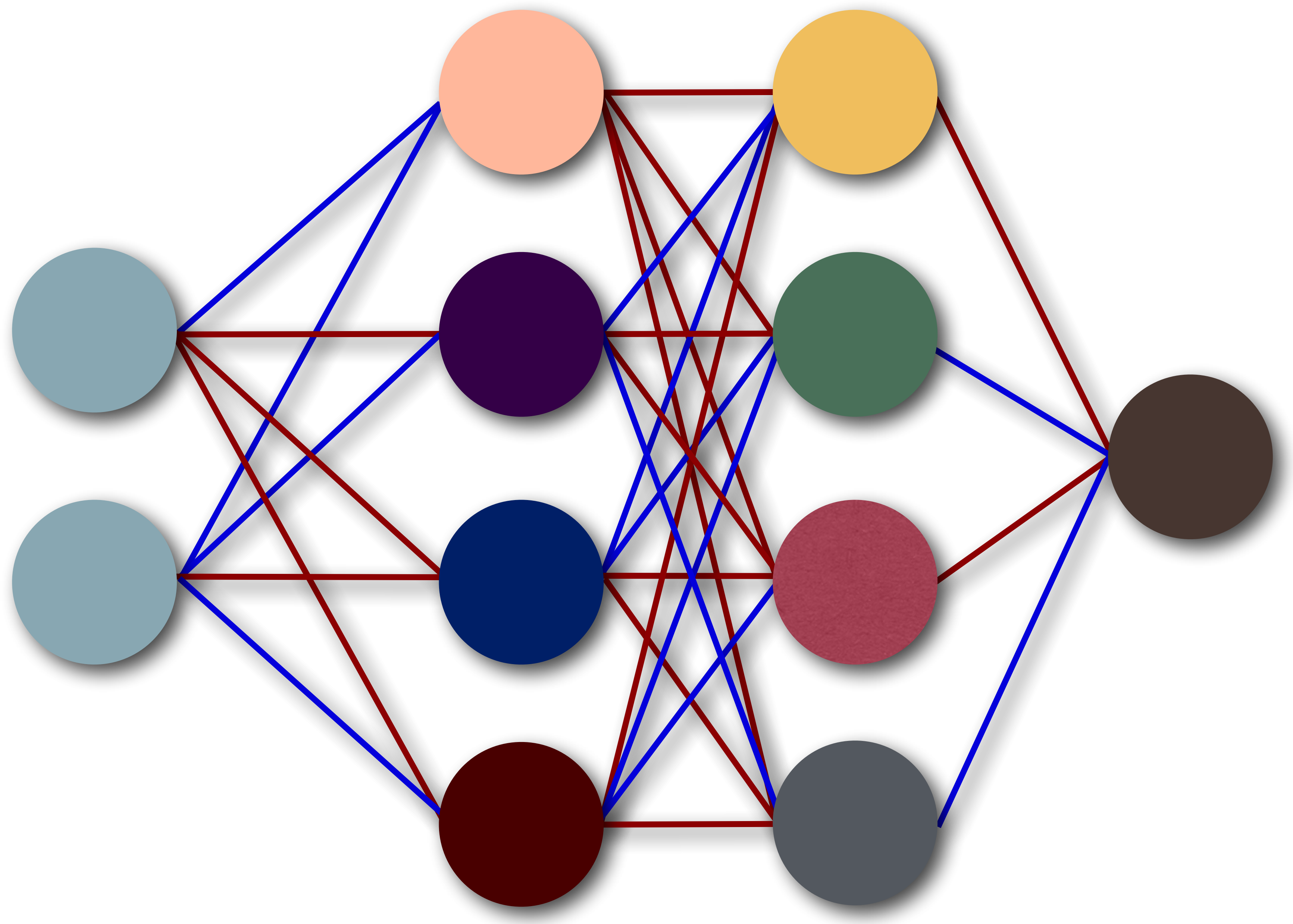
Scale Equivariant Graph Metanetworks

ScaleGMN

- Follows the ***local*** approach.
- Accounts for both ***permutation*** and ***scaling symmetries****.
- Extends the MPNN paradigm by designing ***scale equivariant MSG and UPD*** functions and a ***permutation*** and ***scale invariant READOUT*** function.

*which in various setups, are the only function-preserving symmetries.

Step 1: Graph Initialization (MLP)

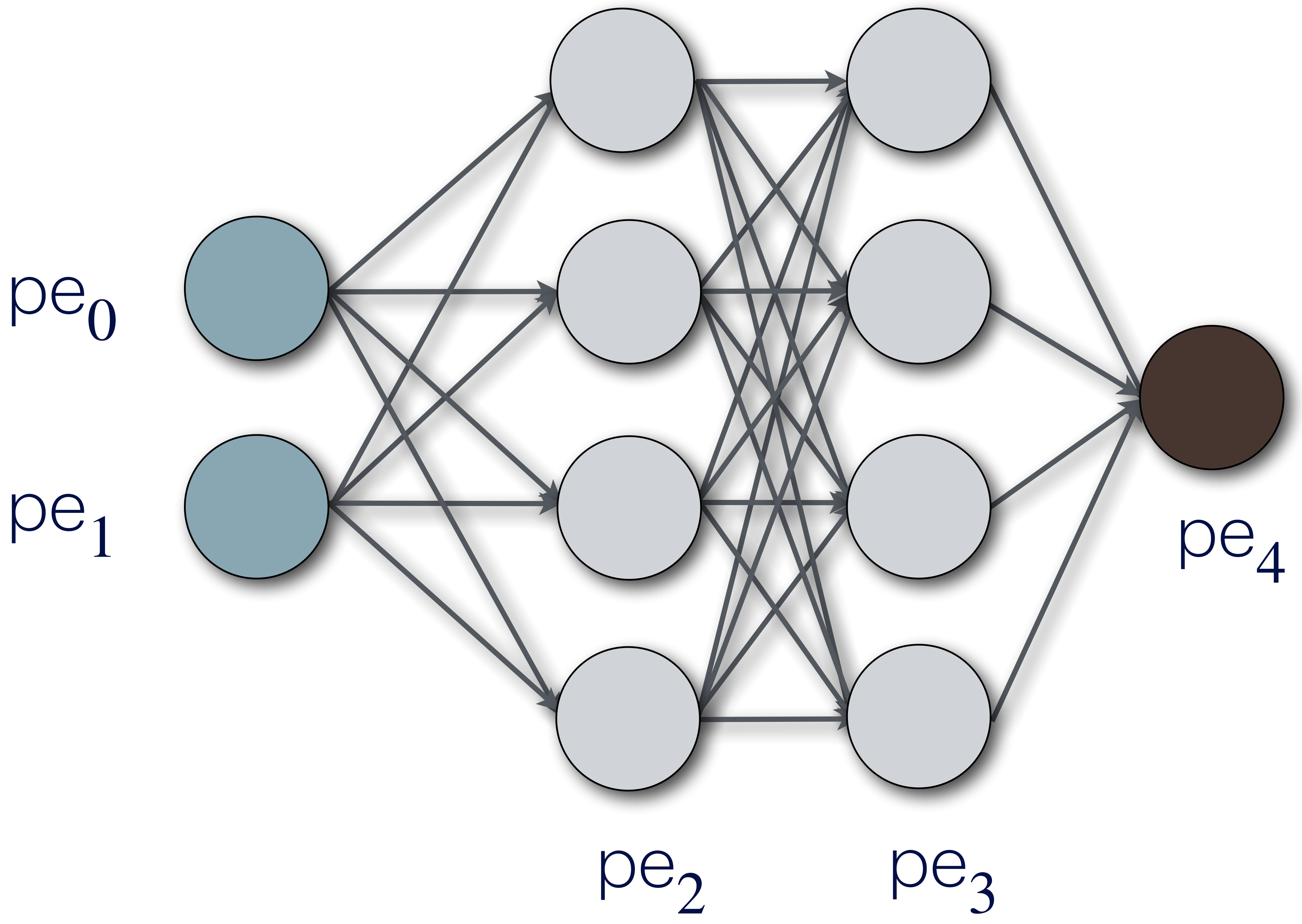


1. Graph $G(V, E, \mathbf{x}_V, \mathbf{x}_E)$
   - Node i : neuron i, node features $\mathbf{x}_V(i) = b(i)$
   - Edge (j,i): weight, edge features $\mathbf{x}_E(i,j) = W(i,j)$
2. Positional Encodings
3. Linear initialization of features

↓

Nodes and edges share same symmetries as biases and weights of input NN

Step 1: Graph Initialization (MLP)



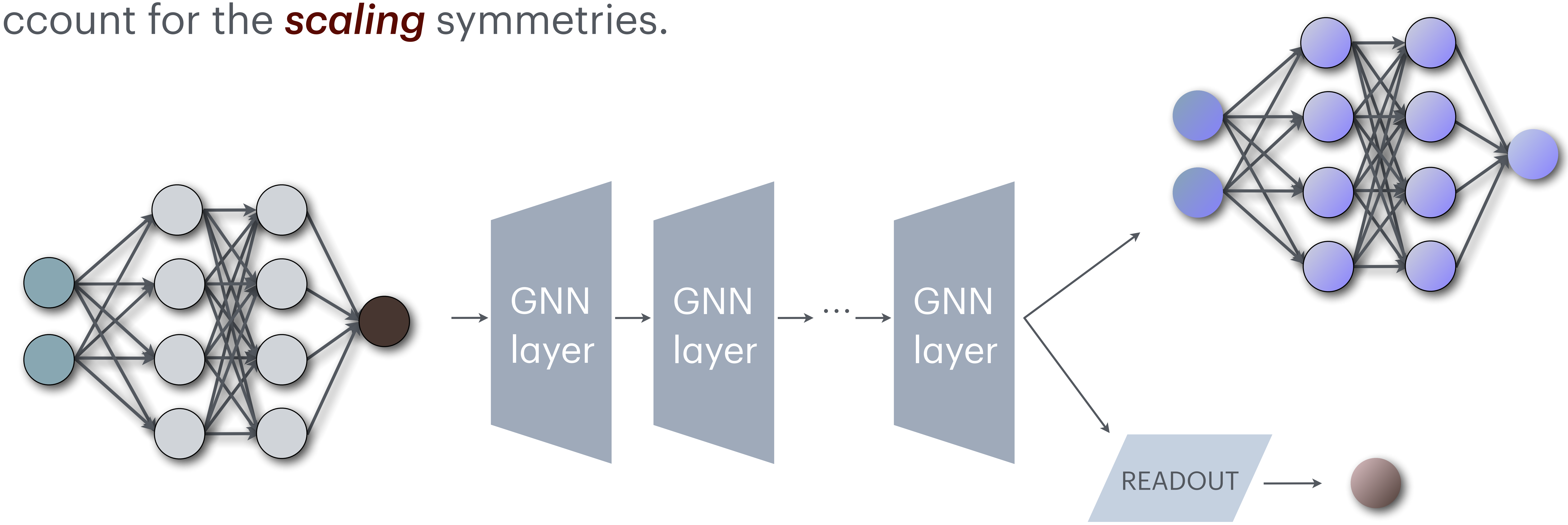
1. Graph $G(V, E, \mathbf{x}_V, \mathbf{x}_E)$
   - Node i : neuron i, node features $\mathbf{x}_V(i) = b(i)$
   - Edge (j,i): weight, edge features $\mathbf{x}_E(i,j) = W(i,j)$
2. Positional Encodings
3. Linear initialization of features

↓

Nodes and edges share same symmetries as biases and weights of input NN

Step 2: Message Passing

- GNN layers are by construction ***permutation*** equivariant.
- Hence, we only need to adapt the ***MSG***, ***UPD*** and ***READOUT*** functions to account for the ***scaling*** symmetries.



Achieving ***Scale*** Equivariance

Scale Invariant

ScaleInv

Scale Equivariant

ScaleEq

ReScale Equivariant*

ReScaleEq

*when scaled by different multipliers

ScaleGMN - 3 building blocks

Scale Invariant

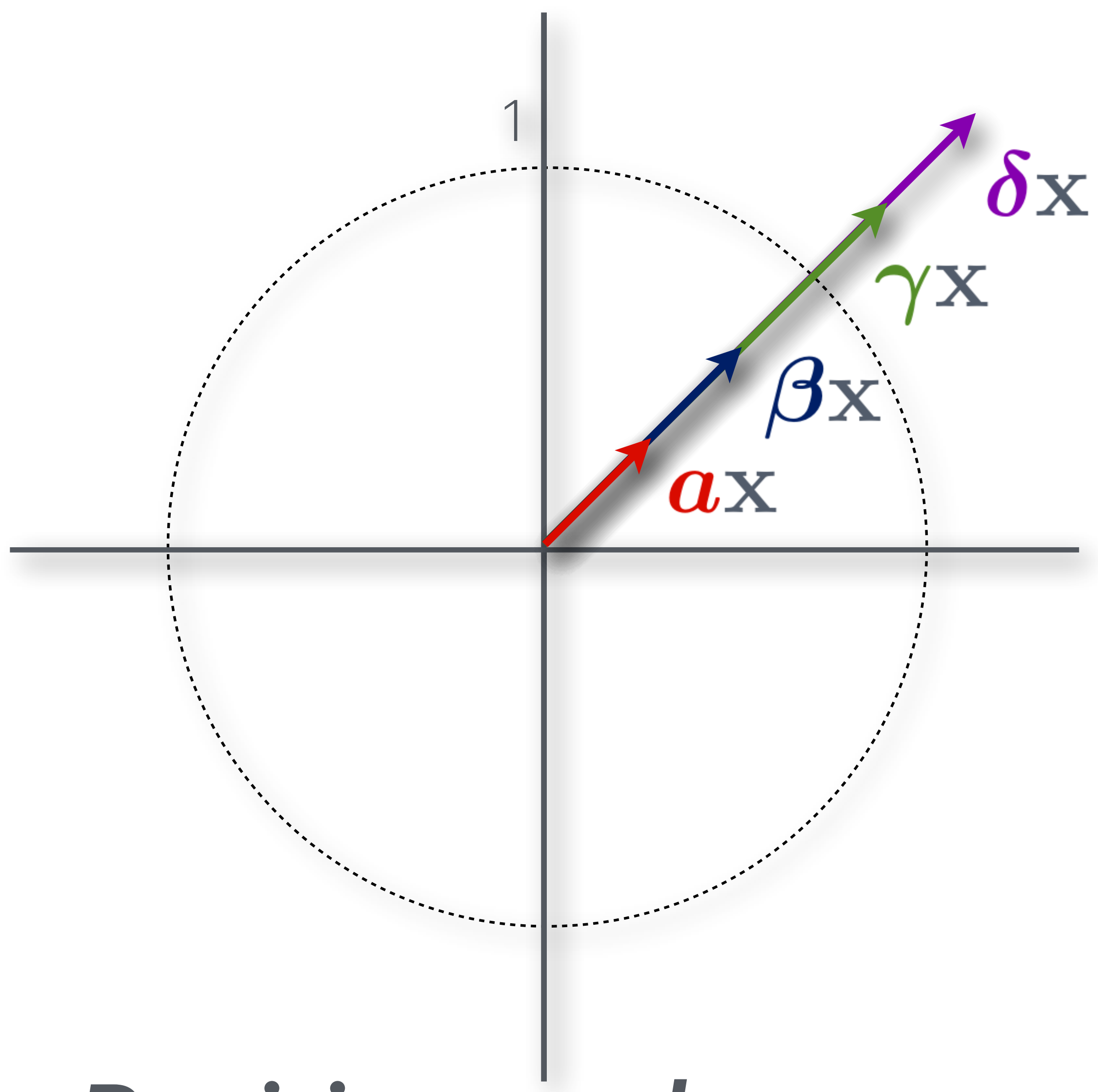
Scale Equivariant

ScaleEq

ReScale Equivariant*

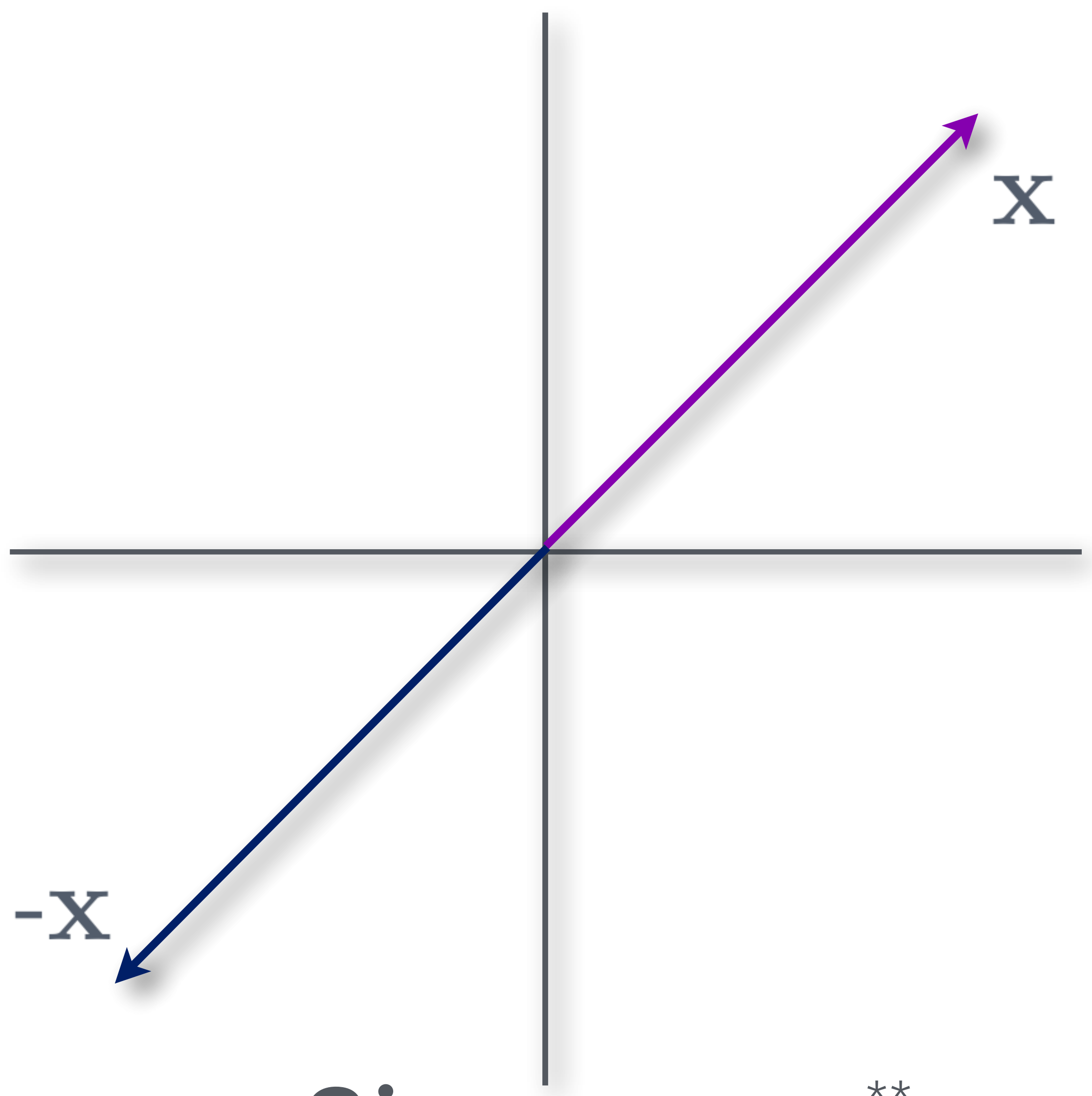
ReScaleEq

$$\text{ScaleInv}(\mathbf{x}) = f_1(\mathbf{x}) = \rho(\tilde{\mathbf{x}})$$



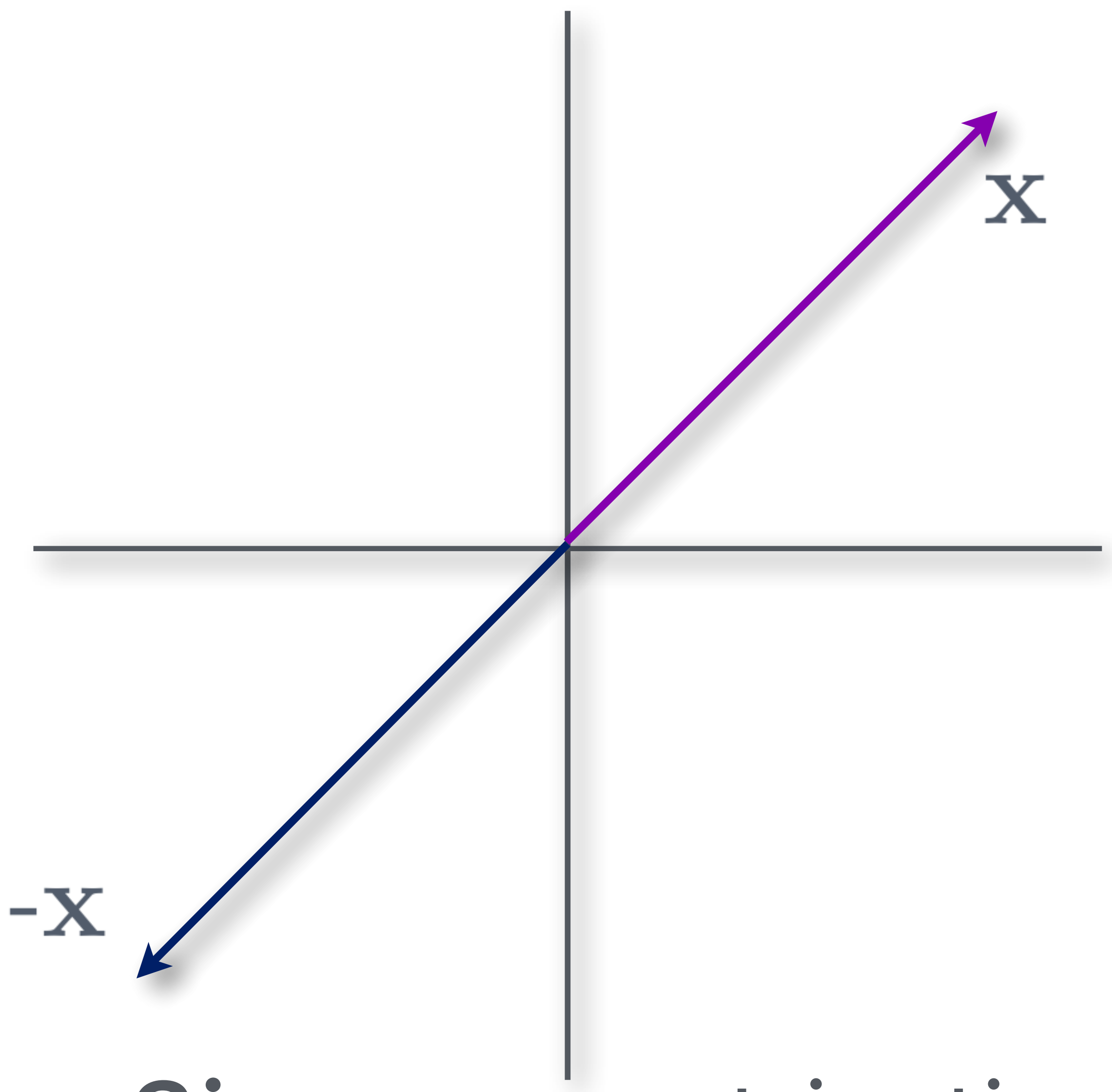
Positive scale canon

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$



Sign canon**

$$\tilde{\mathbf{x}}(i) = |\mathbf{x}(i)|$$



Sign symmetrization**

$$\tilde{\mathbf{x}} = \phi(\mathbf{x}) + \phi(-\mathbf{x})$$

Achieve invariance using either:

- ***canonicalization***
- ***symmetrization***

*when scaled by different multipliers

** Lim, Derek, et al. ICLR 2023

ScaleGMN - 3 building blocks

Scale Invariant

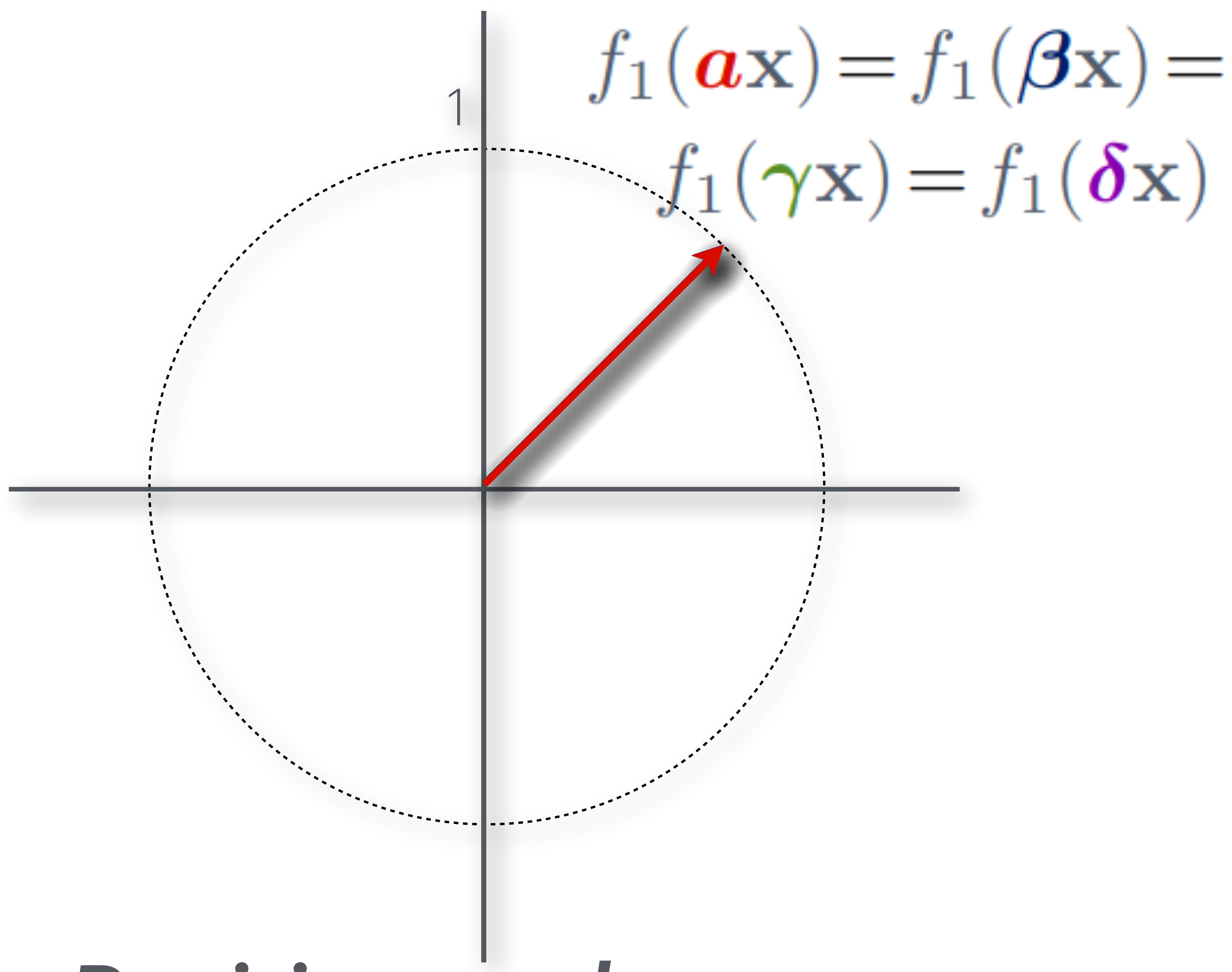
Scale Equivariant

ScaleEq

ReScale Equivariant*

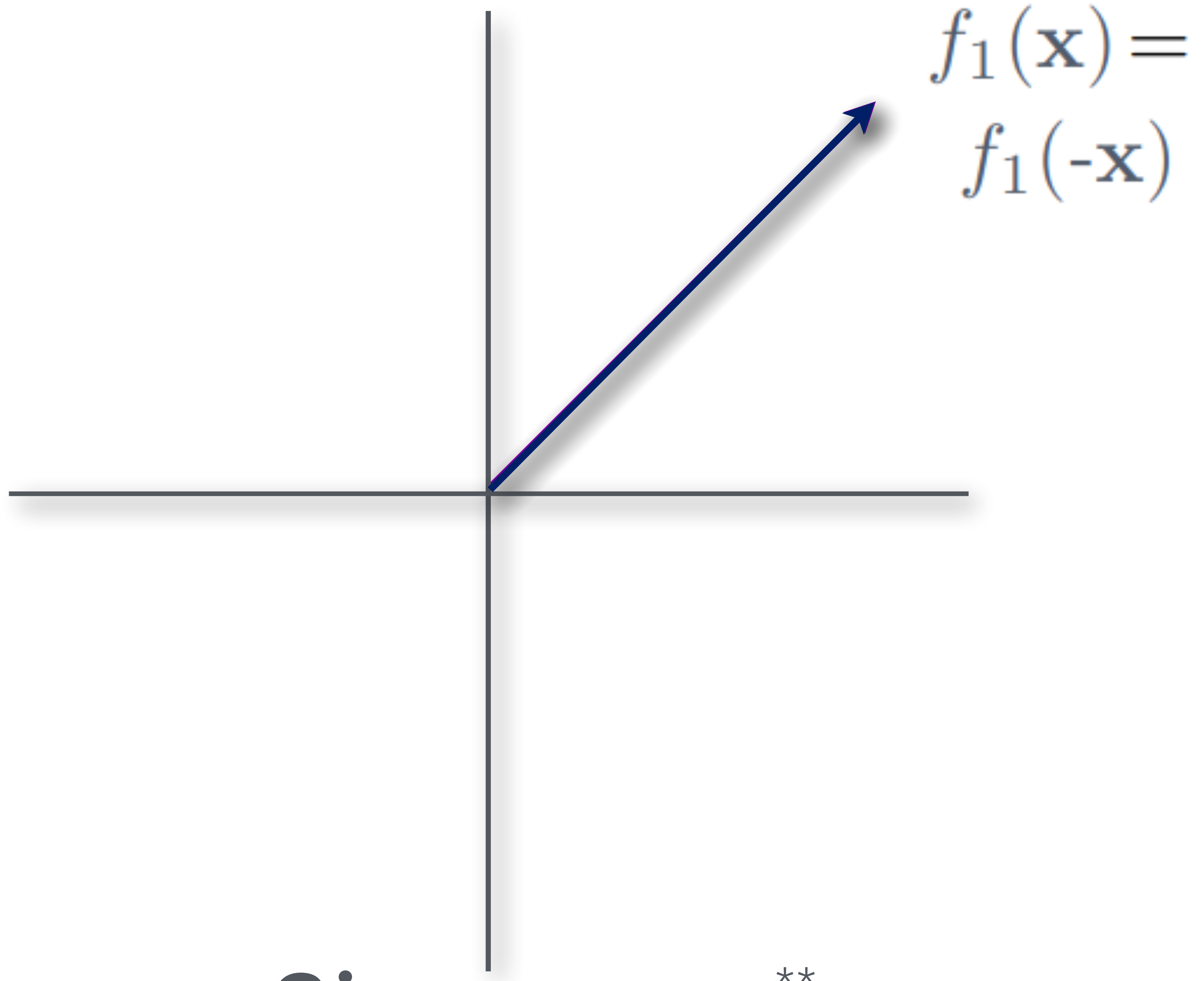
ReScaleEq

$$\text{ScaleInv}(\mathbf{x}) = f_1(\mathbf{x}) = \rho(\tilde{\mathbf{x}})$$



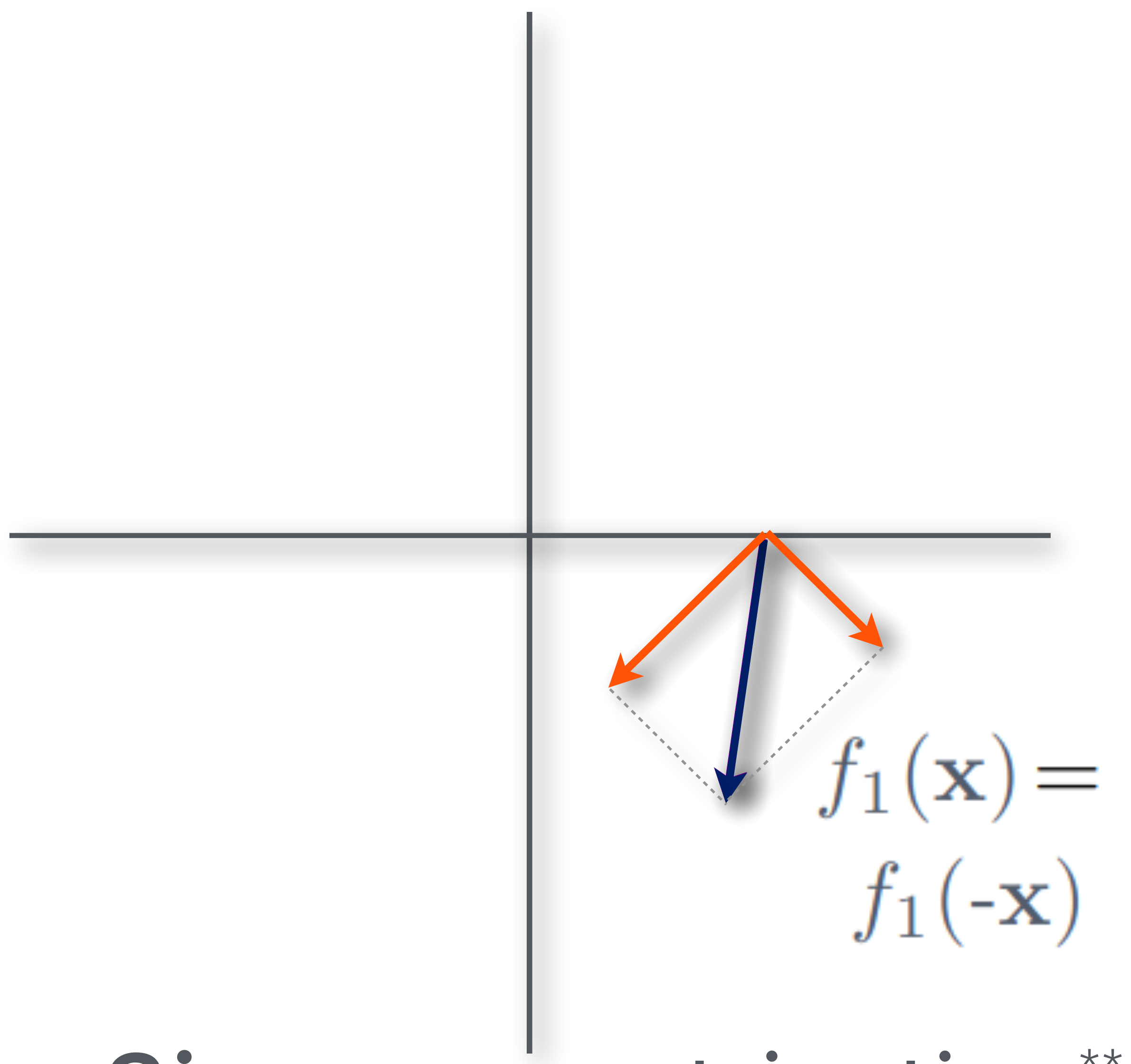
Positive scale canon

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$



Sign canon**

$$\tilde{\mathbf{x}}(i) = |\mathbf{x}(i)|$$



Sign symmetrization**

$$\tilde{\mathbf{x}} = \phi(\mathbf{x}) + \phi(-\mathbf{x})$$

Achieve invariance using either:

- ***canonicalization***
- ***symmetrization***

*when scaled by different multipliers

** Lim, Derek, et al. ICLR 2023

ScaleGMN - 3 building blocks

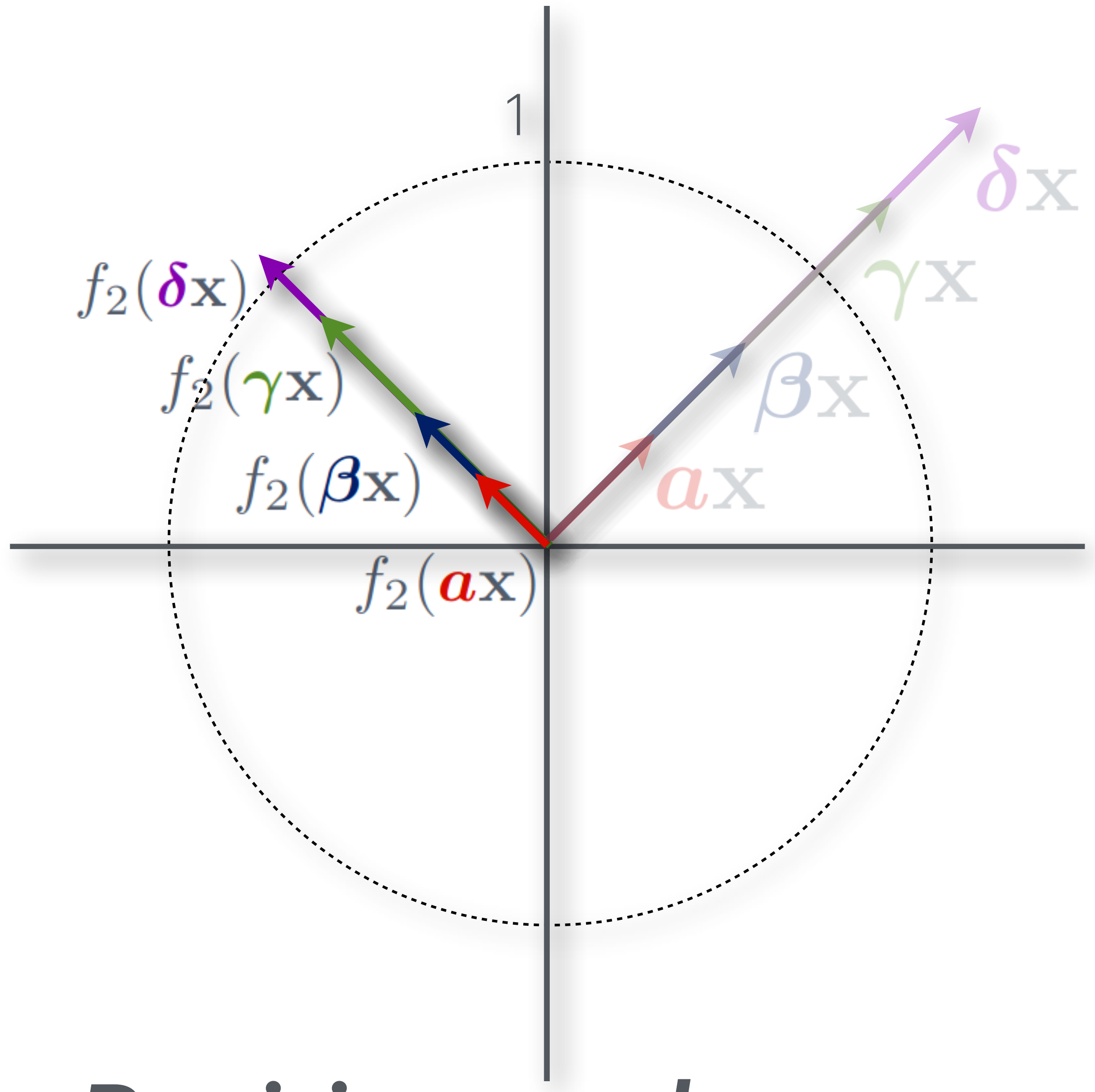
Scale Invariant

Scale Equivariant

ReScale Equivariant*

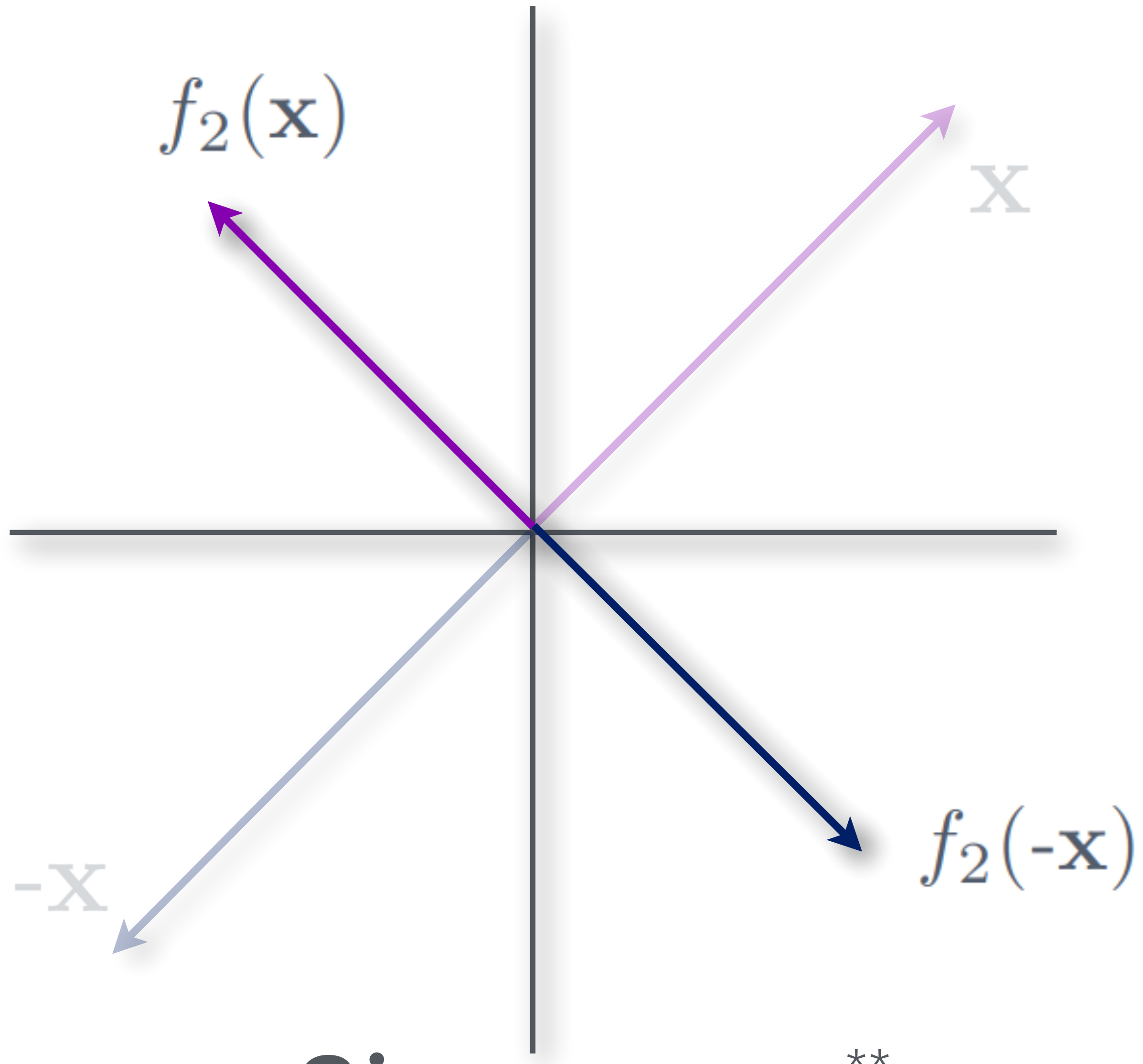
ReScaleEq

$$\text{ScaleEq}(\mathbf{x}) = f_2(\mathbf{x}) = \boldsymbol{\Gamma}\mathbf{x} \odot \text{ScaleInv}(\mathbf{x})$$ **



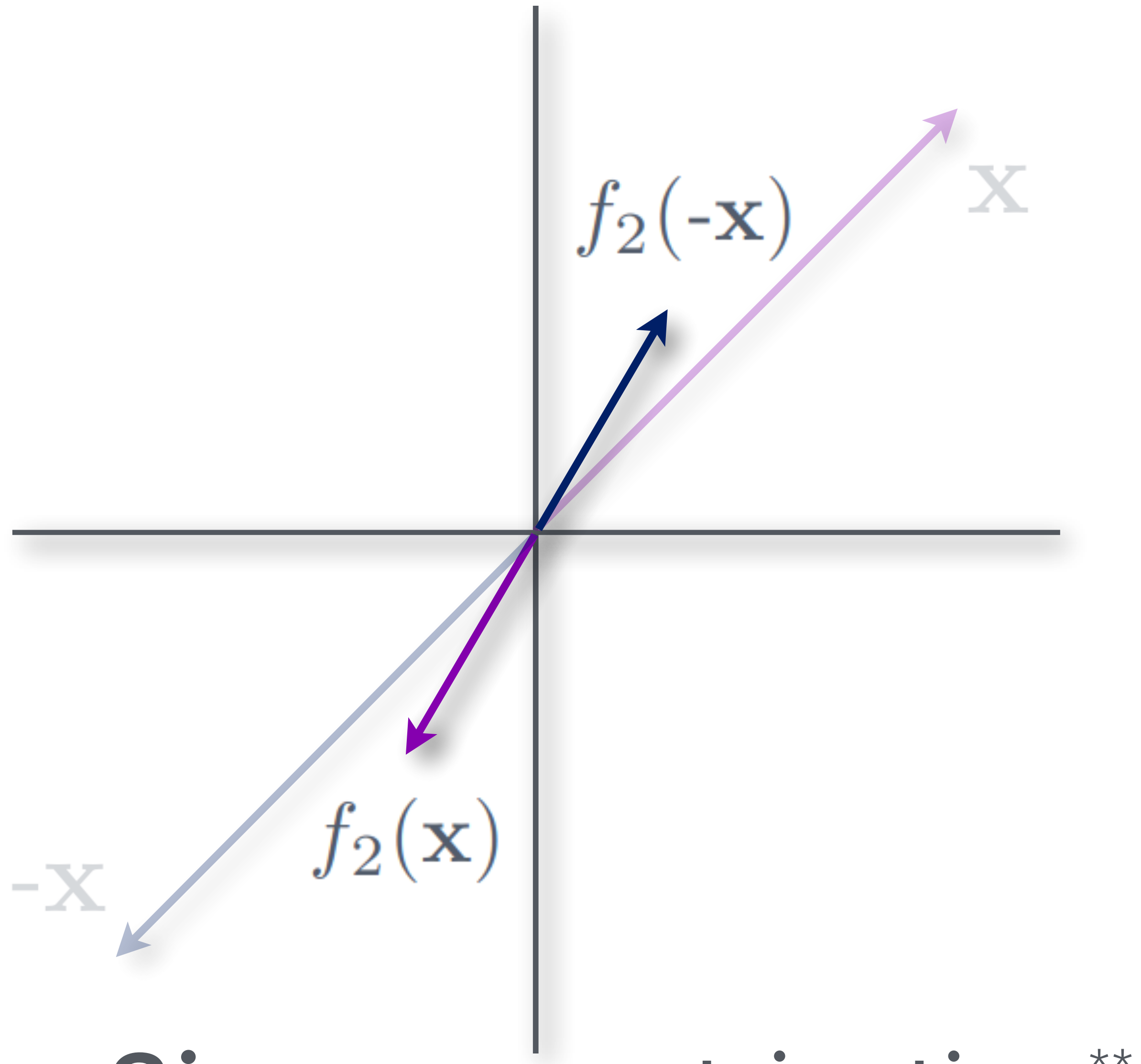
Positive scale canon

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$



Sign canon**

$$\tilde{\mathbf{x}}(i) = |\mathbf{x}(i)|$$



Sign symmetrization**

$$\tilde{\mathbf{x}} = \phi(\mathbf{x}) + \phi(-\mathbf{x})$$

*when scaled by different multipliers

** Lim, Derek, et al. NeurIPS 2024

ScaleGMN - 3 building blocks

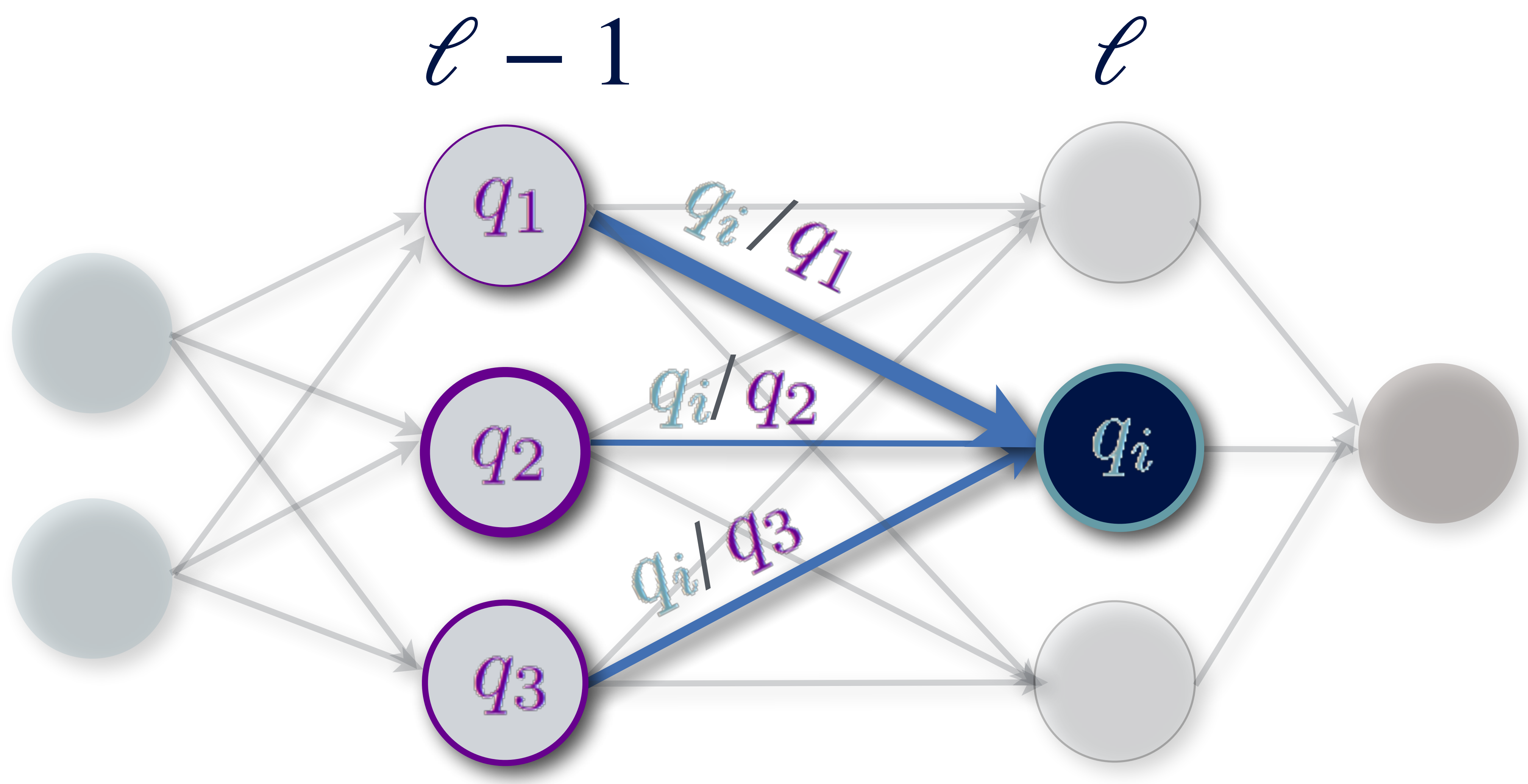
Scale Invariant

Scale Equivariant

ReScale Equivariant*

$$\text{ReScaleEq}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{\Gamma}_1\mathbf{x}_1 \odot \mathbf{\Gamma}_2\mathbf{x}_2$$

Input vectors of the MSG are ***scaled*** by different multipliers:



$$\text{MSG}(h_i, h_j, e_{ji})$$

g

$$\text{MSG}(\underbrace{q_i h_i}_{\text{central vertex}}, \underbrace{q_j h_j}_{\text{neighbor}}, \underbrace{\frac{q_i}{q_j} e_{ji}}_{\text{edge}})$$

The output should only be scaled by q_i .

*when scaled by different multipliers

ScaleGMN - 3 building blocks

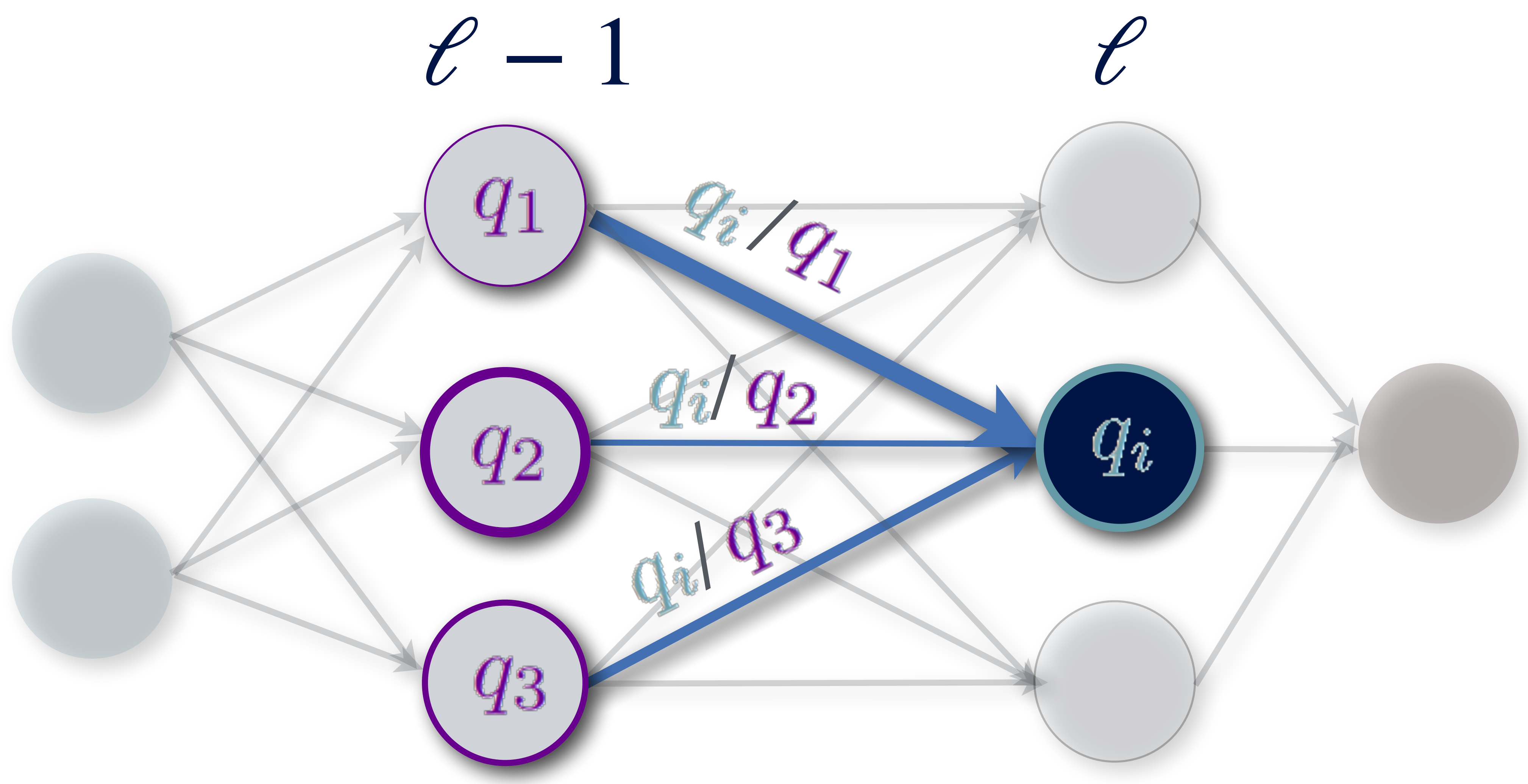
Scale Invariant

Scale Equivariant

ReScale Equivariant*

$$\text{ReScaleEq}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{\Gamma}_1\mathbf{x}_1 \odot \mathbf{\Gamma}_2\mathbf{x}_2$$

ReScaleEq : equivariant to the *product* of the multipliers.



$$\text{MSG}(h_i, \text{ReScaleEq}(h_j, e_{ji}))$$

g

$$\text{MSG}(q_i h_i, \text{ReScaleEq}(q_j h_j, \frac{q_i}{q_j} e_{ji}))$$

$$=$$

$$\text{MSG}(q_i h_i, q_i \text{ReScaleEq}(h_j, e_{ji}))$$

*when scaled by different multipliers

ScaleGMN - Bidirectional variant

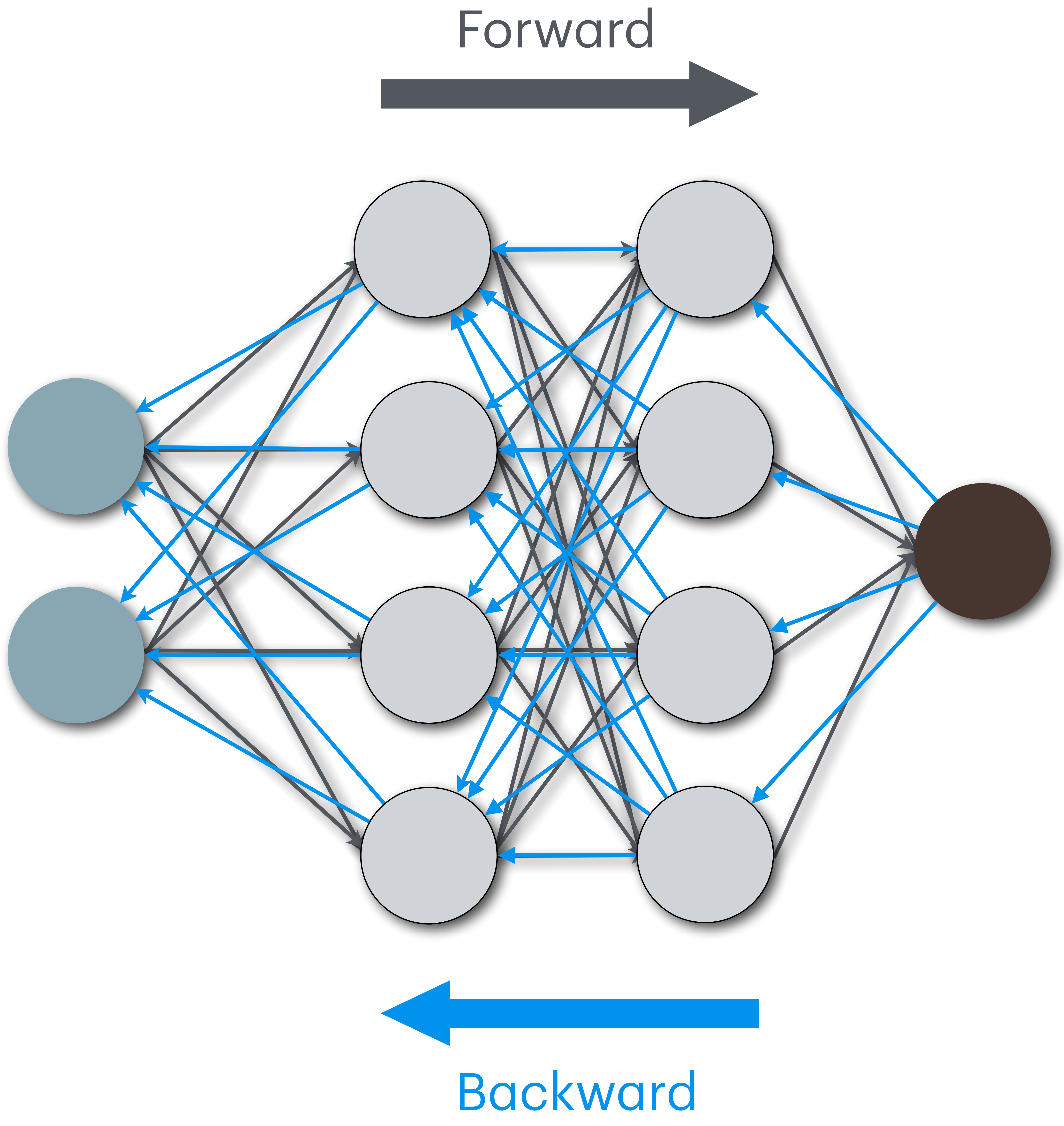
In the forward variant vertices receive information *only from previous layers*:

Detrimental, especially for equivariant tasks.

Solution:

1. Add *backward* edges.

2. Extend to ScaleGMN bidirectional (not straightforward due to multiple ***scalings***).



Theoretical outcomes

Proposition

ScaleGMN is ***permutation*** & ***scale*** ***equivariant***.

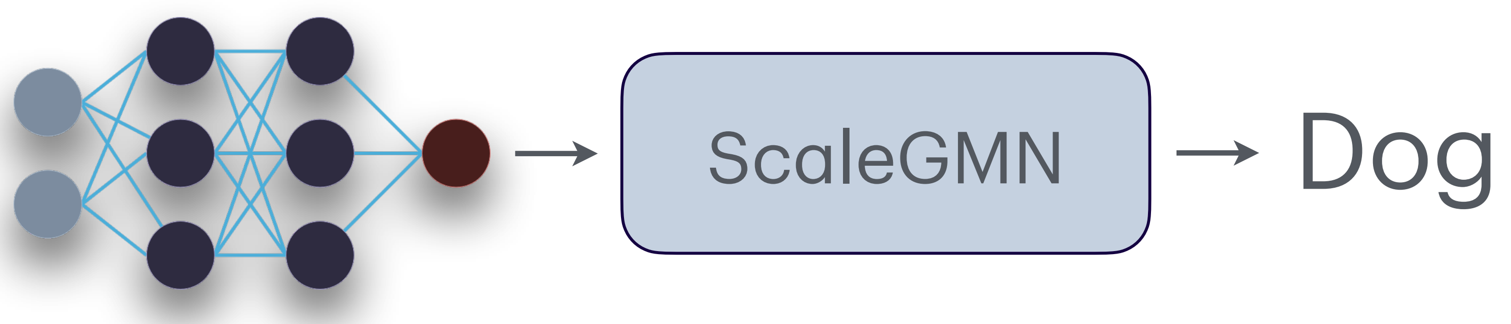
Theorem

Bidirectional ScaleGMN can ***simulate the forward and backward pass*** of any input FFNN.*

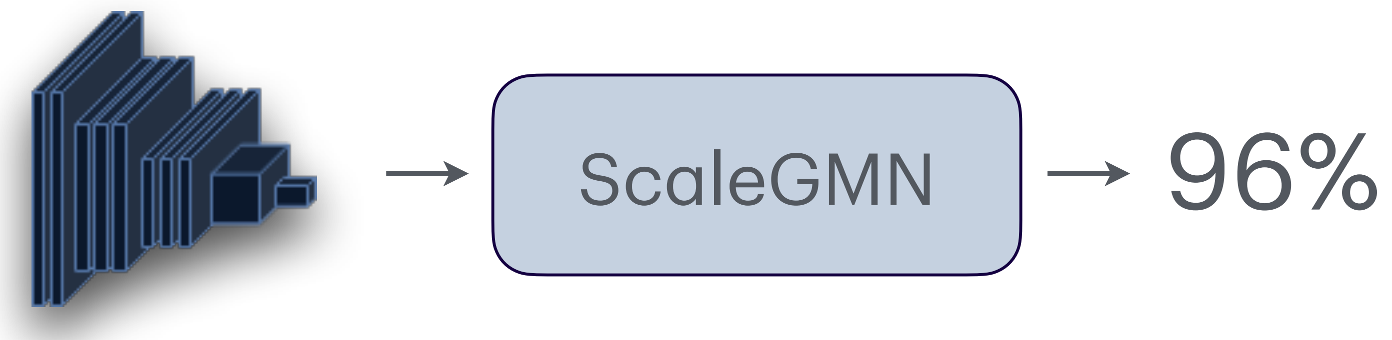
*under mild assumptions

Experiments

1. *INR Classification*



2. *Generalization prediction*



3. *INR Editing*

Experiments

1. *Classify INRs* representing images.

● first ● second ● third

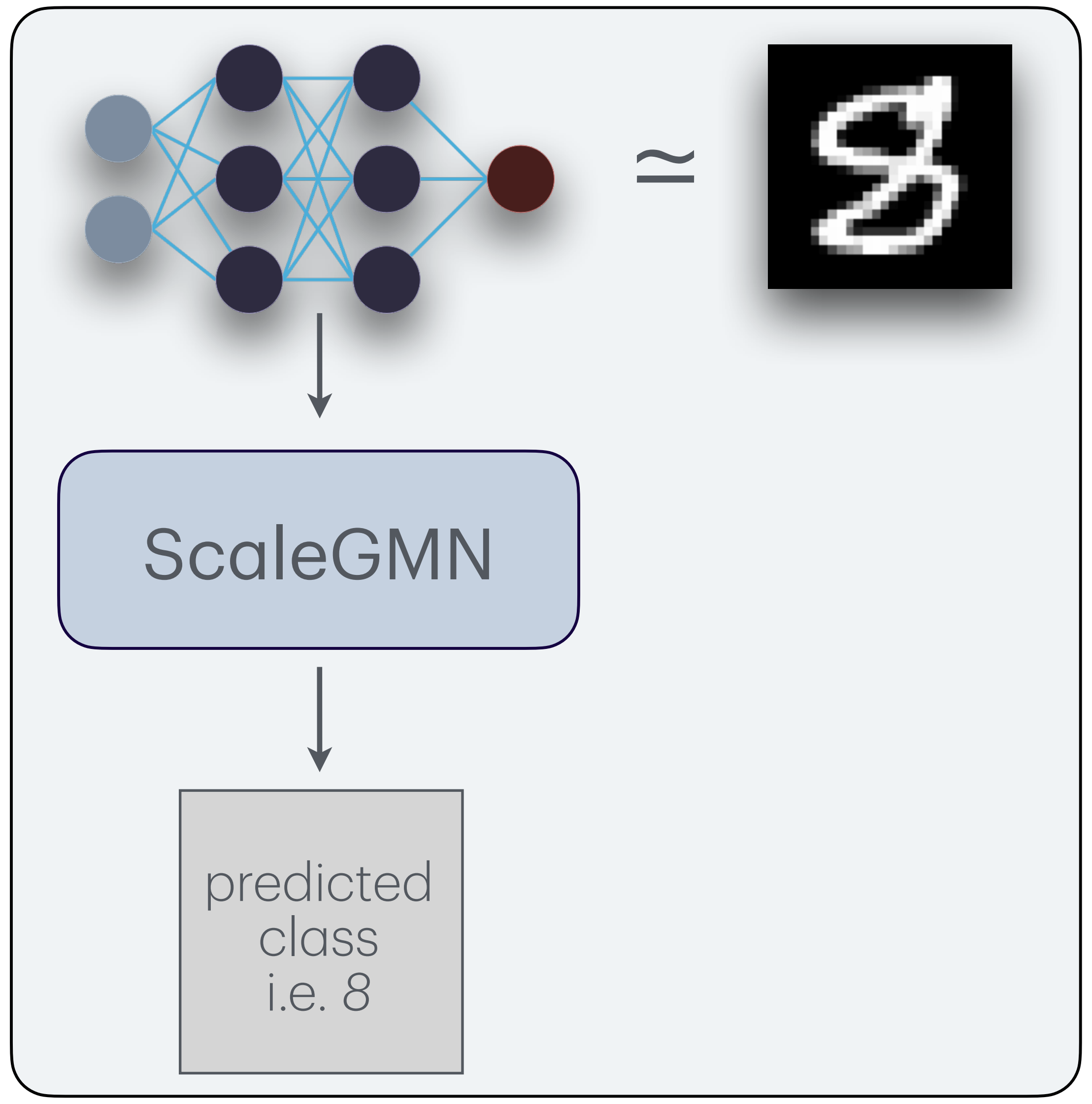
Method	MNIST	F-MNIST	CIFAR-10	Augmented CIFAR-10
MLP	17.55 ± 0.01	19.91 ± 0.47	11.38 ± 0.34*	16.90 ± 0.25
Inr2Vec [47]	23.69 ± 0.10	22.33 ± 0.41	-	-
DWS [54]	85.71 ± 0.57	67.06 ± 0.29	34.45 ± 0.42	41.27 ± 0.026
NFN_{NP} [85]	78.50 ± 0.23*	68.19 ± 0.28*	33.41 ± 0.01*	46.60 ± 0.07
NFN_{HNP} [85]	79.11 ± 0.84*	68.94 ± 0.64*	28.64 ± 0.07*	44.10 ± 0.47
NG-GNN [33]	91.40 ± 0.60	68.00 ± 0.20	36.04 ± 0.44*	45.70 ± 0.20*
ScaleGMN (Ours)	96.57 ± 0.10	80.46 ± 0.32	36.43 ± 0.41	56.62 ± 0.24
ScaleGMN-B (Ours)	**96.59 ± 0.24**	**80.78 ± 0.16**	**38.82 ± 0.10**	**56.95 ± 0.57**

non equiv.

perm. equiv.

perm. & scale equiv.

Invariant task



ScaleGMN outperforms all baselines, *without resorting to additional techniques* such as probe features, advanced architectures or extra training samples.

Experiments

2. *Predict test accuracy* of trained CNNs.

● first ● second ● third

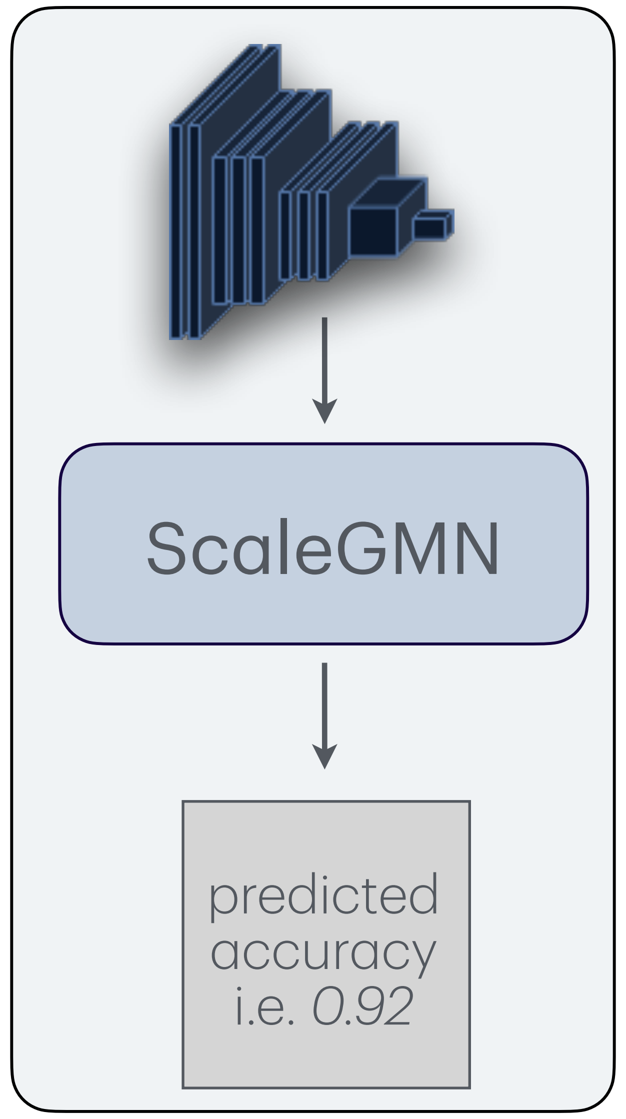
Method	CIFAR-10-GS ReLU	SVHN-GS ReLU	CIFAR-10-GS Tanh	SVHN-GS Tanh	CIFAR-10-GS both act.
StatNN [74]	0.9140 ± 0.001	0.8463 ± 0.004	0.9140 ± 0.000	0.8440 ± 0.001	0.915 ± 0.002
NFN_{NP} [85]	0.9190 ± 0.010	0.8586 ± 0.003	0.9251 ± 0.001	0.8580 ± 0.004	0.922 ± 0.001
NFN_{HNP} [85]	0.9270 ± 0.001	0.8636 ± 0.002	0.9339 ± 0.000	0.8586 ± 0.004	0.934 ± 0.001
NG-GNN [33]	0.9010 ± 0.060	0.8549 ± 0.002	0.9340 ± 0.001	0.8620 ± 0.003	0.931 ± 0.002
ScaleGMN (Ours)	0.9276 ± 0.002	**0.8689 ± 0.003**	0.9418± 0.005	**0.8736 ± 0.003**	0.941 ± 0.006
ScaleGMN-B (Ours)	**0.9282 ± 0.003**	0.8651 ± 0.001	**0.9425 ± 0.004**	0.8655 ± 0.004	**0.941 ± 0.000**

non equiv.

perm. equiv.

perm. & scale equiv.

Invariant task



Evaluate ScaleGMN on:

1. Each symmetry individually (ReLU: positive scale, Tanh: sign)
2. ***Heterogeneous*** activation functions

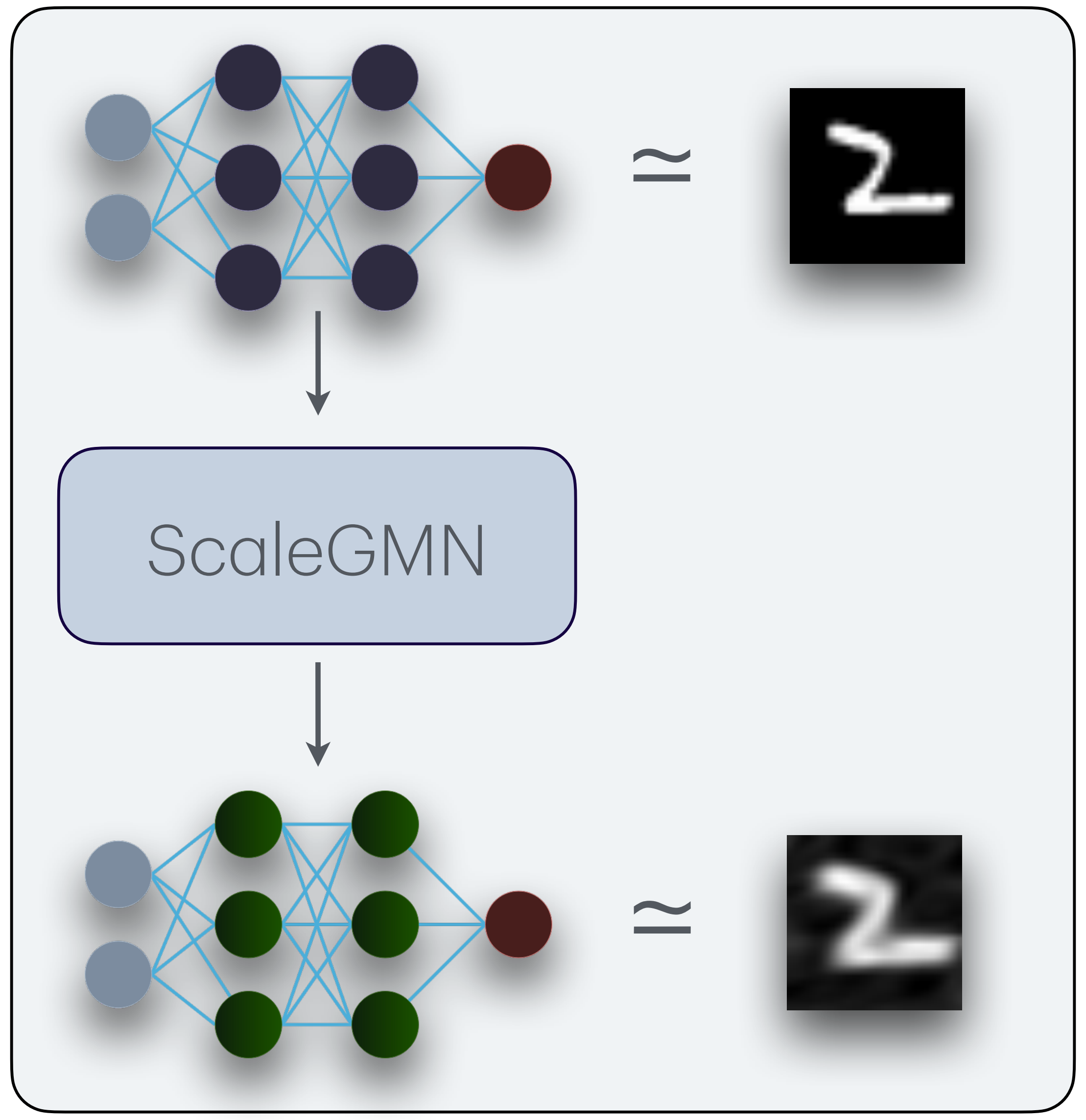
Experiments

3. *INR Editing*: Dilate digits of the MNIST INR dataset.

● first ● second ● third

Method	MSE in 10^{-2}	
MLP	5.35 ± 0.00	non equiv.
DWS [54]	2.58 ± 0.00	perm. equiv.
NFN_{NP} [85]	2.55 ± 0.00	perm. equiv.
NFN_{HNP} [85]	2.65 ± 0.01	perm. equiv.
NG-GNN-0 [33]	2.38 ± 0.02	perm. equiv.
NG-GNN-64 [33]	2.06 ± 0.01	perm. equiv.
ScaleGMN (Ours)	2.56 ± 0.03	perm. & scale equiv.
ScaleGMN-B (Ours)	$\mathbf{1.89 \pm 0.00}$	perm. & scale equiv.

Equivariant task



- ***Bidirectional*** variant performs significantly better than the forward one.
- Best test loss was achieved when *increasing the depth of ScaleGMN-B*. (validates previous theorem)

Takeaways

ScaleGMN:

1. introduces a strong inductive bias: ***accounting for function-preserving scaling*** symmetries arising from ***activation functions***.

2. can be applied to NNs with ***various*** (heterogeneous) activation functions.

3. enjoys desirable ***theoretical guarantees***.

4. ***empirically demonstrates the significance*** of scaling symmetries.

Want to learn more? Find us in the poster session!

- *Poster Session 5*
- *Poster #3010*