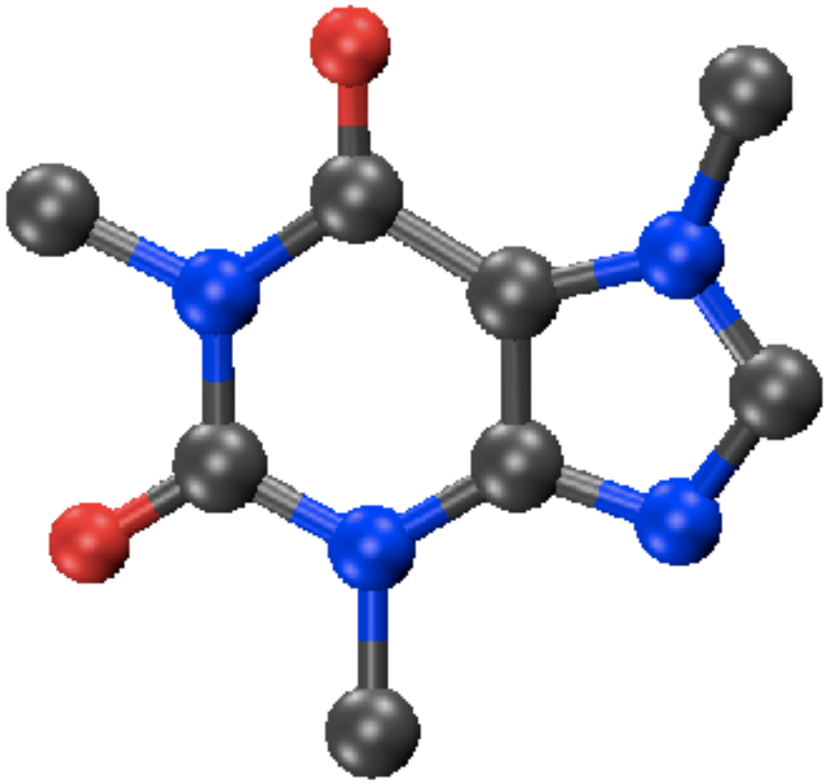
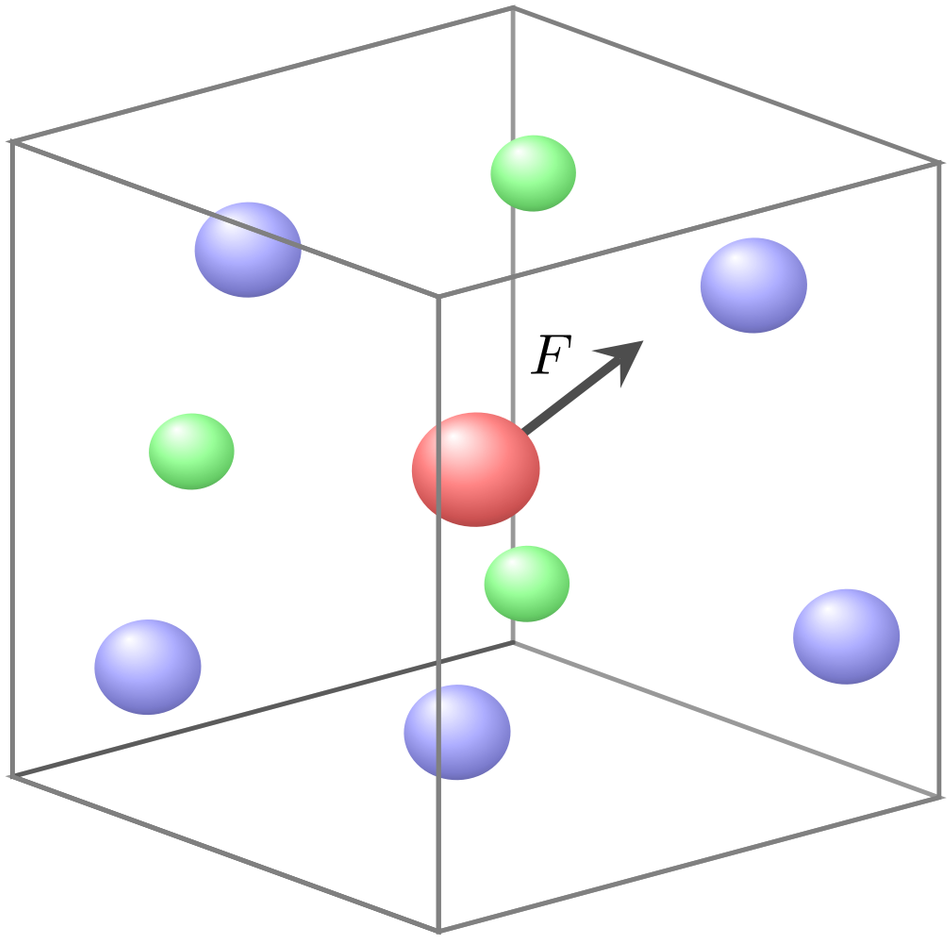
Covariant neural networks for learning from graphs, molecules or (almost) anything else

Risi Kondor
The University of Chicago



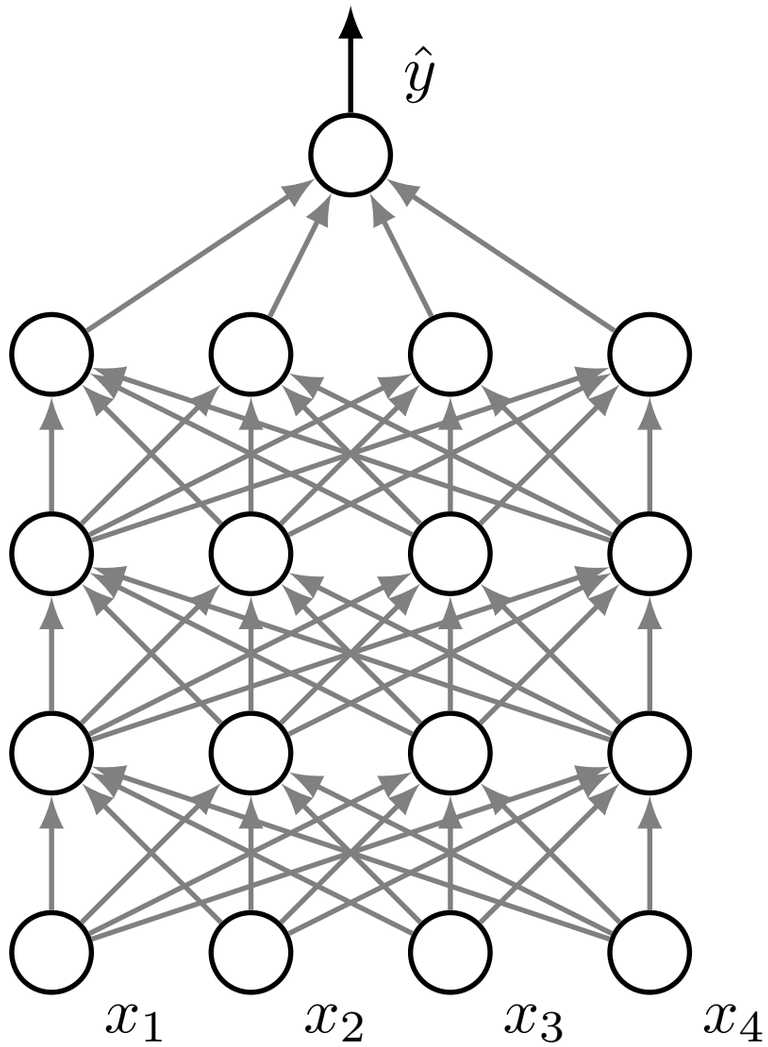
$\phi(G)$



$F(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m)$

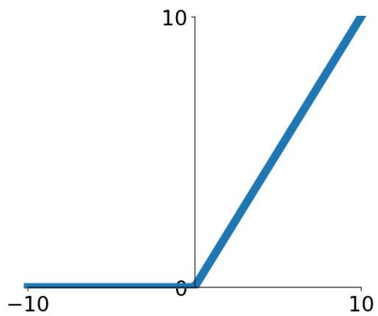
Feed-forward Neural Networks

$$f_{\text{out}} = \sigma\left(\sum_i w_i f_{\text{in}}^{(i)} + b\right)$$

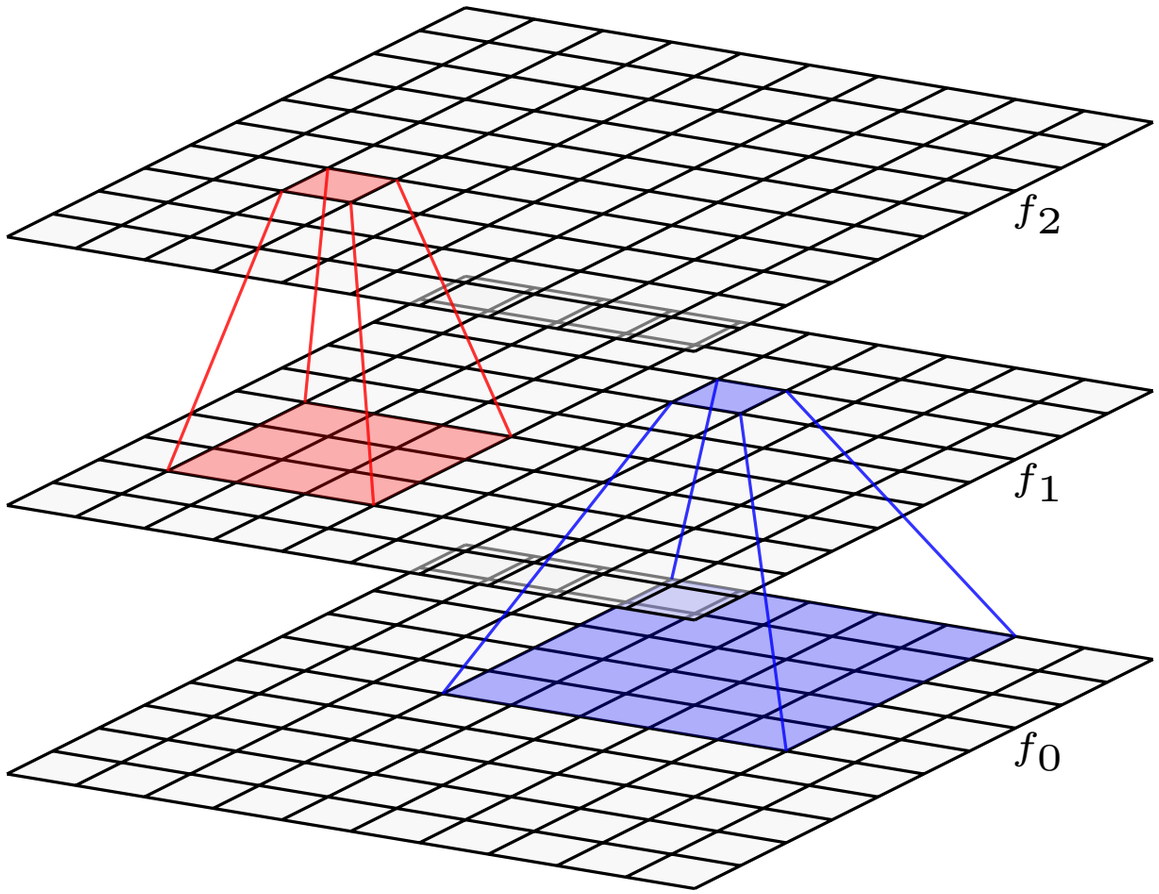


Common choice of nonlinearity:

$$\sigma_{\text{ReLU}}(z) = \max(0, x)$$

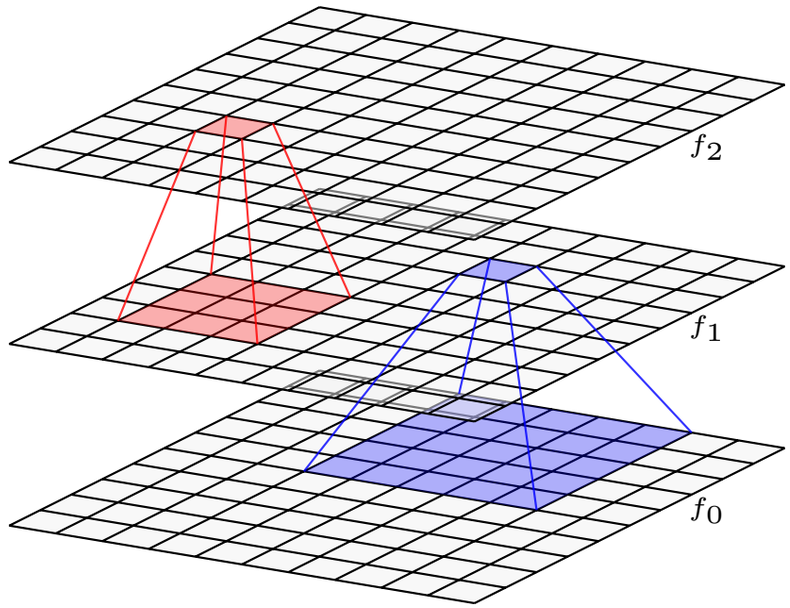


Convolutional Neural Networks



$$f_0 \longmapsto \phi_1(f_0) \overset{\sigma}{\longmapsto} f_1 \longmapsto \phi_2(f_1) \overset{\sigma}{\longmapsto} f_2 \longmapsto \ldots \longmapsto \phi_L(f_{\ell-1}) \longmapsto f_L$$

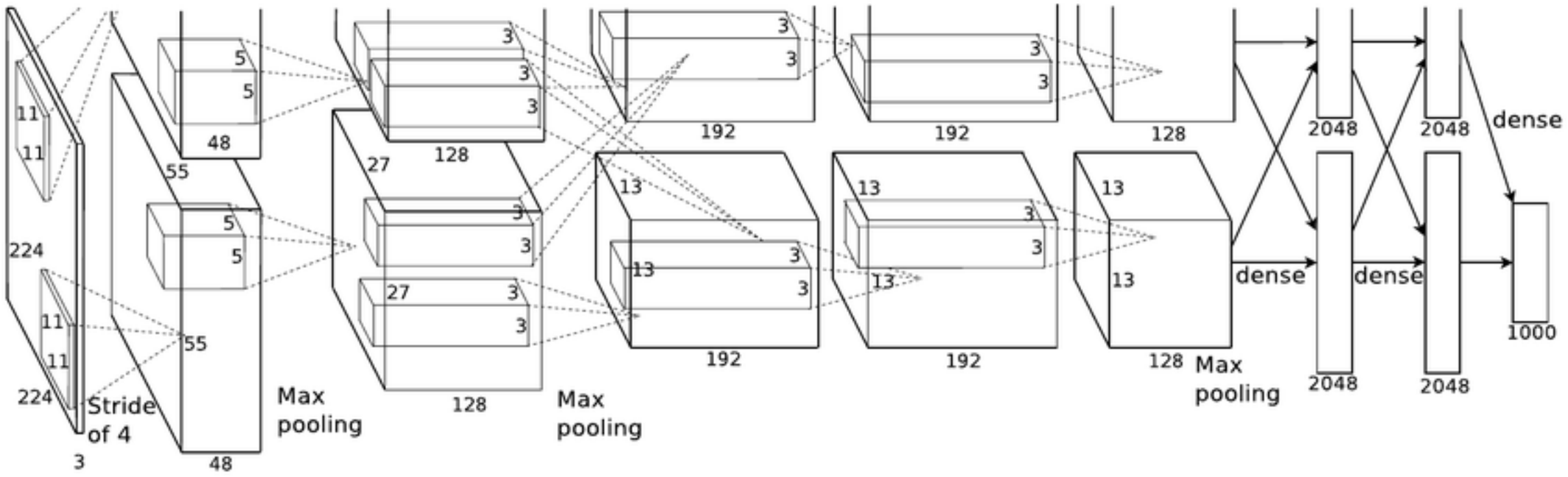
Convolutional Neural Networks



$$\phi_\ell(f_{\ell-1}) = (f_{\ell-1} * g_\ell)(x) = \sum_{y \in \mathbb{Z}^2} f_{\ell-1}(x-y)\, g_\ell(y)$$

Filter at layer ℓ

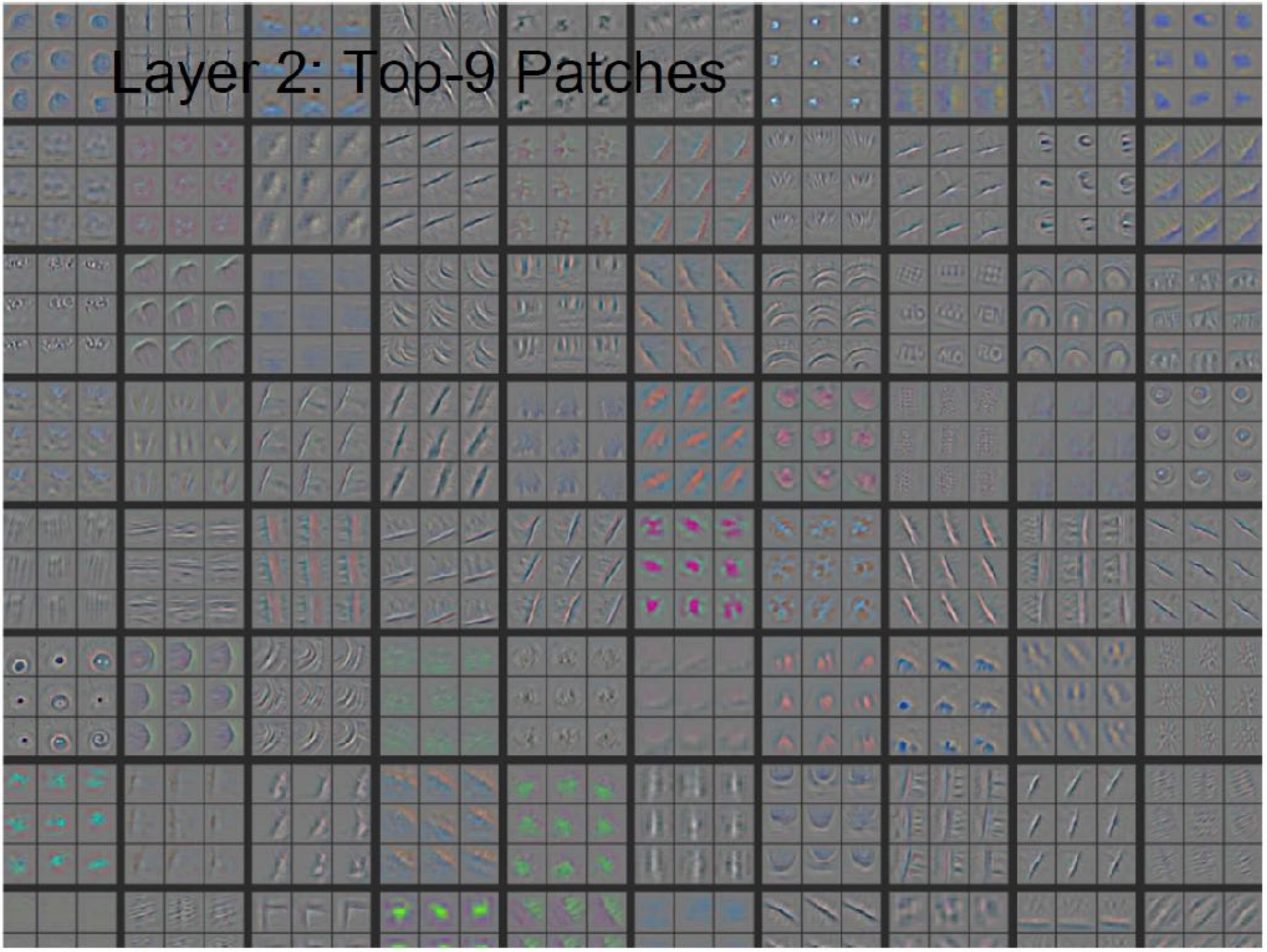
Convolutional Neural Networks



$$f_\ell(x) = \sigma\left(\sum_{\mathbf{y}} f_{\ell-1}(\mathbf{x}-\mathbf{y})\,\chi(\mathbf{y}) + b\right)$$

[LeCun et al, 1989; Krizhevsky, Sutskever & Hinton, 2012]

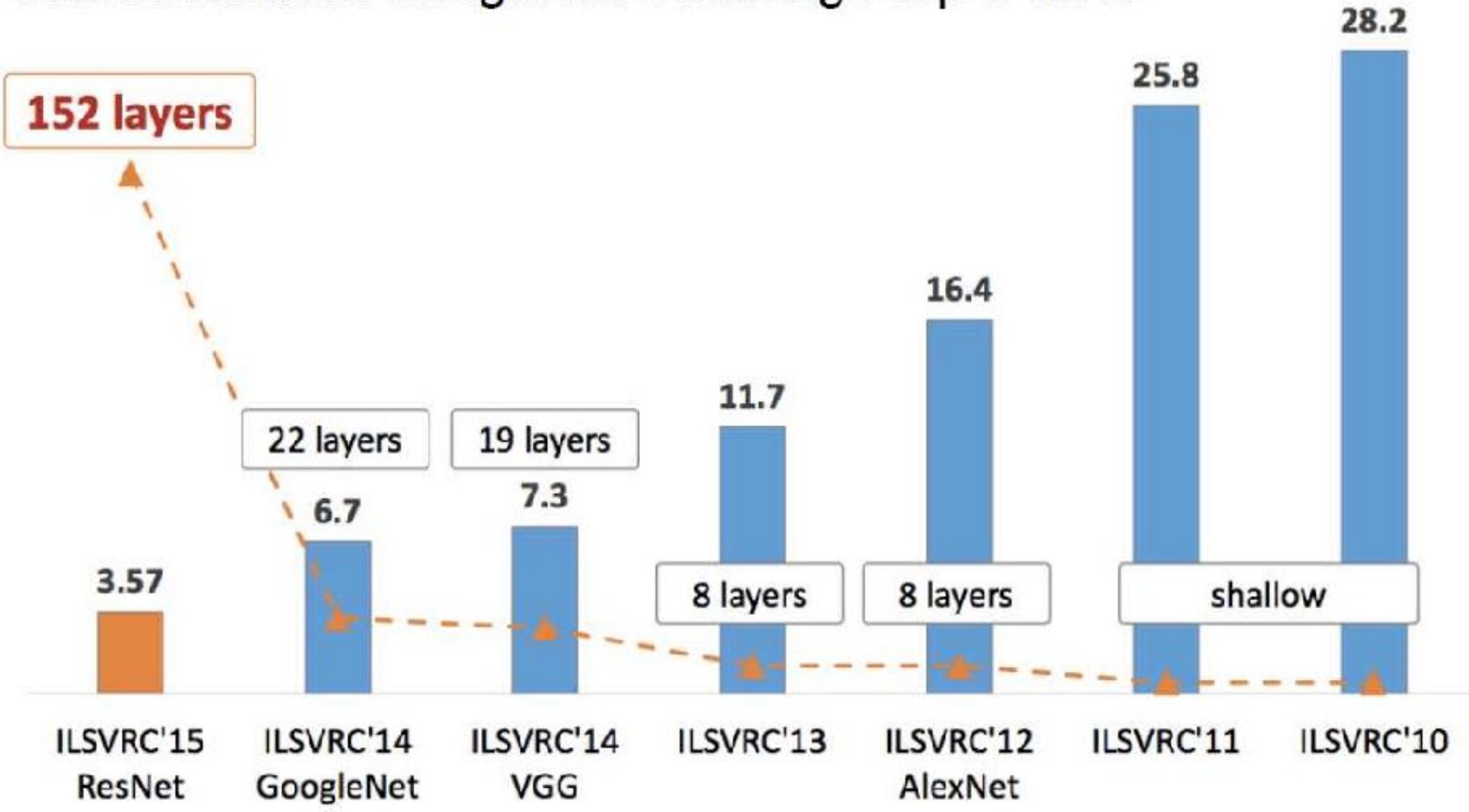
Layer 2: Top-9 Patches



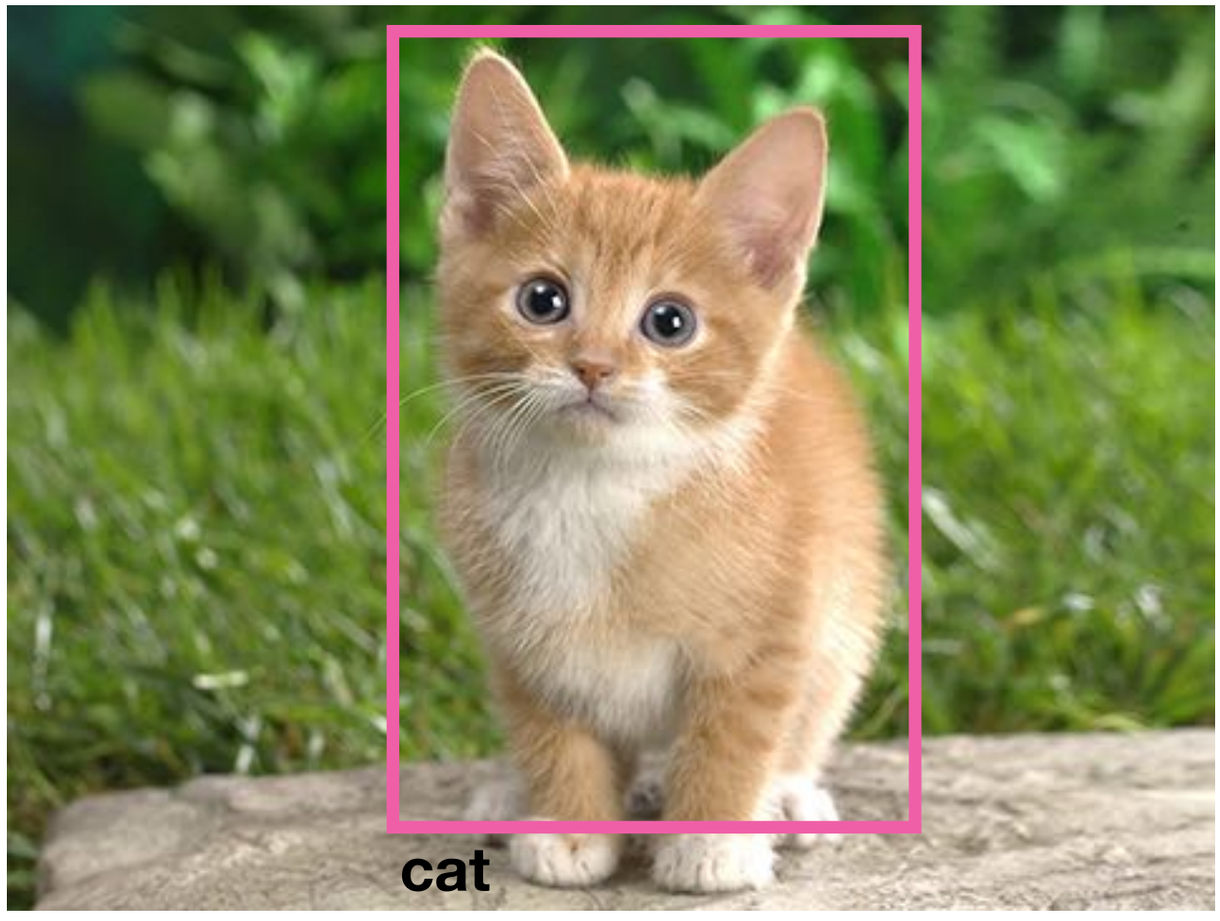
Layer 3: Top-9 Patches

Layer 4: Top-9 Patches

Classification: ImageNet Challenge top-5 error



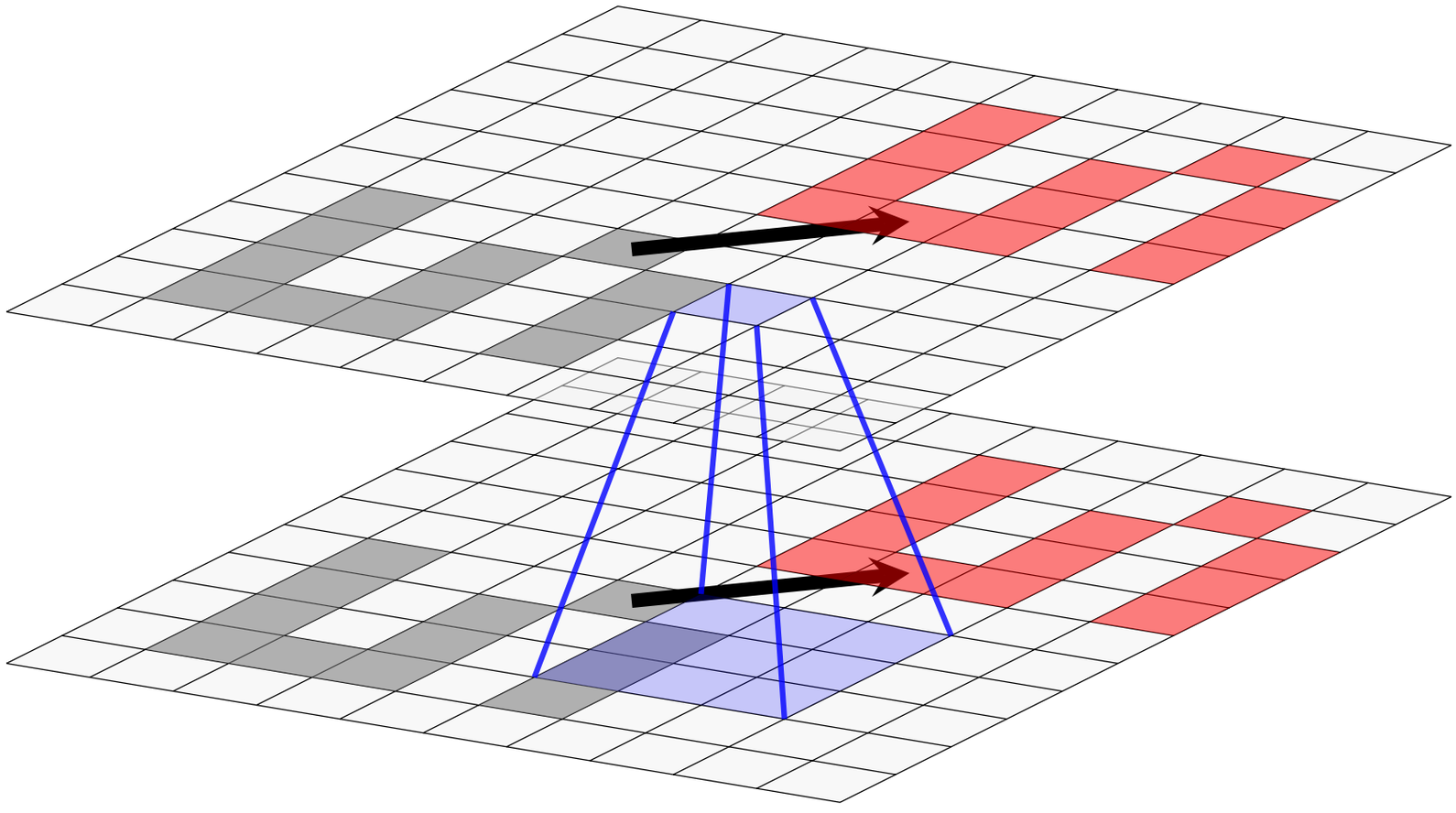
Current capability:



source: NVIDIA

Equivariance to translations

f_ℓ

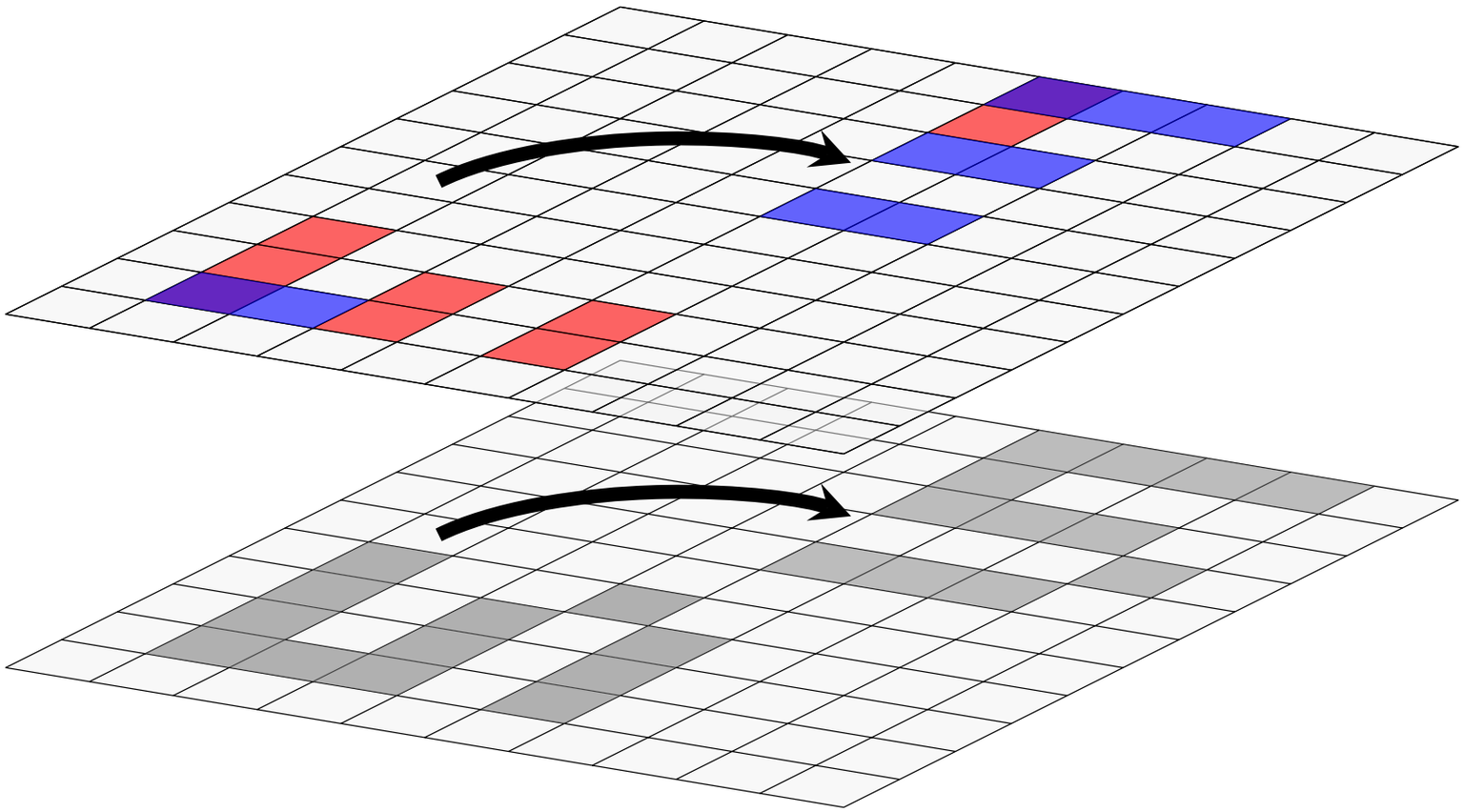


$f_{\ell-1}$

$$f'_\ell(T(\mathbf{x})) = f_\ell(\mathbf{x})$$

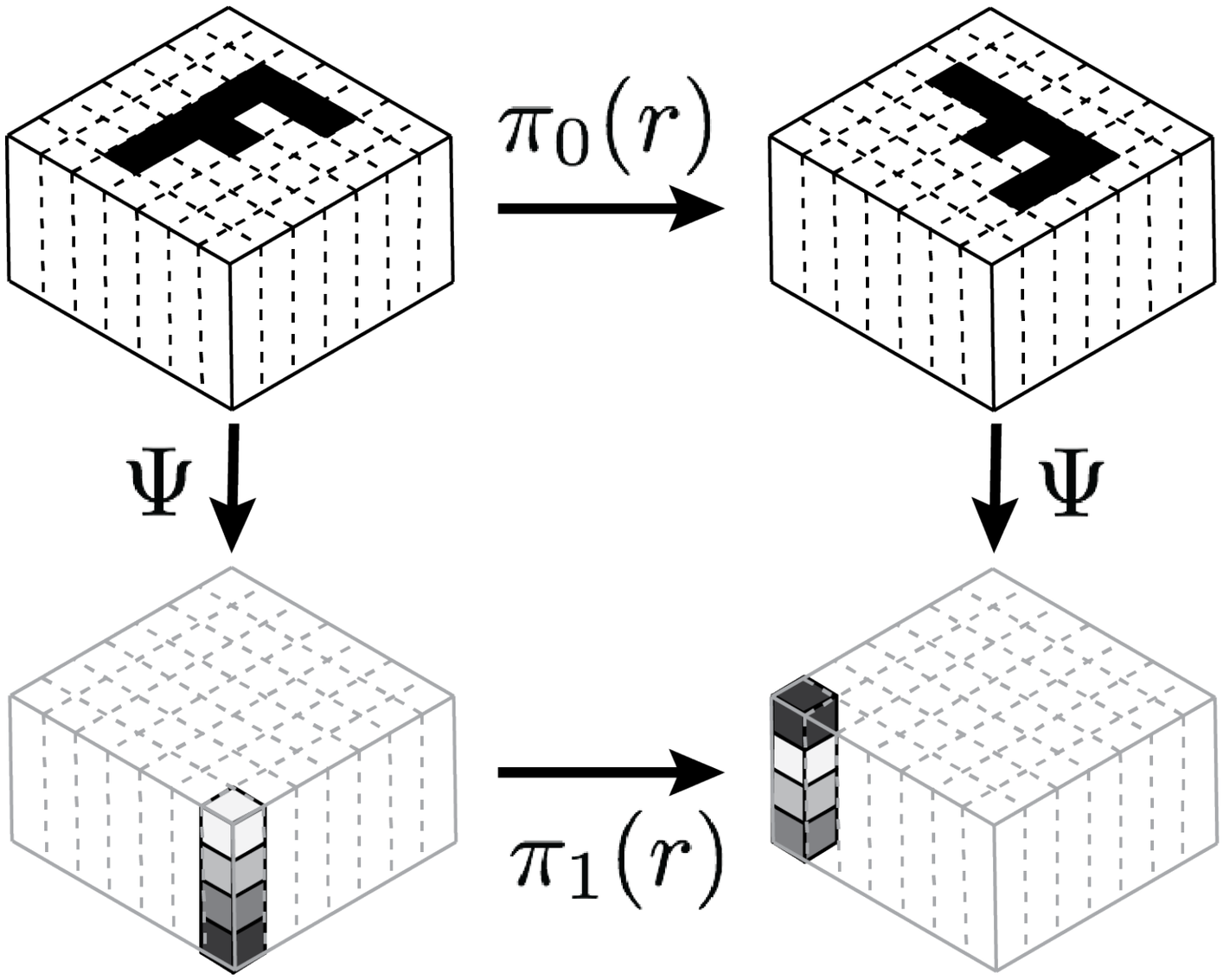
Steerability

f_ℓ



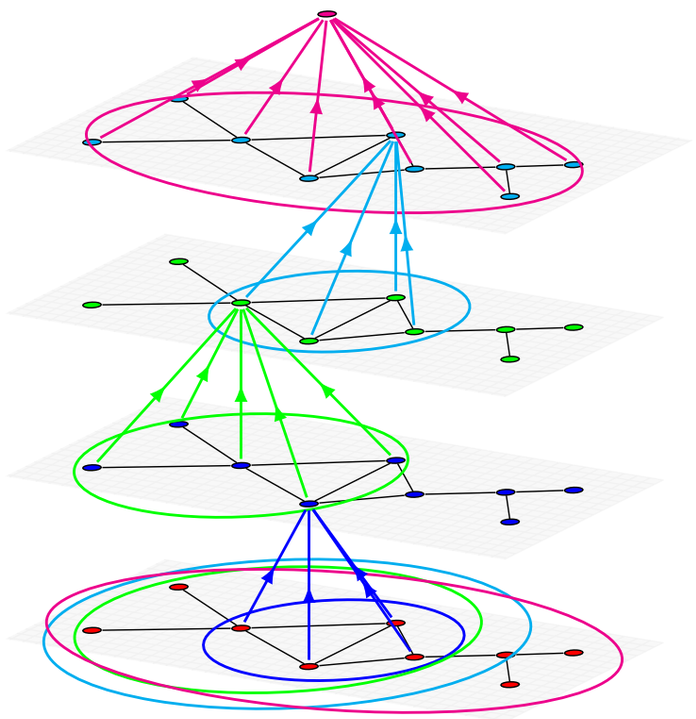
$f_{\ell-1}$

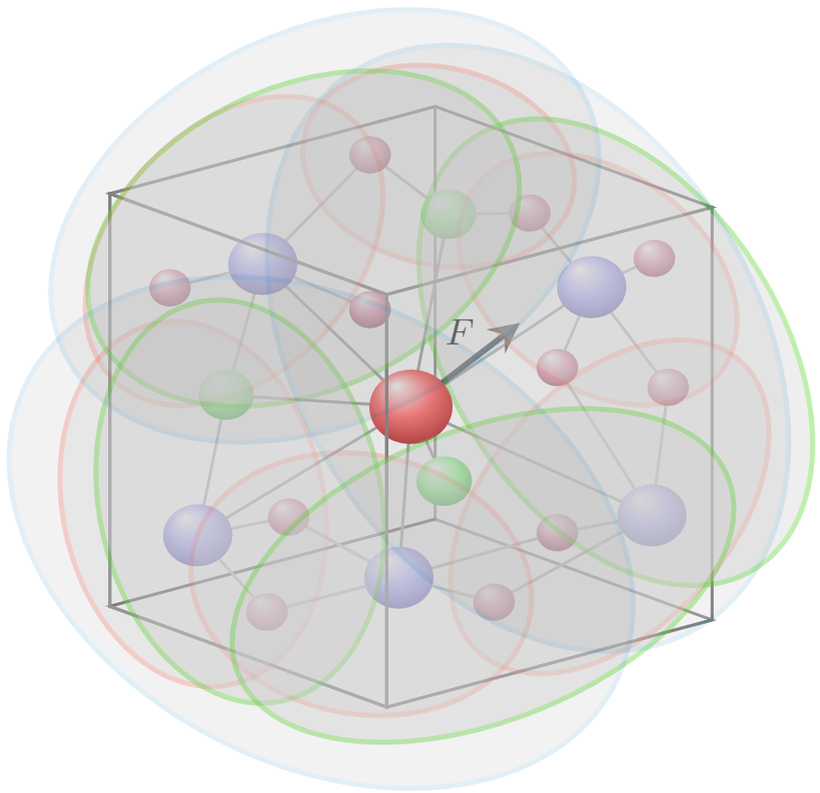
$$f'_\ell(T(\mathbf{x})) = R_T(f_\ell(\mathbf{x}))$$

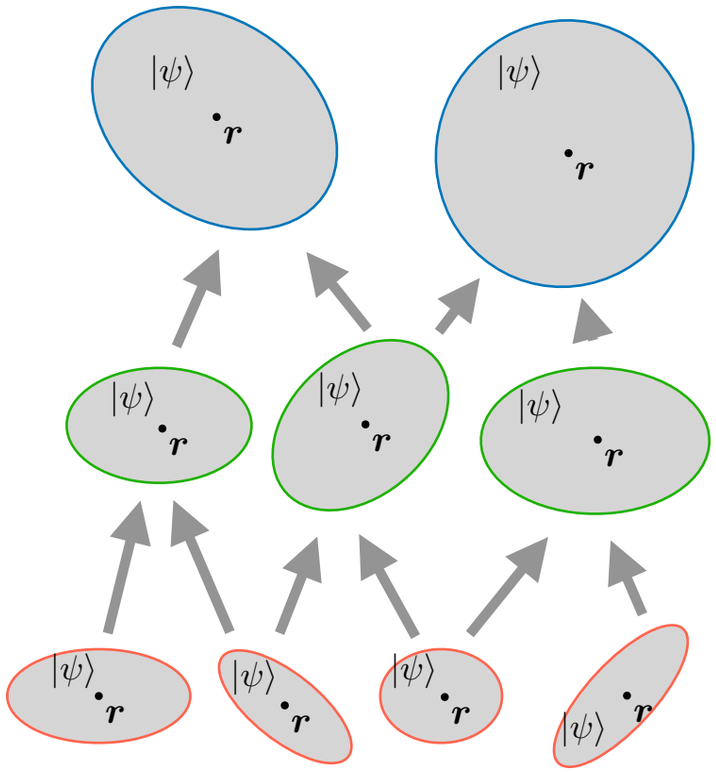


[Cohen & Welling, 2016]

What is the analog of convolutions on graphs?







1. Multiscale structure

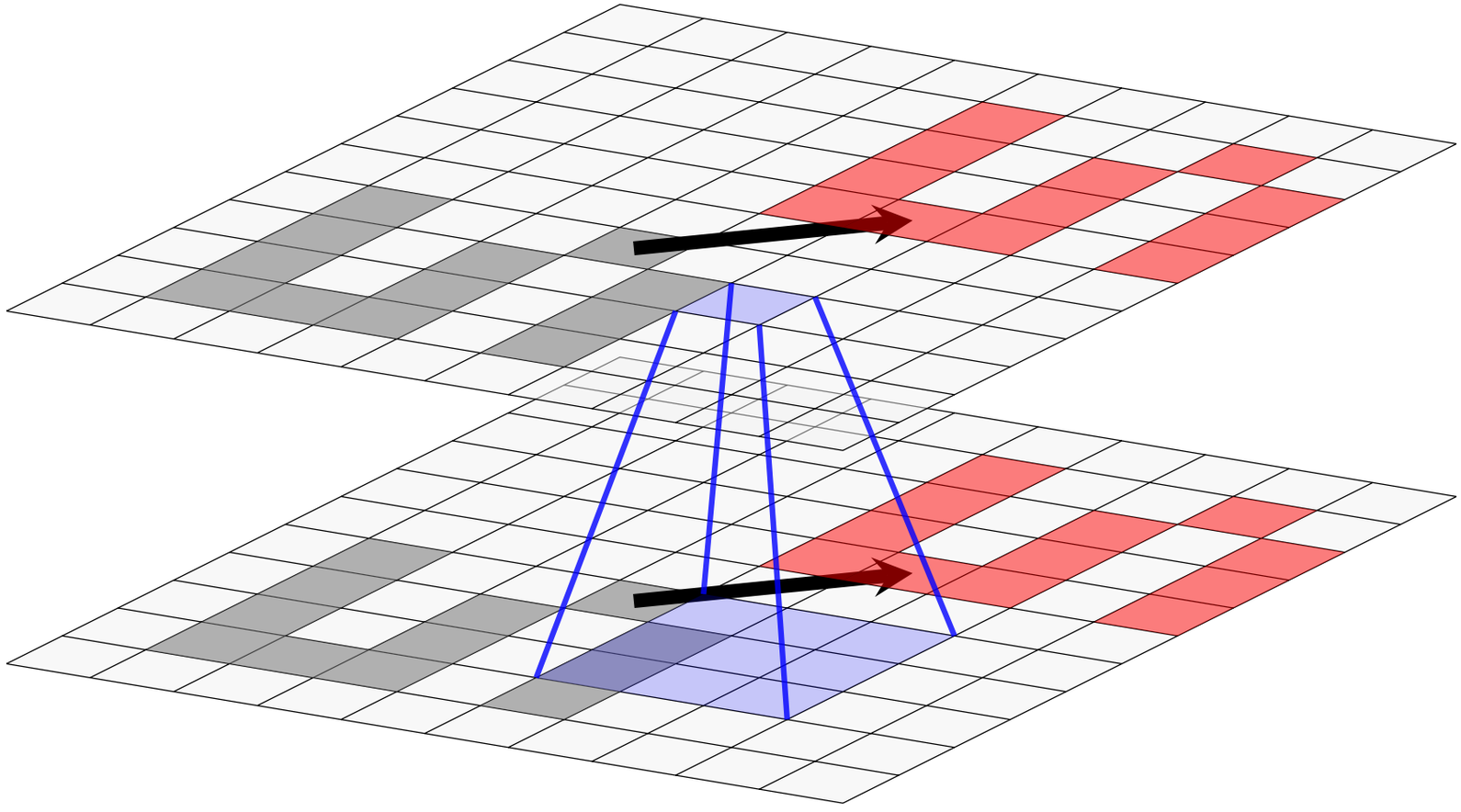
2. Covariance to the action of the symmetry group (permutations or rotations)

1. Theory

K. & Trivedi, ICML 2018

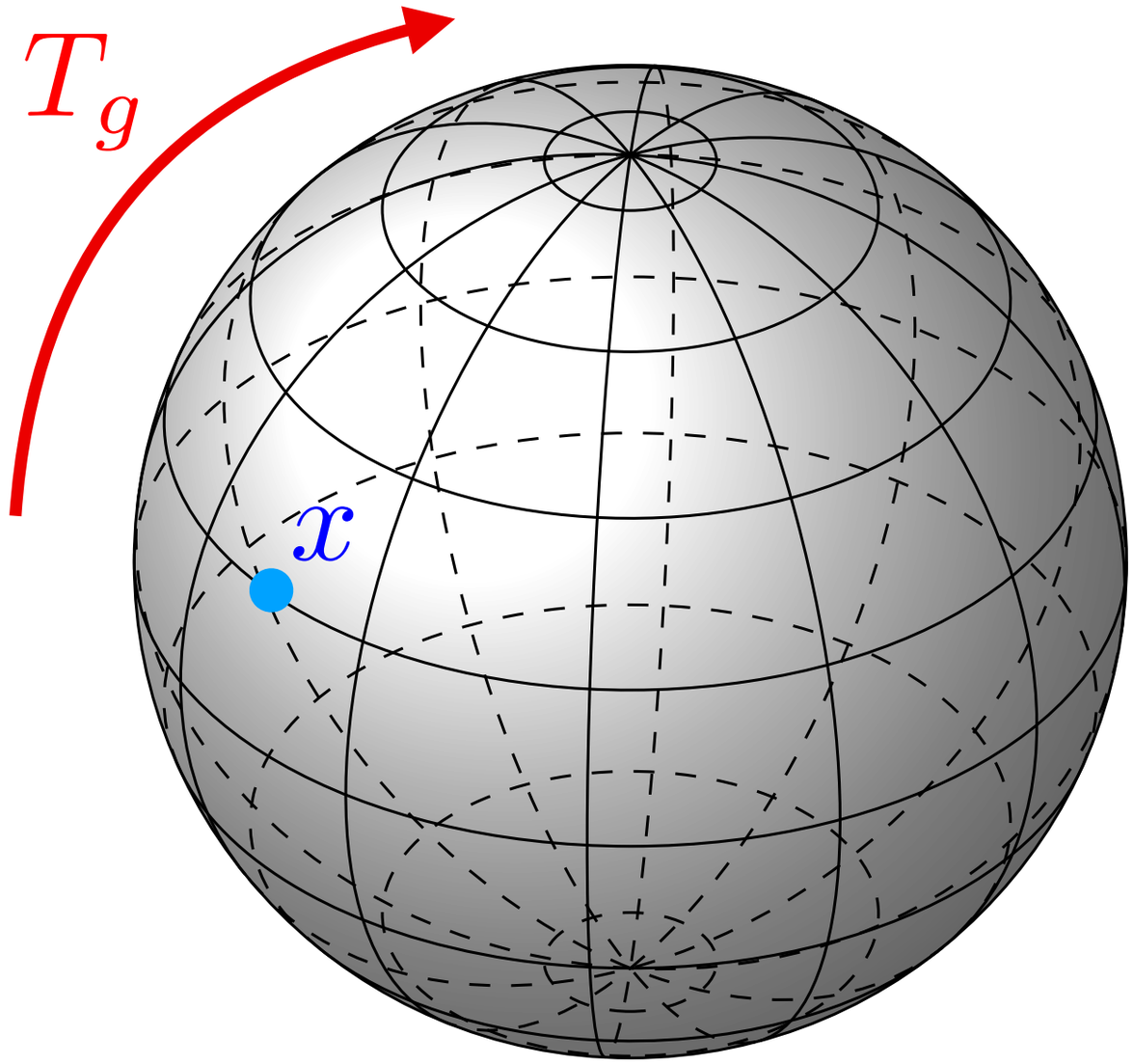
How do we generalize convolution to the action of general (compact) groups?

f_ℓ



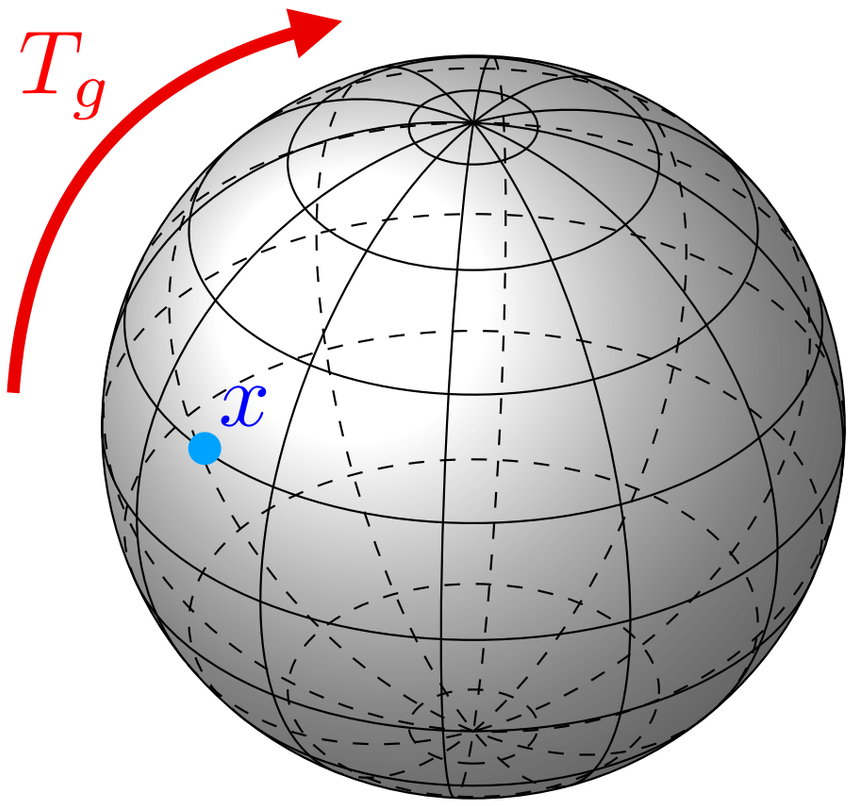
$f_{\ell-1}$

$$f'_\ell(T(\mathbf{x})) = f_\ell(\mathbf{x})$$



[Cohen, Geiger, Köhler & Welling, 2018]

Group actions



1. Our function lives on a space $\mathcal{X}$

$$f : \mathcal{X} \to \mathbb{C}$$

2. We have a group G acting on $\mathcal{X}$

$$x \mapsto T_g(x)$$

3. This induces an action on functions

$$f \overset{\mathbb{T}_g}{\longmapsto} f' \qquad f'(x) = f(T_g^{-1}(x))$$

Equivariance

Equivariance

1. We have to different spaces $\mathcal{X}_1$ and $\mathcal{X}_2$ on which G acts by

$$T_g^{(1)} : \mathcal{X}_1 \to \mathcal{X}_1 \qquad\qquad T_g^{(2)} : \mathcal{X}_2 \to \mathcal{X}_2$$

2. We have corresponding actions on functions

$$f \mapsto \mathbb{T}_g^{(1)}(f) \qquad\qquad \mathbb{T}_g^{(1)}(f)(x) = f((T_g^{(1)})^{-1}(x))$$
$$f \mapsto \mathbb{T}_g^{(2)}(f) \qquad\qquad \mathbb{T}_g^{(2)}(f)(x) = f((T_g^{(2)})^{-1}(x))$$

3. A map $\phi : L(\mathcal{X}_1) \to L(\mathcal{X}_2)$ is equivariant to these actions if

$$\phi(\mathbb{T}_g^{(1)}(f)) = \mathbb{T}_g^{(2)}(\phi(f))$$

for all $f \in L(\mathcal{X}_1)$.

$$
\begin{array}{ccc}
L(\mathcal{X}_1) & \xrightarrow{\mathbb{T}_g^{(1)}} & L(\mathcal{X}_1) \\
\downarrow{\scriptstyle \phi} & & \downarrow{\scriptstyle \phi} \\
L(\mathcal{X}_2) & \xrightarrow{\mathbb{T}_g^{(2)}} & L(\mathcal{X}_2)
\end{array}
$$

$$(f * g)(u) = \int_G f(uv^{-1})\, g(v)\, d\mu(v)$$

$$(f * g)(u) = \int_G f{\uparrow^G}(uv^{-1})\, g{\uparrow^G}(v)\, d\mu(v)$$

$$\widehat{f}(k) = \int f(x)\, e^{-ikx}\, dx$$

$$\widehat{(f * \chi)}(k) = \widehat{f}(k) \cdot \widehat{\chi}(k)$$

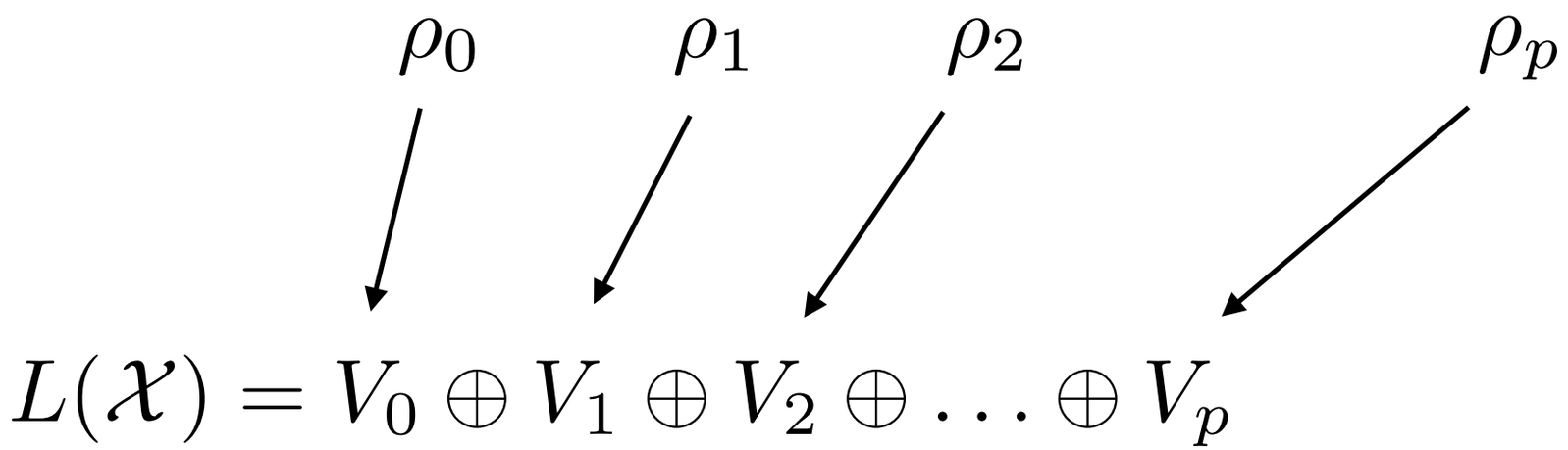
$$\widehat{f}(\rho) = \int f(x)\, \rho(x)\, d\mu(x)$$

$$\widehat{(f * \chi)}(\rho) = \widehat{f}(\rho) \cdot \widehat{\chi}(\rho)$$

Main theorem

A feed-forward neural network is equivariant to the action of a compact group G if and only if the linear operation in each layer is of the form

$$\phi_\ell(f_{\ell-1}) = f_{\ell-1} * g_\ell.$$



$$V_i = W_i^1 \oplus W_i^2 \oplus \ldots \oplus W_i^{m_i}$$

Consequences

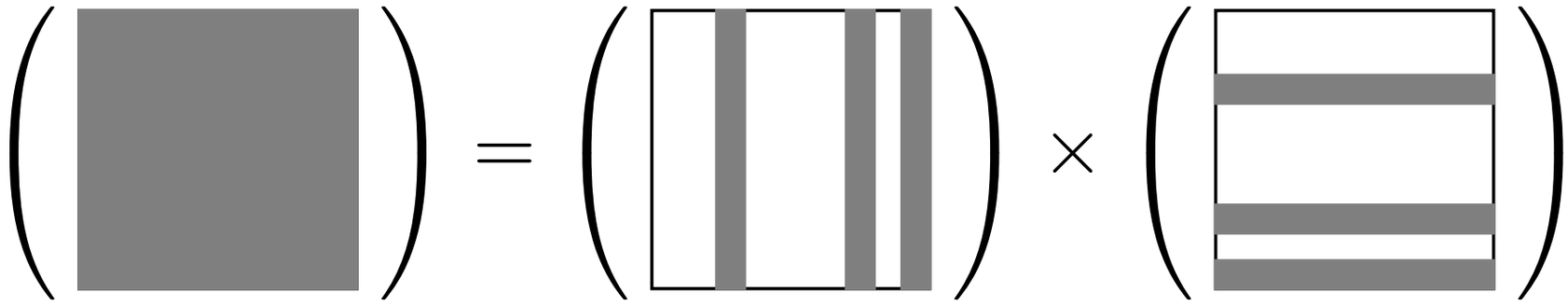
$$\widehat{f}(\rho_i) = \int_G f(u)\ \rho_i(u)\ d\mu(u) \qquad\qquad i = 0, 1, 2, \ldots$$

$$\widehat{f * g}(\rho_i) = \widehat{f}(\rho_i) \cdot \widehat{g}(\rho_i)$$

matrix multiplication

Case 1:

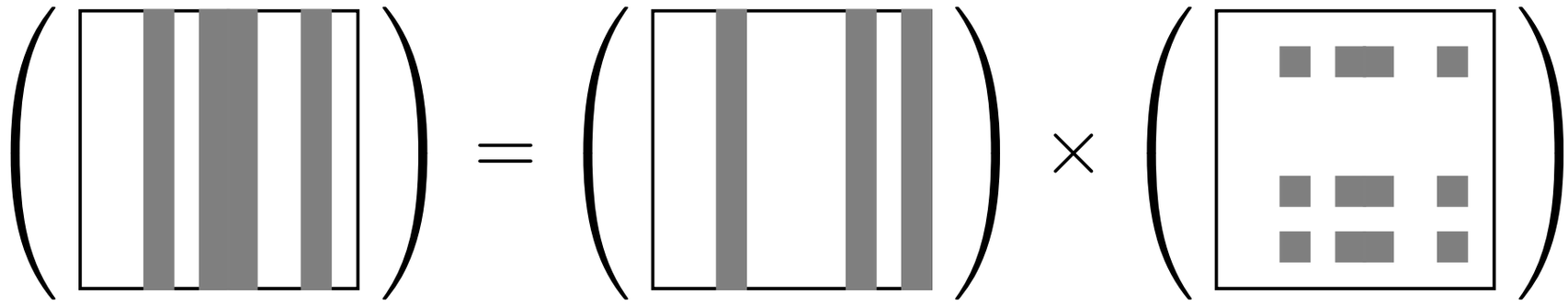
$$f_{\ell-1} \colon G/H \to \mathbb{C} \qquad f_\ell \colon G \to \mathbb{C}$$



$\widehat{f * g}(\rho)$ $\widehat{f\uparrow^G}(\rho)$ $\widehat{g\uparrow^G}(\rho)$

Case 2:

$$f_{\ell-1}\colon G/H \to \mathbb{C} \qquad f_{\ell}\colon G/K \to \mathbb{C}$$



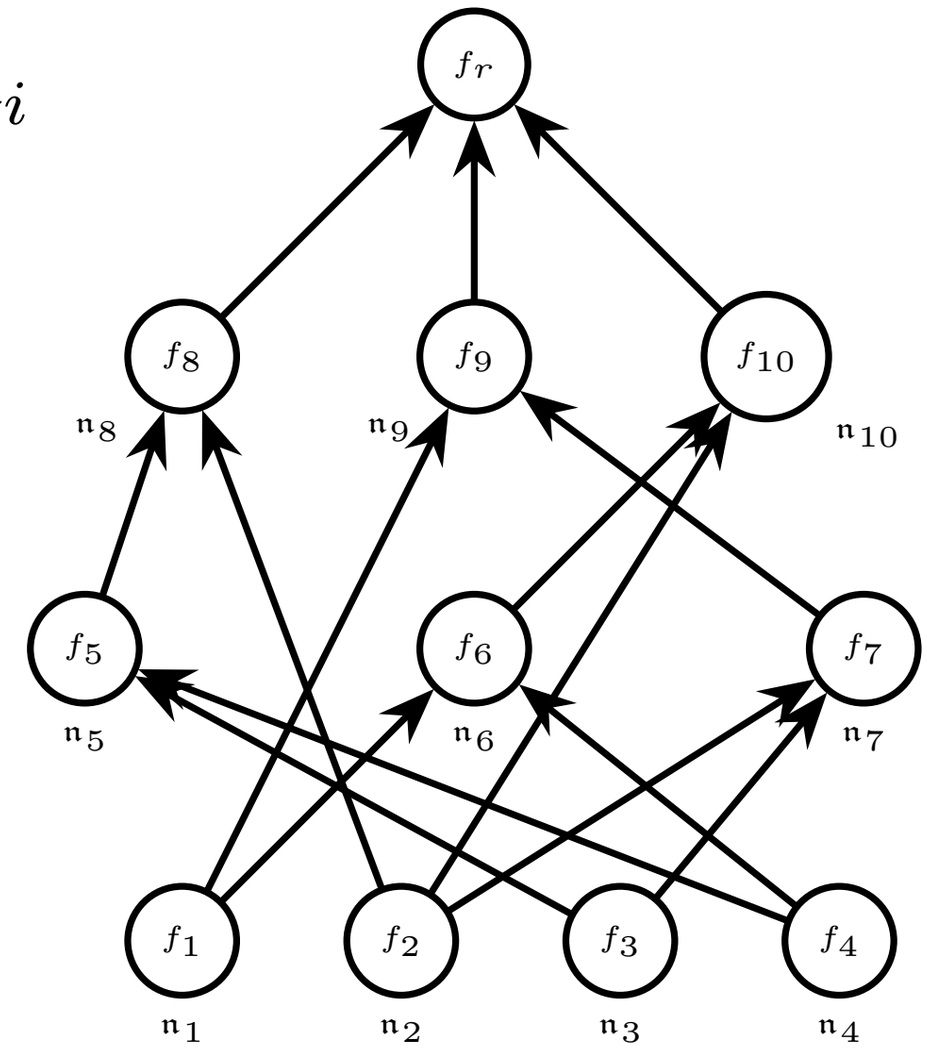
$$\widehat{f * g}(\rho) \qquad \widehat{f\uparrow^G}(\rho) \qquad \widehat{g\uparrow^G}(\rho)$$

Fourier space neural networks

Covariant neural networks

1. Each node in the graph is a neuron $\mathfrak{n}_i$ and its activation f_i is covariant to the action of G.

2. Each activation f_i is stored in Fourier space.



Recent example: Cohen et al.'s Spherical CNNs

Related: Hinton's capsules

1. Spherical CNNs

[Cohen, Geiger, Köhler & Welling, 2018]

[K., Lin and Trivedi, 2018]

2. Steerability and conv-nets for manifolds

[Marcos, Volpi et al., 2017]

[Masci, Boscaini et al., 2015]

[Worral, Garbin et al., 2017]

3. Neural nets for graphs

[Duvenaud at al., 2015]

[Gilmer et al., 2017]

[Son, Trivedi et al. 2018]

4. Neural nets for physical systems

[…]

Also see:

[Cohen, Geiger & Weiler, 2018]

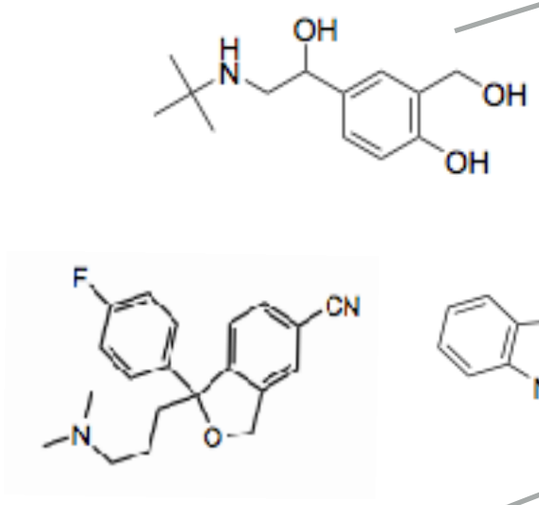
2. CCNs for graphs

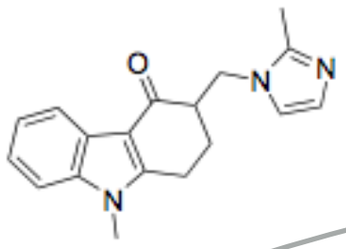
[Hy, Trivedi, Pan, Anderson and K, JCP special issue]

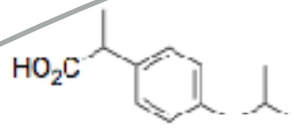
Kernel approach

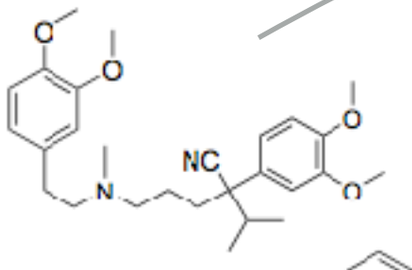
$k(G_1, G_2)$

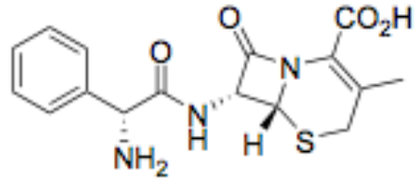
$\mathcal{H}$

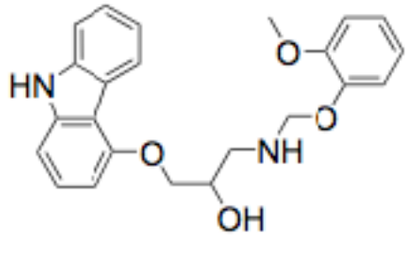






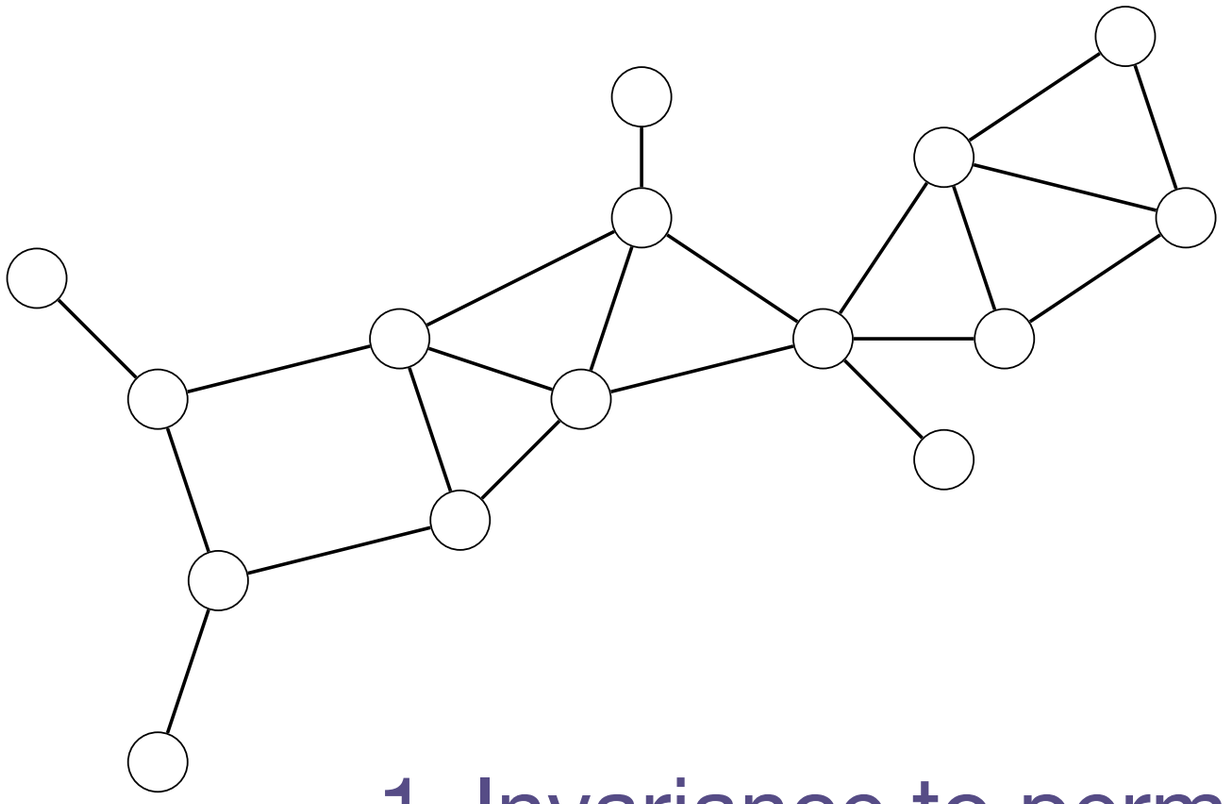






1. Random walks and spectral ideas
   [Gartner, 2002] [Vishwanathan et al., 2010]
2. Shortest Paths
   [Borgwardt & Kriegel, 2005]
3. Counting subgrapgs
   [Shervashidze et al., 2009] [Feragen et al., 2013]
3. Algebraic approach
   [K. & Borgwardt, 2008]
4. Label Propagation
   [Shervashidze et al., 2009]
5. Hierarchical
   [K.& Pan, 2016]

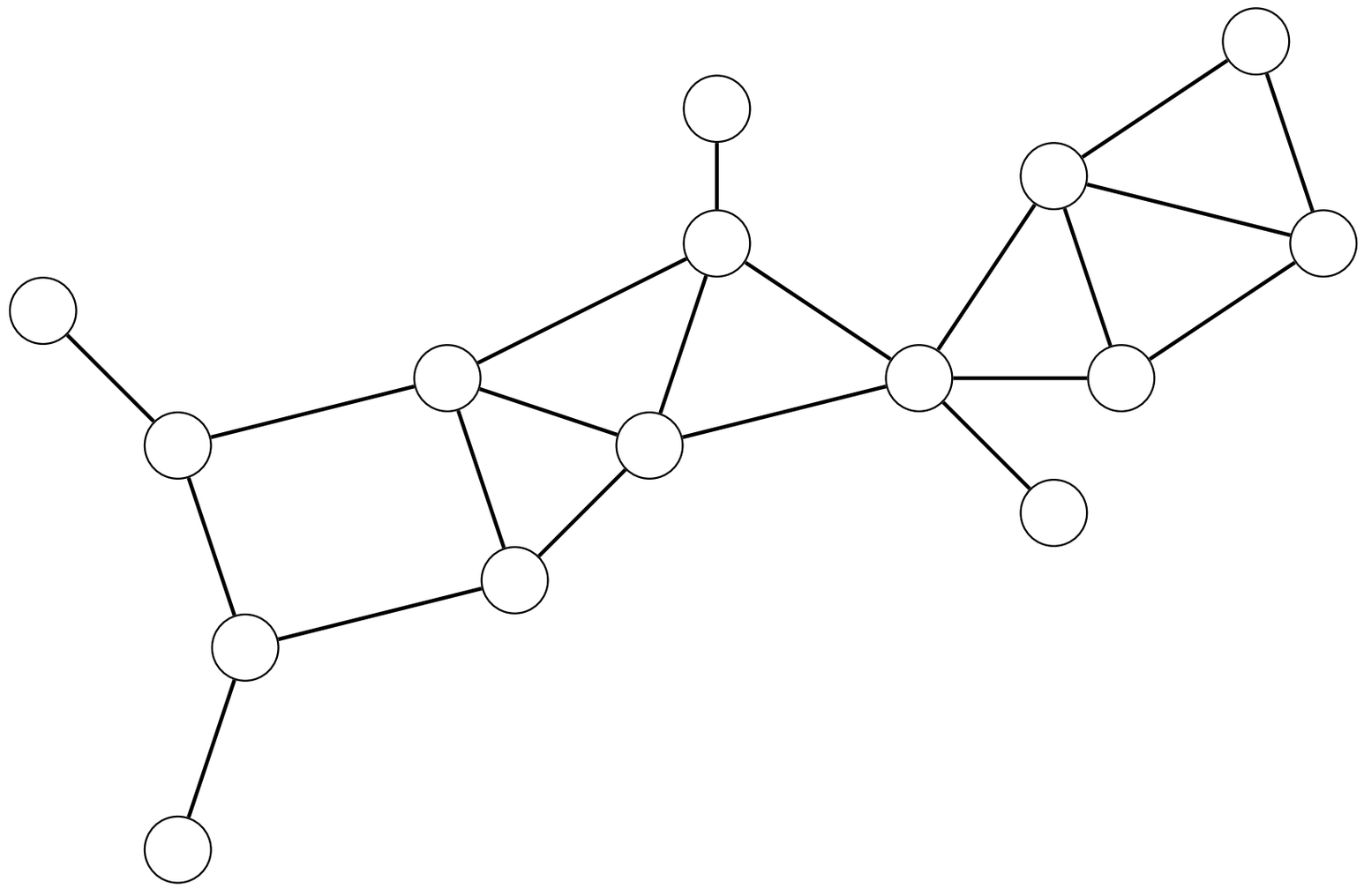
The kernel approach is an inherently fixed representation.

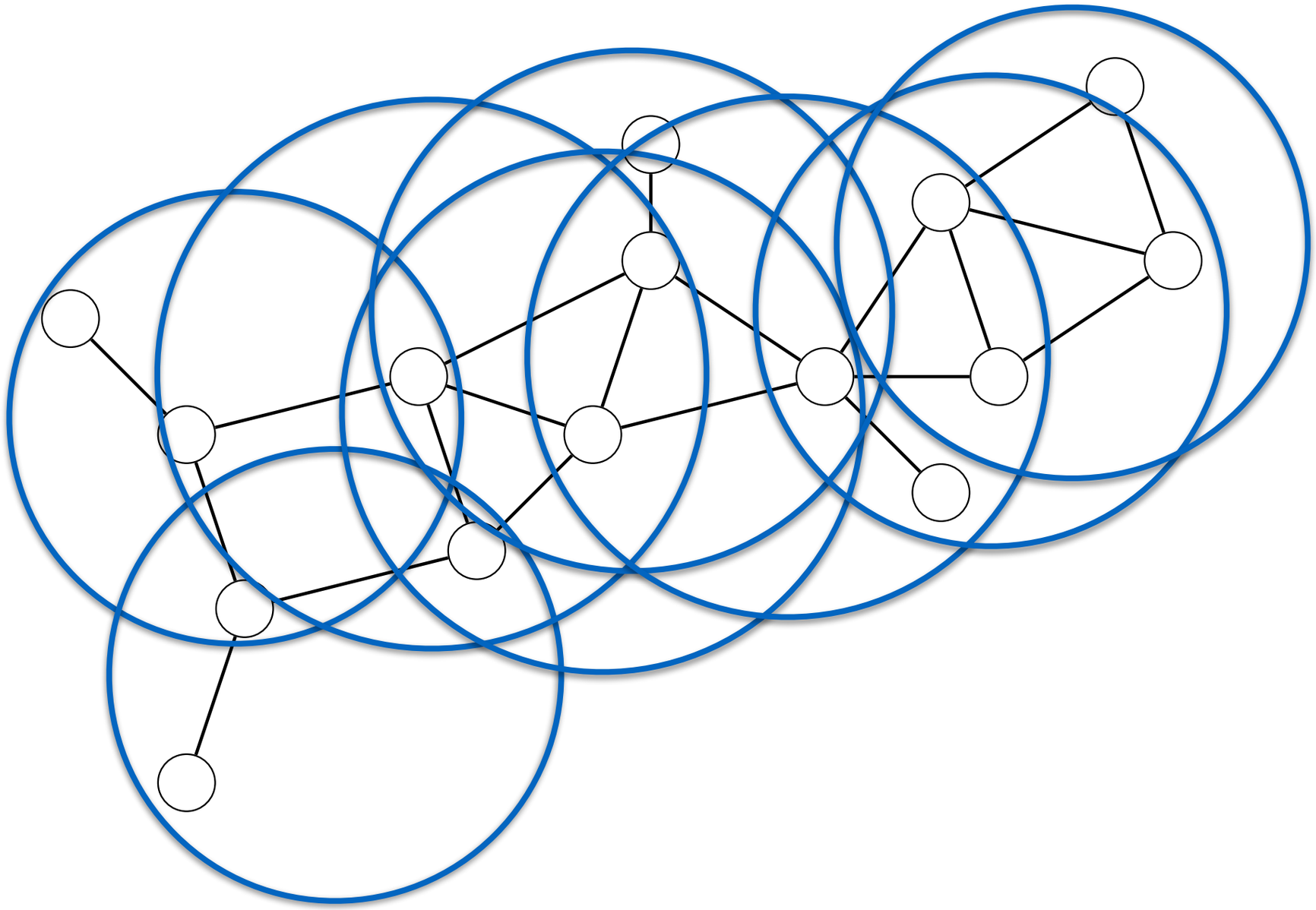


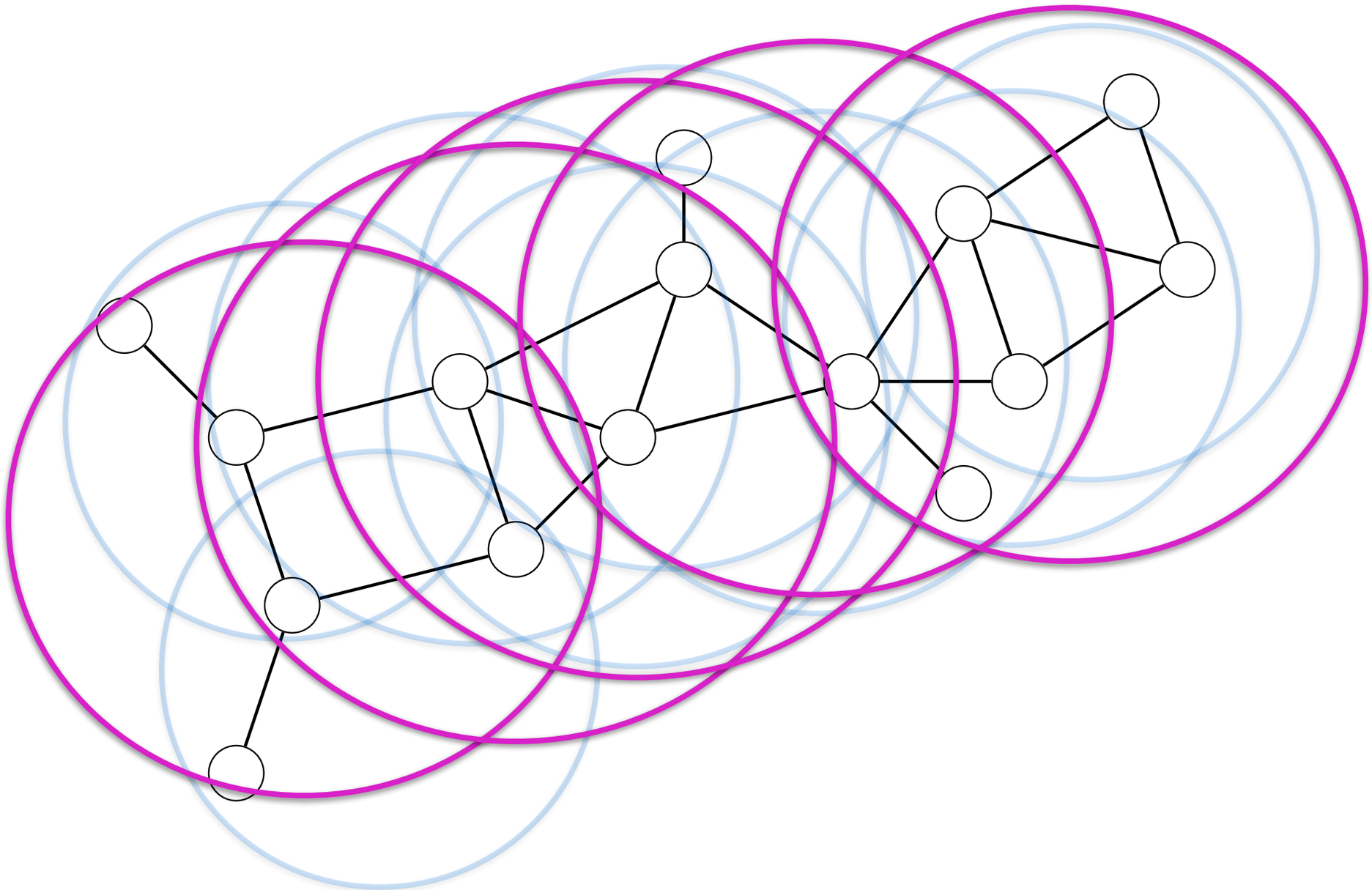
1. Invariance to permutations of vertices

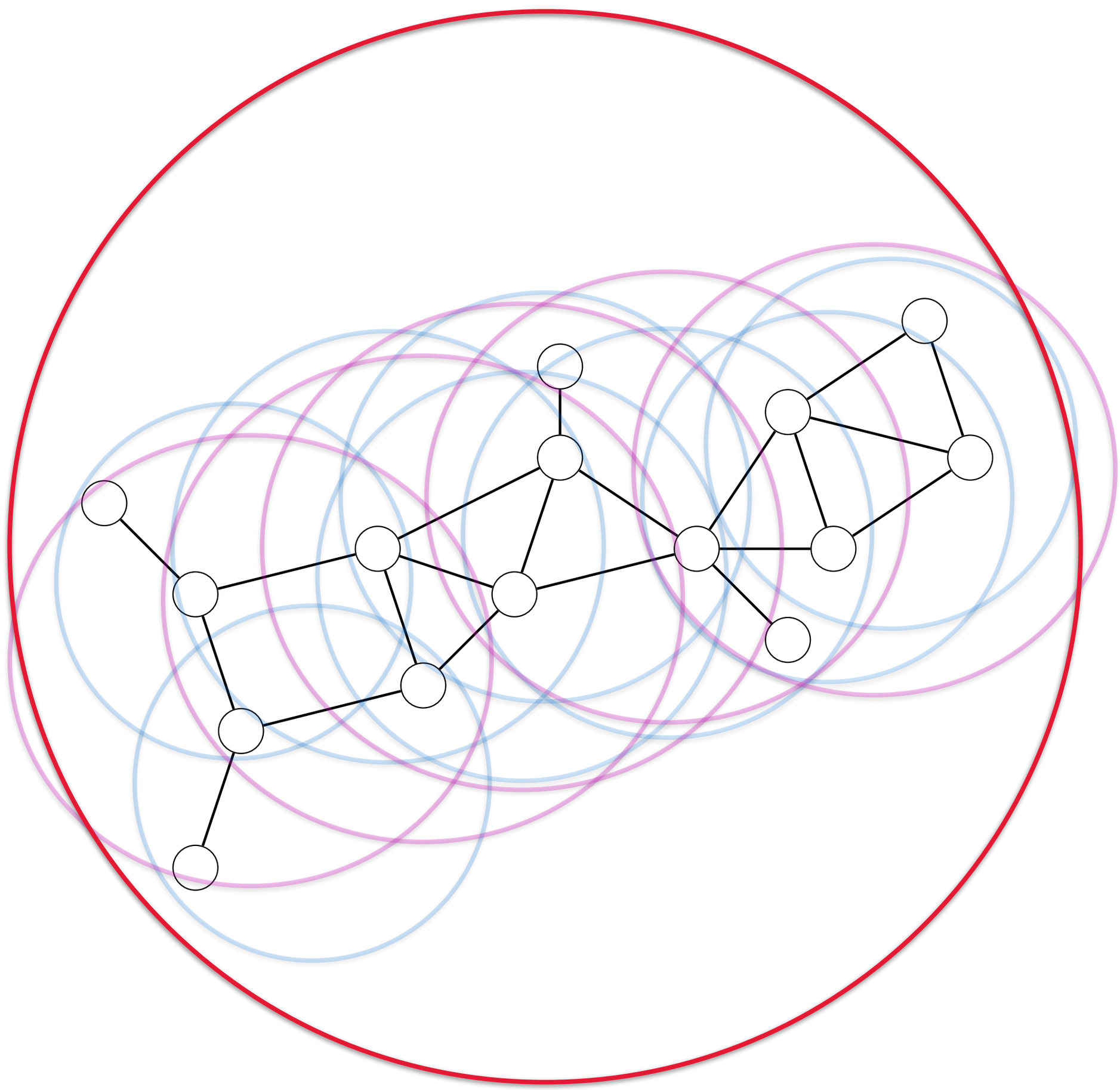
2. Ability to capture structure at multiple scales

Compositional approach

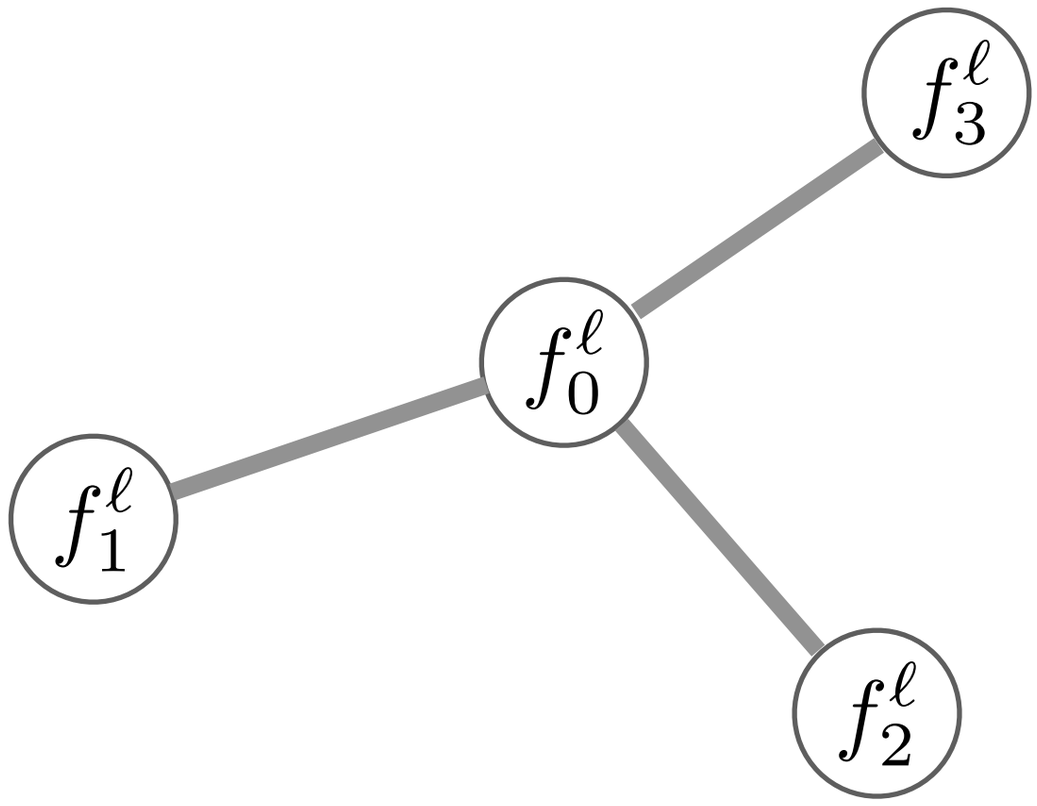




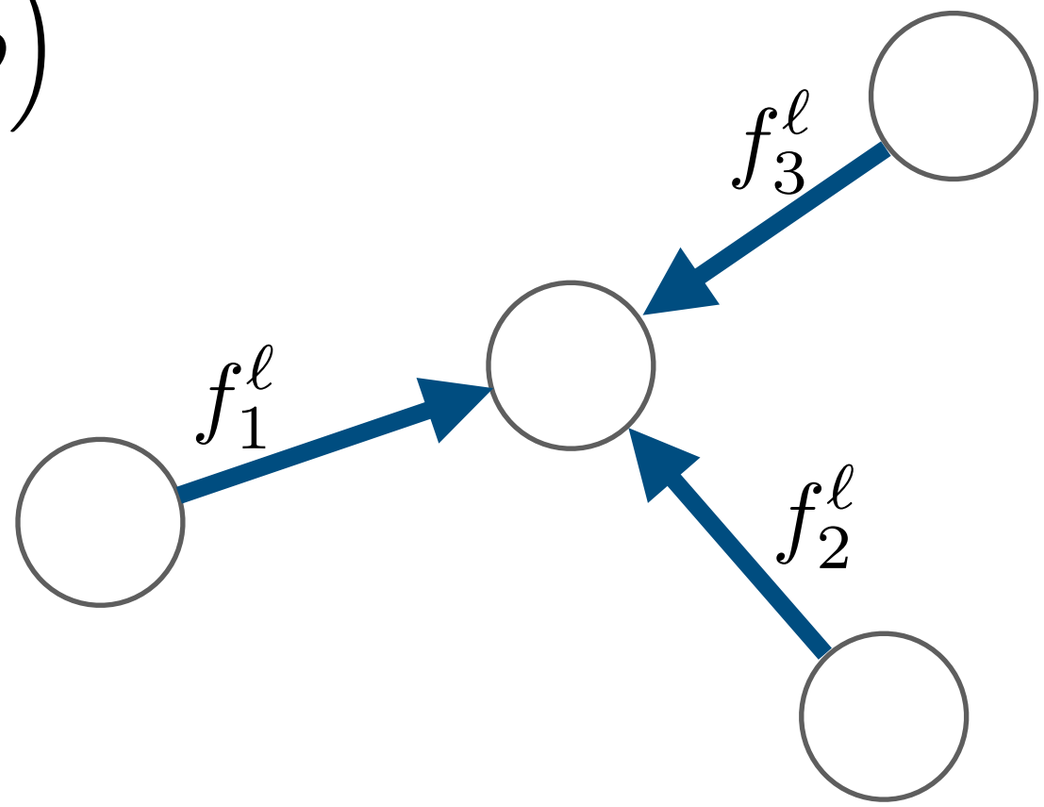




Label propagation schemes

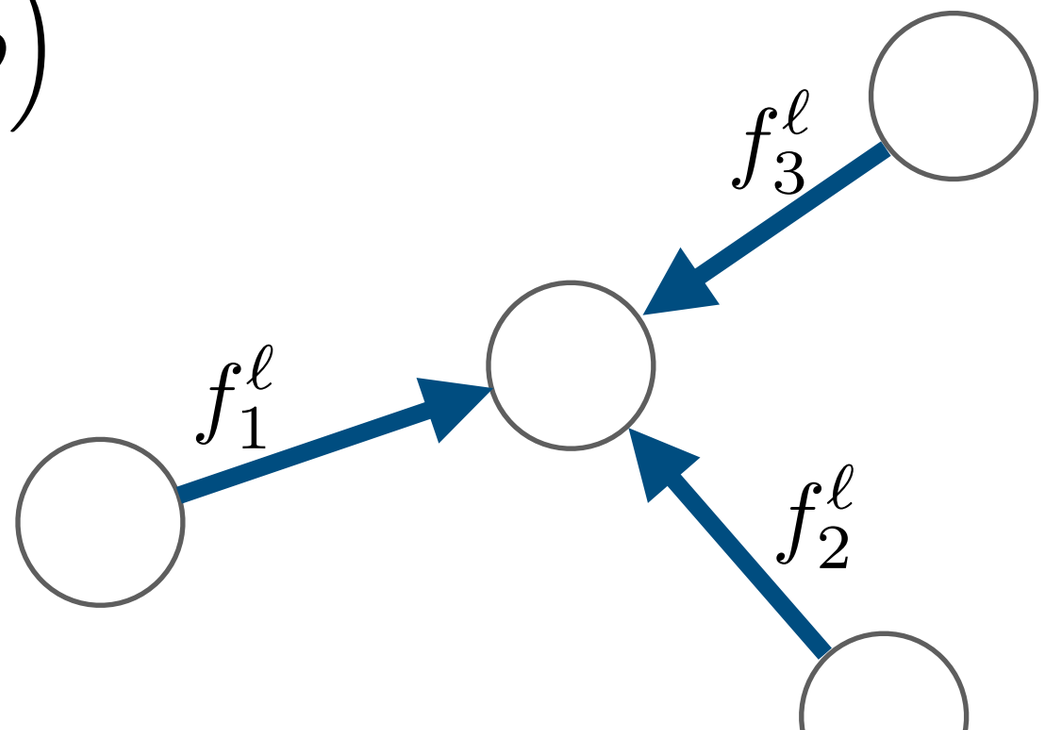


$$f_i^{\ell+1} = \xi\left(W \sum_{j \in \mathcal{N}(i)} f_j^{\ell} + b\right)$$

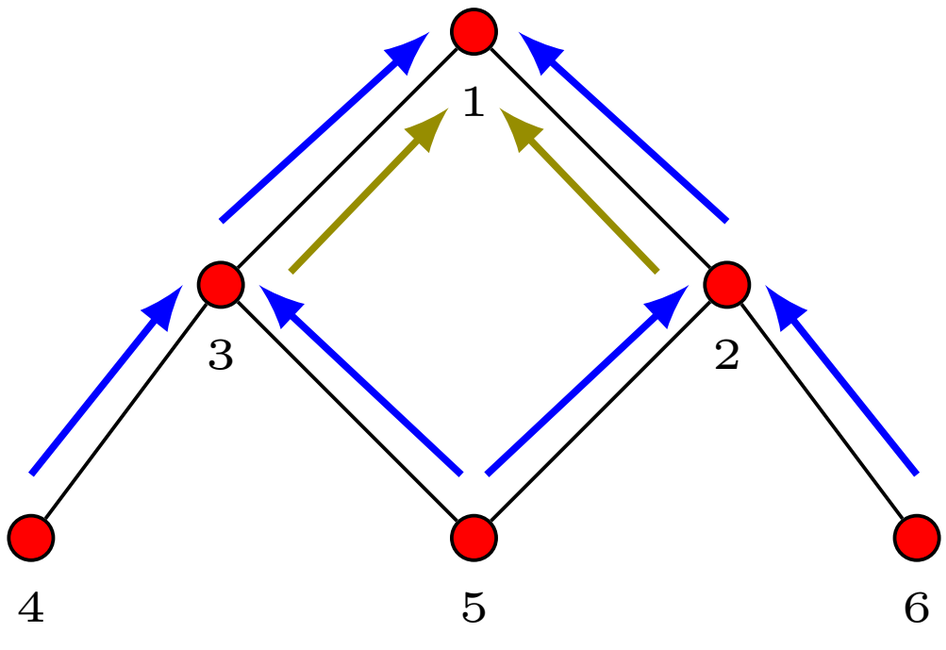


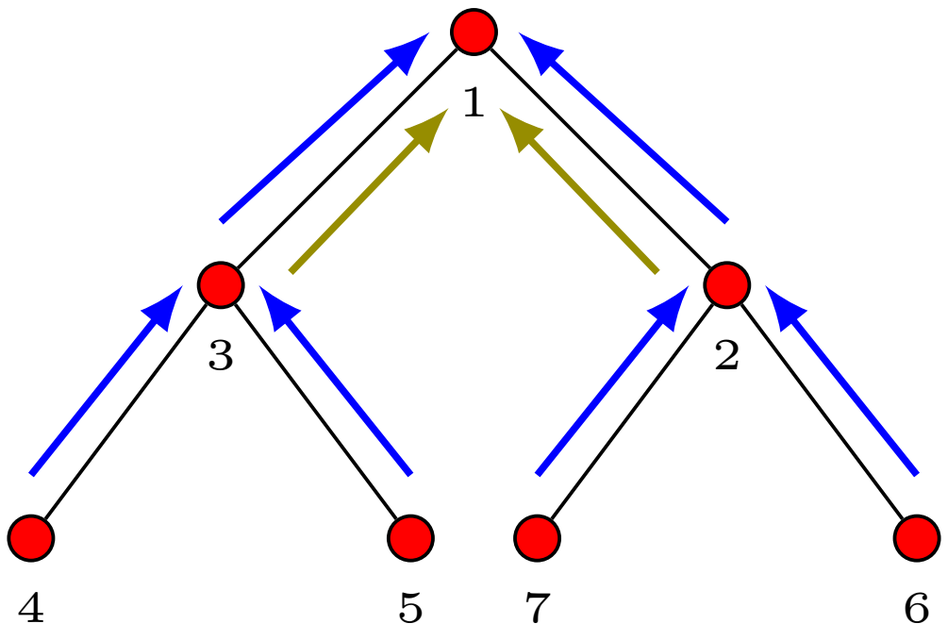
[Gilmer et al, ’17] [Kriege, ’16] [Niepert, ’16] [Duvenaud et al., ’15] [Dai, Dai & Song, ’16]

$$f_i^{\ell+1} = \xi\left(W \sum_{j \in \mathcal{N}(i)} f_j^{\ell} + b\right)$$



1. Satisfies permutation invariance
2. Aggregates information at multiple different scales
3. Does not fully account for topology

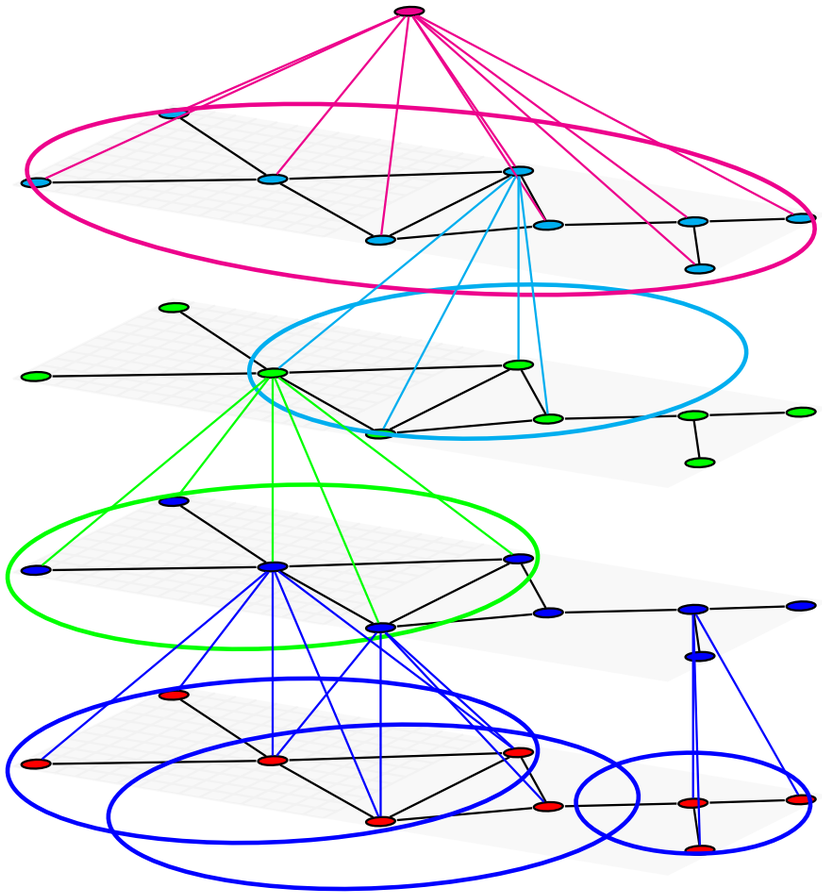

1
3
2
4
5
6

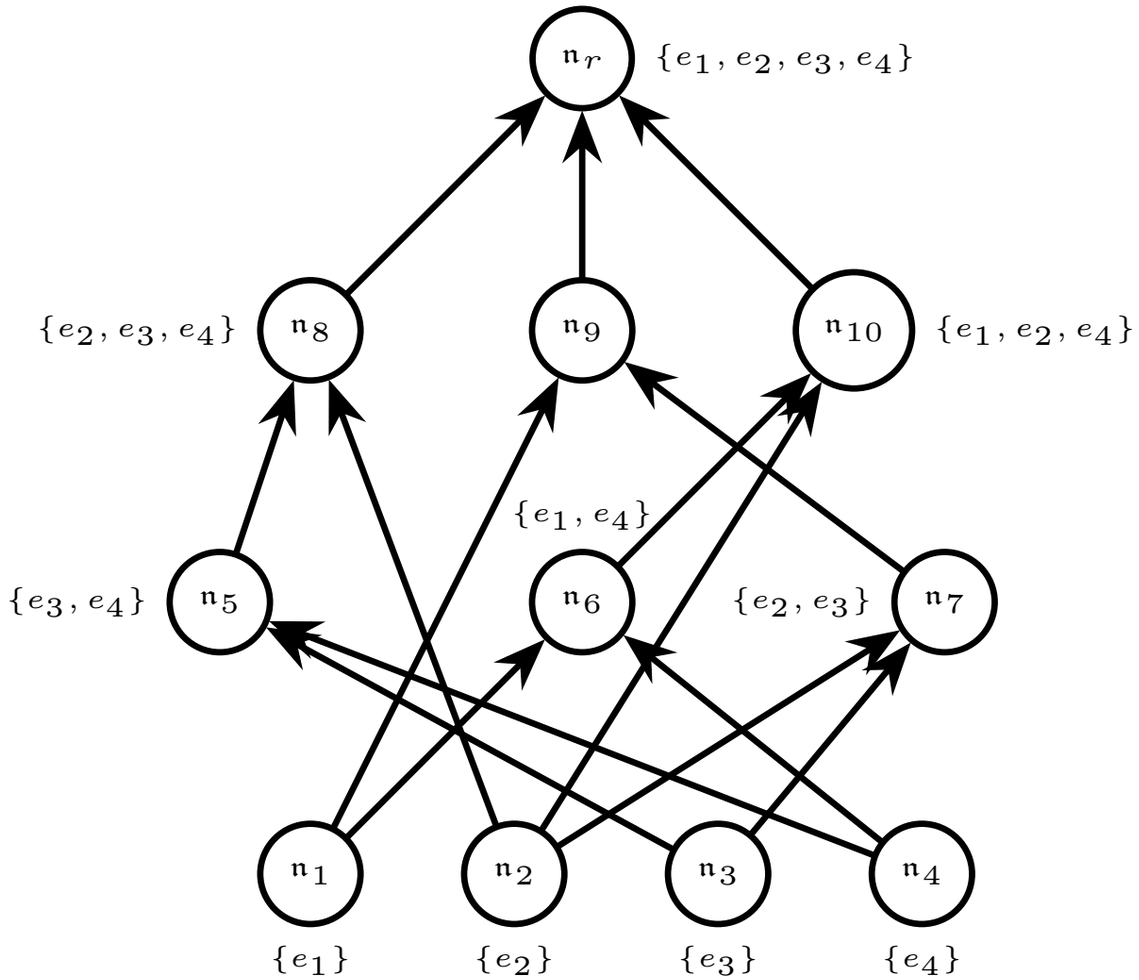

1
3
2
4
5
7
6

Compositional neural networks

[K., Pan, Hy-Truong, Trivedi & Anderson]

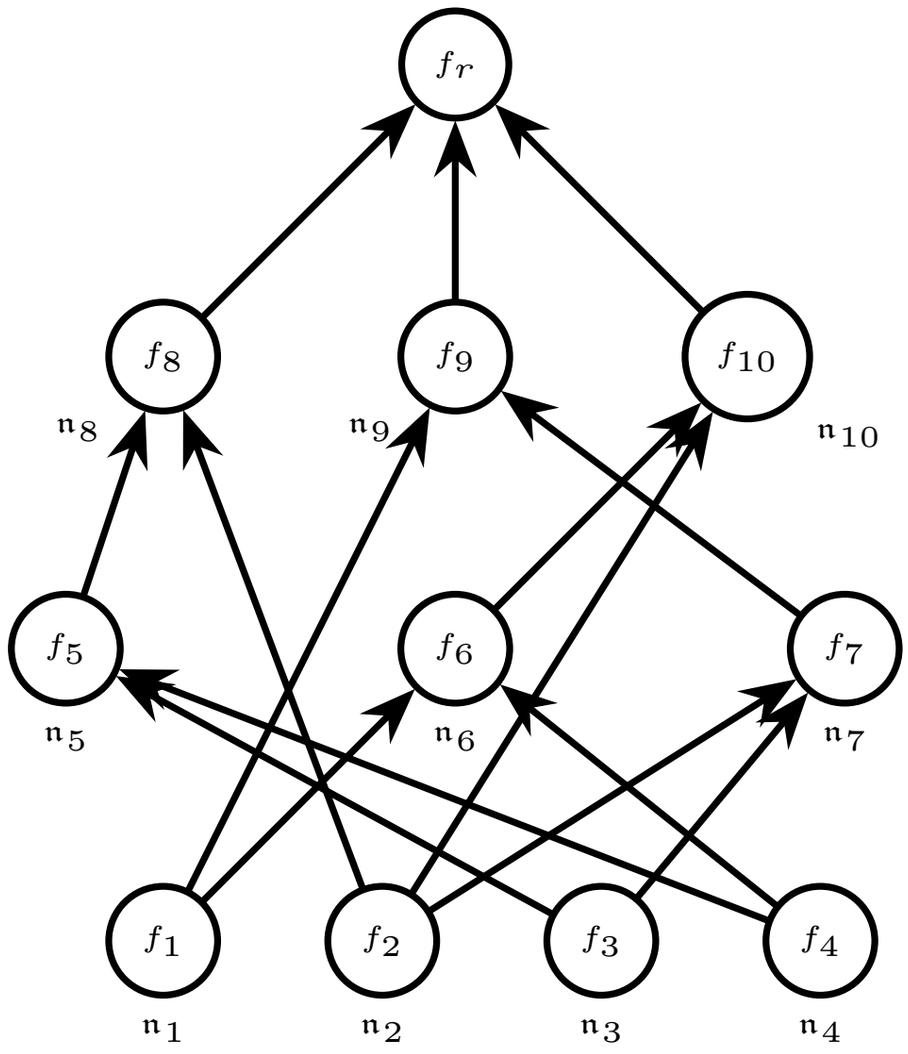
Composition scheme





Compositional networks (comp-nets)

$$f_i = \Phi(f_{c_1}, f_{c_2}, \ldots, f_{c_k})$$



Covariant Compositional network (CCN)

Quasi-invariant:

$$\Phi(f_{c_{\sigma(1)}}, f_{c_{\sigma(2)}}, \ldots, f_{c_{\sigma(k)}}) = \Phi(f_{c_1}, f_{c_2}, \ldots, f_{c_k})$$

Covariant:

$$\Phi(f_{c_{\sigma(1)}}, f_{c_{\sigma(2)}}, \ldots, f_{c_{\sigma(k)}}) = R_\sigma(\Phi(f_{c_1}, f_{c_2}, \ldots, f_{c_k}))$$

Here R_σ is a **representation** of $\mathbb{S}_k$.

0th order:

$$F_i \overset{\sigma}{\longmapsto} F_i$$

1st order:

$$F_i \overset{\sigma}{\longmapsto} P_\sigma F_i$$

2nd order:

$$F_i \overset{\sigma}{\longmapsto} P_\sigma F_i P_\sigma^\top$$

k'th order:

$$F_{i_1,i_2,\ldots,i_k} \overset{\sigma}{\longmapsto} [P_\sigma]_{i_1}^{j_1} [P_\sigma]_{i_2}^{j_2} \cdots [P_\sigma]_{i_k}^{j_k} F_{j_1,j_2,\ldots,j_k}$$

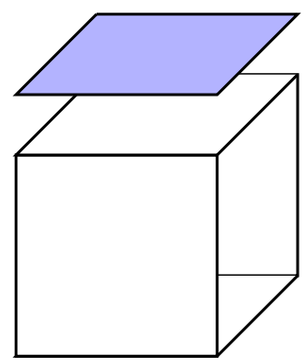
$$C = A \otimes B \qquad C_{i_1,i_2,\ldots,i_{k+p}} = A_{i_1,i_2,\ldots,i_k}\, B_{i_{k+1},i_{k+2},\ldots,i_{k+p}}$$

$$C = A \odot_{(a_1,\ldots,a_p)} B \qquad C_{i_1,i_2,\ldots,i_k} = A_{i_1,i_2,\ldots,i_k}\, B_{i_{a_1},i_{a_2},\ldots,i_{a_p}}$$

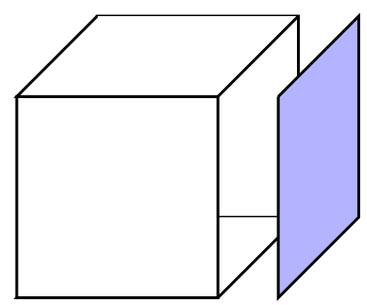
$$C = A{\downarrow_{a_1,\ldots,a_p}} \qquad C_{i_1,i_2,\ldots,i_k} = \sum_{i_{a_1}} \sum_{i_{a_2}} \cdots \sum_{i_{a_p}} A_{i_1,i_2,\ldots,i_k},$$

$$C_{i_1,i_2,\ldots,i_k} = A_{i_1,i_2,\ldots,i_k}\, \delta^{i_a,i_b}$$

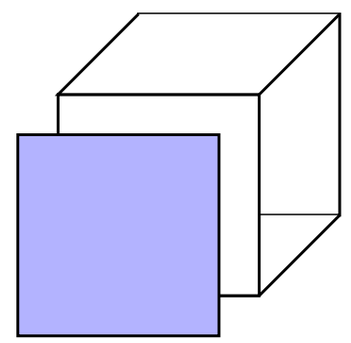
$$C_{i_1,i_2,\ldots,i_k} = \sum_j A_{i_1,\ldots,i_{a-1},j,i_{a+i},\ldots,i_{b-1},j,i_{b+1},\ldots,k}$$



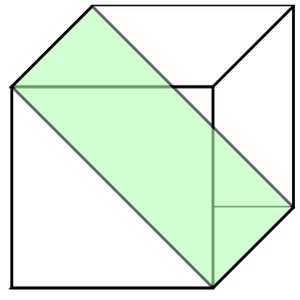
$$C_{i,j} = \sum_a A_{a,i,j}$$



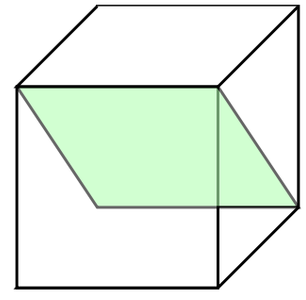
$$C_{i,j} = \sum_j A_{i,a,j}$$



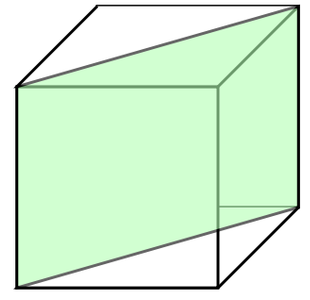
$$C_{i,j} = \sum_k A_{i,j,a}$$



$$C_{i,j} = \sum_i A_{i,i,j}$$



$$C_{i,j} = \sum_i A_{i,j,i}$$



$$C_{i,j} = \sum_i A_{i,j,j}$$

Figure 1: There are six different ways of covariantly reducing a third order tensor to a second order tensor: three different ways of projecting along each of its dimensions, and three different ways of taking the "trace" along a pair of dimensions.

Proposition. Assume that A and B are k'th and p'th order P–tensors, respectively. Then

1. $A \otimes B$ is a $k+p$'th order P–tensor.
2. $A \odot_{(a_1,\ldots,a_p)} B$ is a k'th order P–tensor.
3. $A\downarrow_{a_1,\ldots,a_p}$ is a $k-p$'th order P–tensor.
4. $A_{i_1,i_2,\ldots,i_k}\, \delta^{a_1^1,\ldots,a_{p_1}^1} \ldots \delta^{a_1^q,\ldots,a_{p_q}^q}$ is a $k-\sum_j p_j$'th order P–tensor.

In addition, if $A_1, \ldots, A_u$ are P–tensors and $\alpha_1, \ldots, \alpha_u$ are scalars, then $\sum_j \alpha_j A_j$ is a P–tensor.

Proposition Assume that node $\mathfrak{n}_a$ is a descendant of node $\mathfrak{n}_b$ in a comp-net $\mathcal{N}$, $\mathcal{P}_a = (e_{p_1}, \ldots, e_{p_m})$ and $\mathcal{P}_b = (e_{q_1}, \ldots, e_{q_{m'}})$ are the corresponding ordered receptive fields, and $\chi^{a\to b} \in \mathbb{R}^{m\times m'}$ is an indicator matrix defined

$$\chi_{i,j}^{a\to b} = \begin{cases} 1 & \text{if } \; q_i = p_j \\ 0 & \text{otherwise.} \end{cases}$$

Assume that F is a k'th order P–tensor with respect to permutations of $(e_{p_1}, \ldots, e_{p_m})$. Then, dropping the $^{a\to b}$ superscript for clarity,

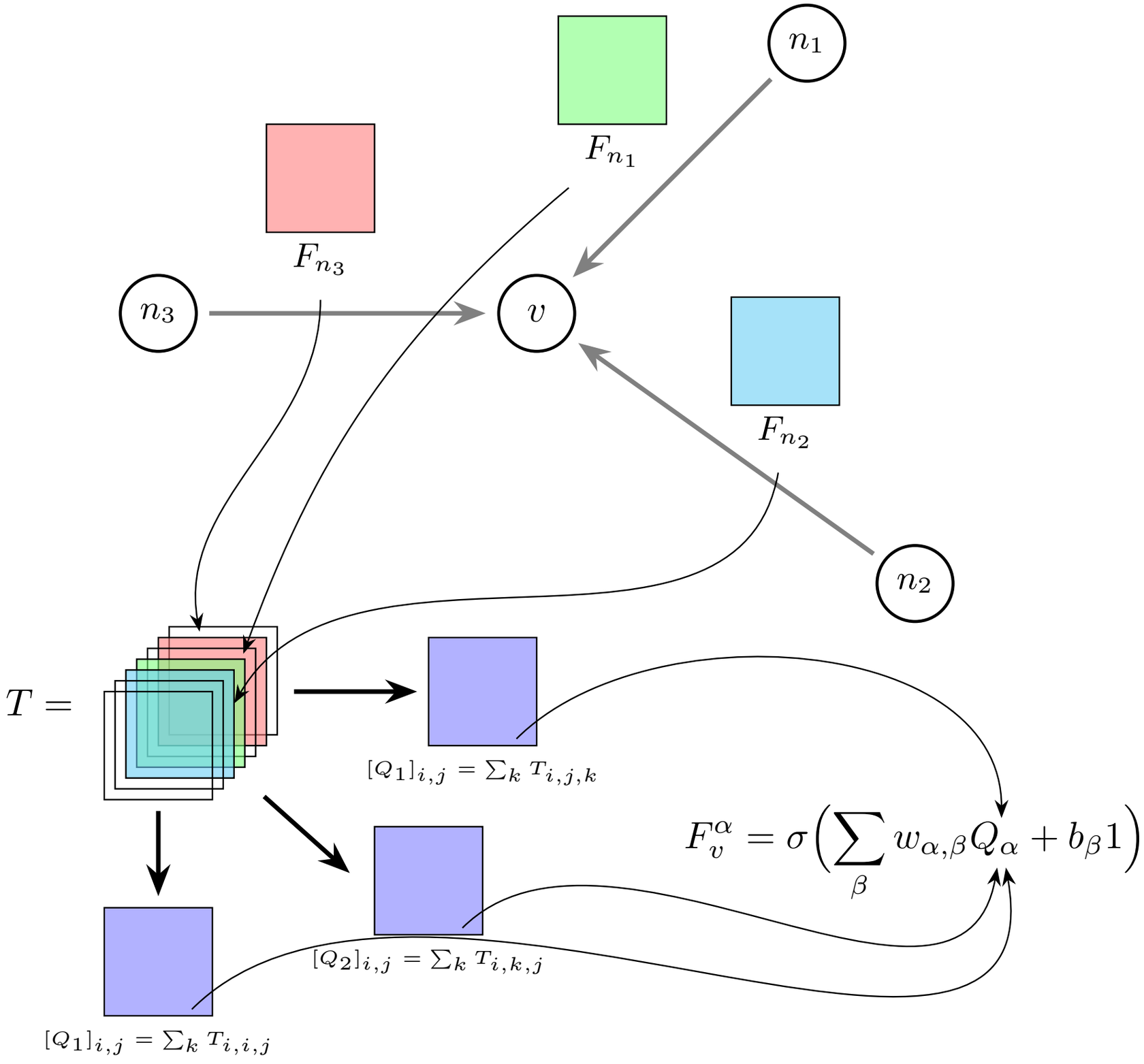
$$\widetilde{F}_{i_1,\ldots,i_k} = \chi_{i_1}^{\;j_1}\, \chi_{i_2}^{\;j_2} \;\cdots\; \chi_{i_k}^{\;j_k}\, F_{j_1,\ldots,j_k} \tag{1}$$

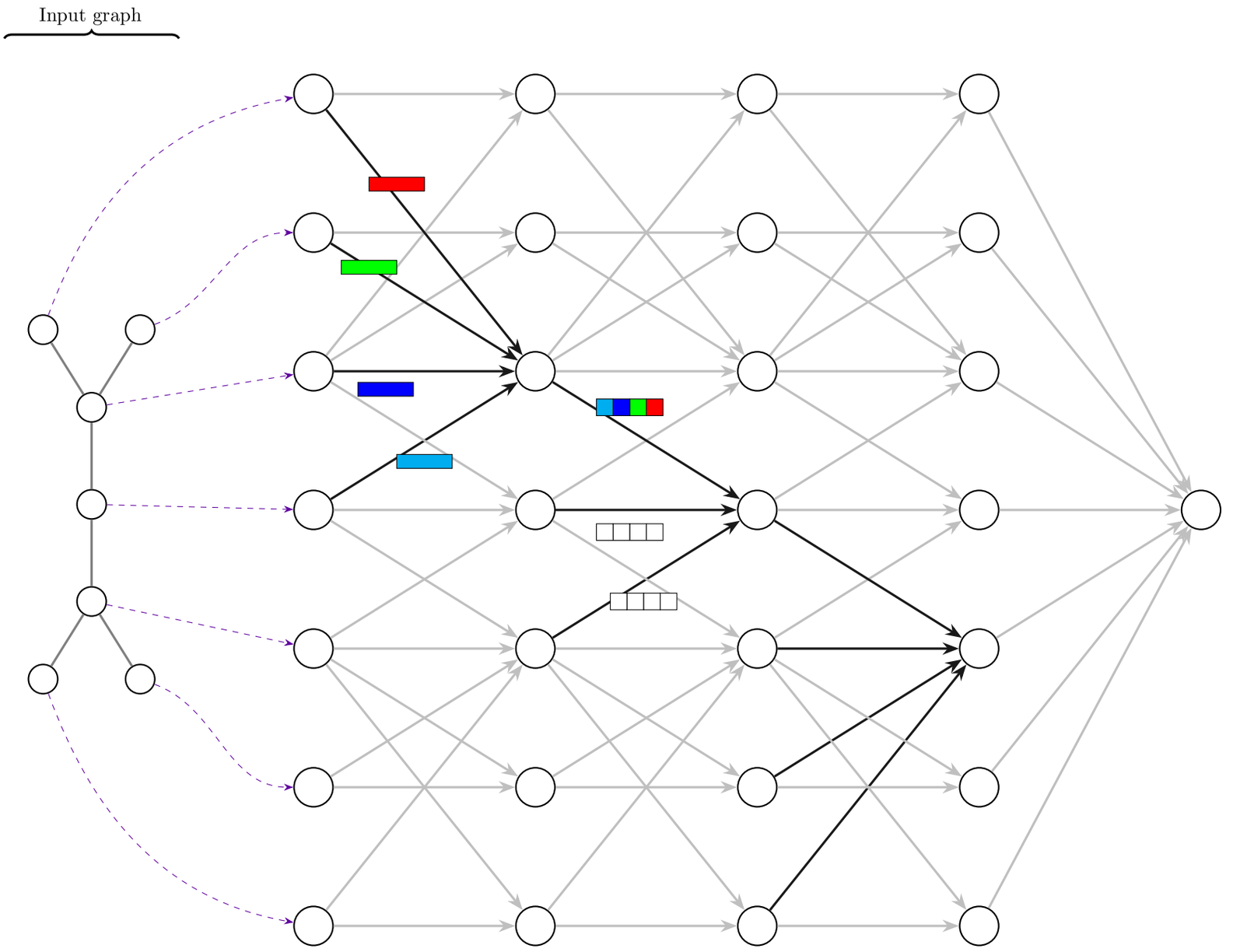
is a k'th order P–tensor with respect to permutations of $(e_{q_1}, \ldots, e_{q_{m'}})$.

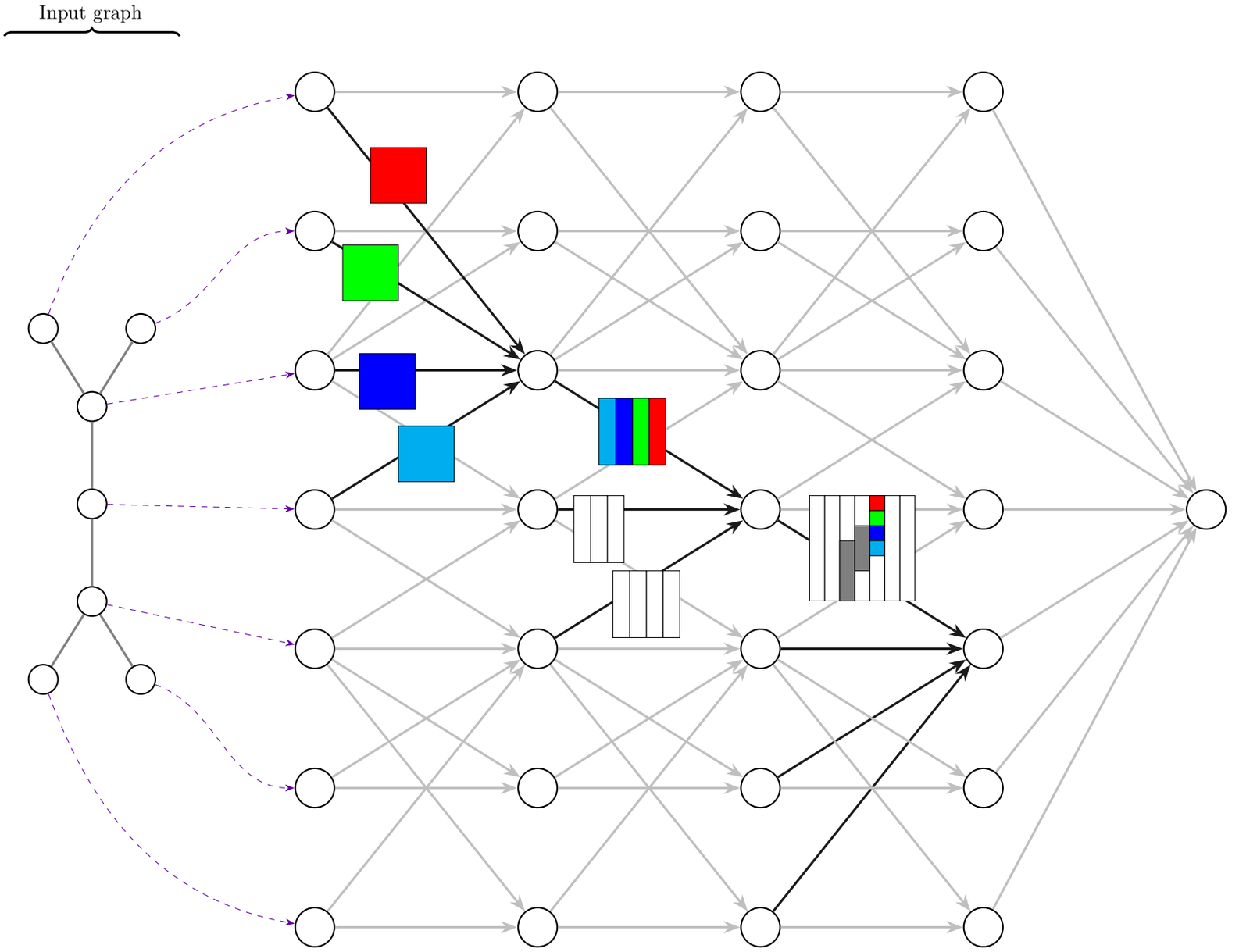
1. Collect all the k'th order activations $F_{c_1}, \ldots, F_{c_s}$ of the children.
2. Promote each activation to $\widetilde{F}_{c_1}, \ldots, \widetilde{F}_{c_s}$.
3. Stack $\widetilde{F}_{c_1}, \ldots, \widetilde{F}_{c_s}$ together into a $k+1$ order tensor T.
4. Optionally form the tensor product of T with $A{\downarrow_{\mathcal{P}_t}}$ to get a $k+3$ order tensor H (otherwise just set $H = T$).
5. Contract H along some number of combinations of dimensions to get s separate lower order tensors $Q_1, \ldots, Q_s$.
6. Mix $Q_1, \ldots, Q_s$ with a matrix $W \in \mathbb{R}^{s' \times s}$ and apply a nonlinearity Υ to get the final activation of the neuron, which consists of the s' output tensors

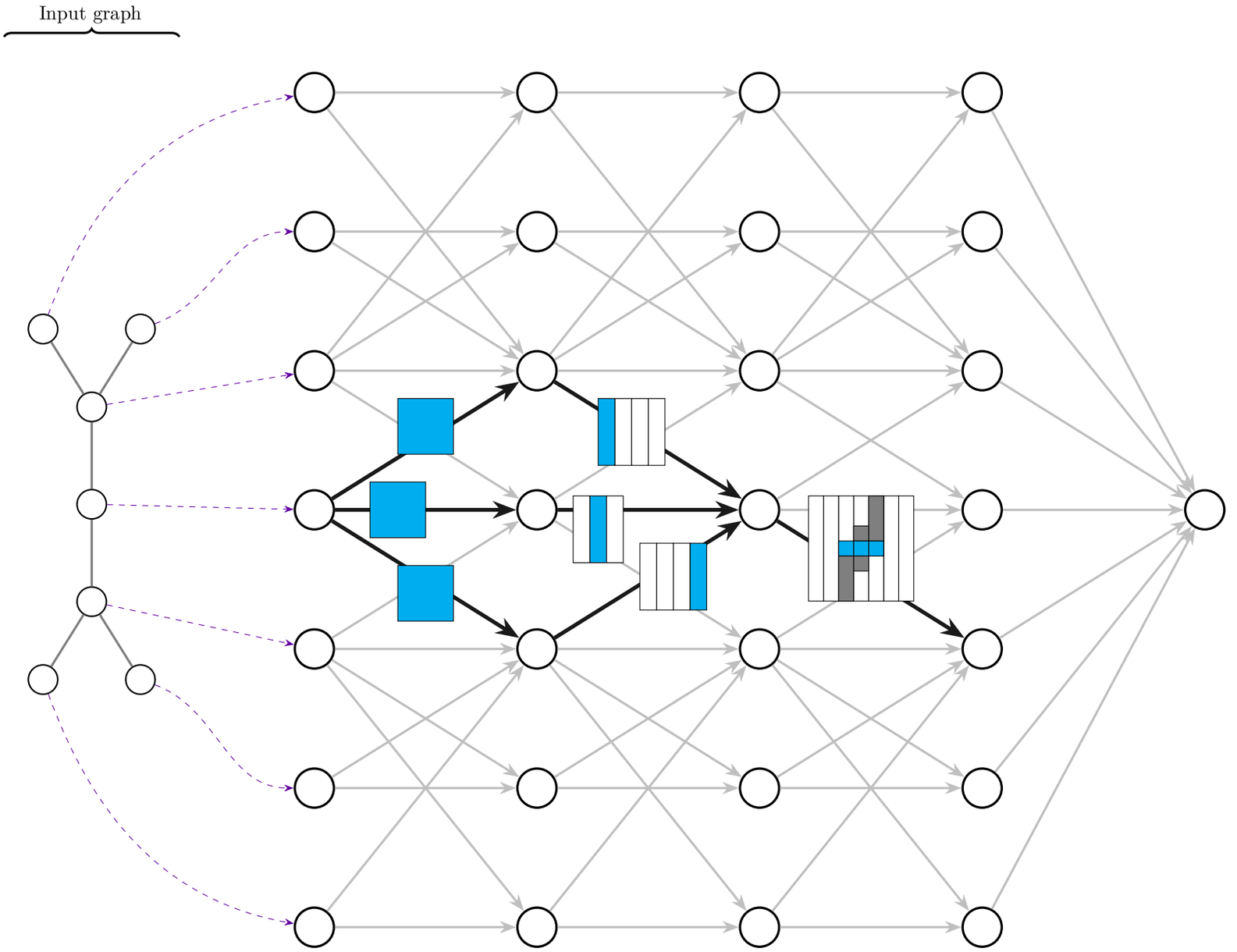
$$F^{(i)} = \Upsilon\Big[\sum_{j=1}^{s} W_{i,j}\, Q_j + b_i\Big] \qquad\qquad i = 1, 2, \ldots s',$$

where the b_i scalars are bias terms.


n_1
F_{n_1}
F_{n_3}
n_3
v
F_{n_2}
n_2
$T =$
$[Q_1]_{i,j} = \sum_k T_{i,j,k}$
$[Q_2]_{i,j} = \sum_k T_{i,k,j}$
$[Q_1]_{i,j} = \sum_k T_{i,i,j}$
$F_v^\alpha = \sigma\left(\sum_\beta w_{\alpha,\beta} Q_\alpha + b_\beta 1\right)$


Input graph


Input graph

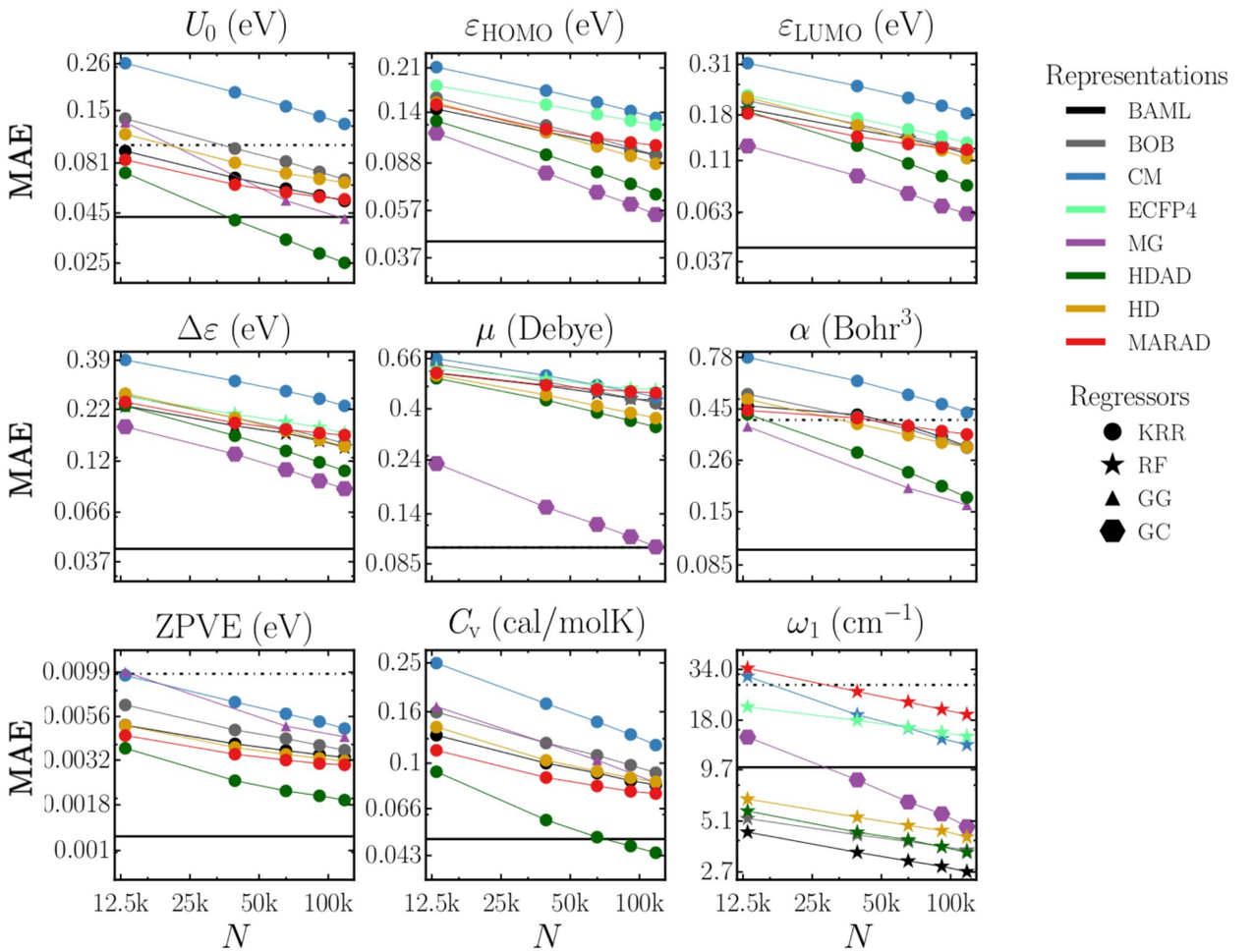

Input graph

HCEP results

Method	Train MAE	Train RMSE	Test MAE	Test RMSE
Lasso	0.863	1.190	0.867	1.437
Ridge Regression	0.849	1.164	0.854	1.376
Random Forest	0.999	1.331	1.004	1.799
Gradient Boosted Tree	0.676	0.939	0.704	1.005
Weisfeiler-Lehman Graph Kernel	0.805	1.111	0.805	1.096
Neural Graph Fingerprint	0.848	1.187	0.851	1.177
Learning Convolution Neural Network	0.704	0.972	0.718	0.973
CCN 2D	0.562	0.773	0.570	0.773

[Duvenaud et al., 2015] [Kriege, 2016] [Niepert, 2016]
[Hachmann et al., 2011]

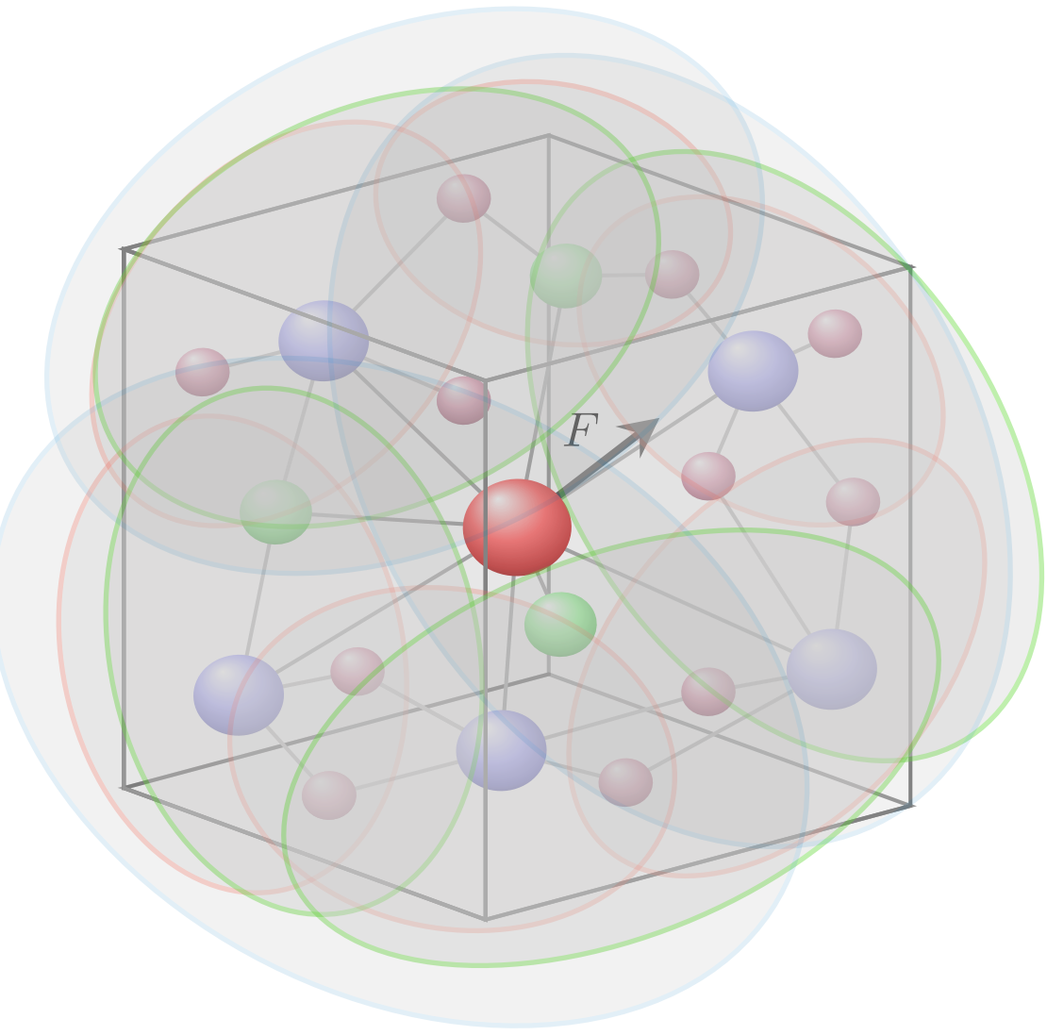
QM9

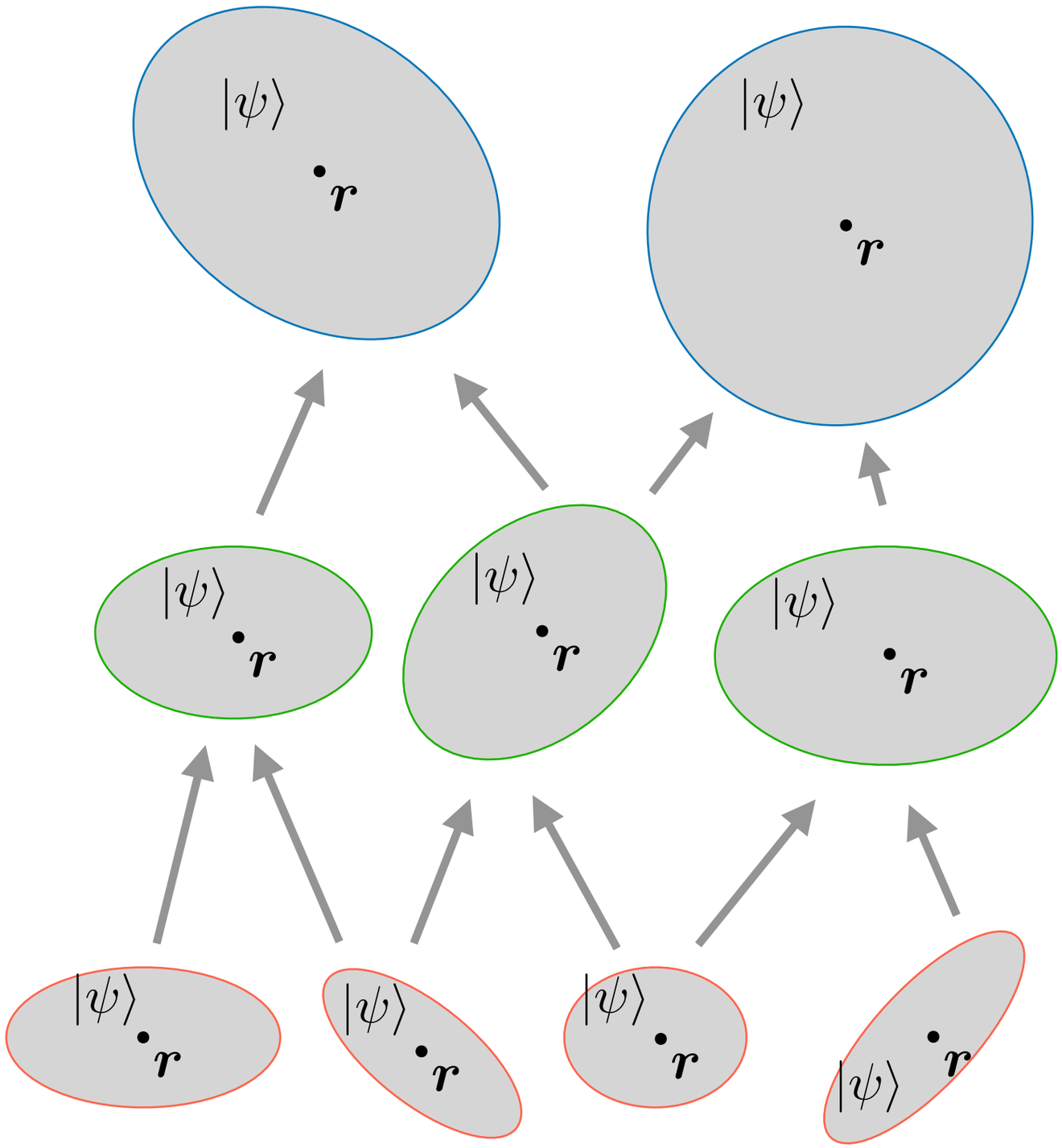


	CCN	DFT error
α (Bohr^3)	**0.22**	0.4
C_v (cal/(mol K))	**0.07**	0.34
G (eV)	**0.06**	0.1
GAP (eV)	**0.12**	1.2
H (eV)	**0.06**	0.1
HOMO (eV)	**0.09**	2.0
LUMO (eV)	**0.09**	2.6
μ (Debye)	0.48	**0.1**
ω_1 (cm^{-1})	**2.81**	28
R_2 (Bohr^2)	4.00	-
U (eV)	**0.06**	0.1
U_0 (eV)	**0.05**	0.1
ZPVE (eV)	**0.0039**	0.0097

3. N-body networks

[K., 2018] [Anderson, Hy & K, in preparation]


F


$|\psi\rangle$
$\boldsymbol{r}$

1. Behavior of states under action of G:

$$|\psi_i\rangle \mapsto \rho_i(g)\,|\psi_i\rangle$$

where $\{\rho_i(g)\}_{g\in G}$ is some representation of G.

2. Behavior of $\boldsymbol{r}_i$ under action of G:

$$\boldsymbol{r}_i \mapsto \rho_0(g)\,\boldsymbol{r}_i$$

General form of aggregation:

$$|\psi\rangle = \sigma\Big(W \sum_i (\boldsymbol{r}_i - \boldsymbol{r})^{\otimes k} \otimes |\psi_i\rangle^{\otimes p} \Big)$$

What form does W need to take to ensure that

$$|\psi\rangle \mapsto \rho(g)\,|\psi\rangle$$

[c.f. “Tensor Field Networks” by Thomas et al., 2018]

The sum $\sum_i (\boldsymbol{r}_i - \boldsymbol{r})^{\otimes k} \otimes |\psi_i\rangle^{\otimes p}$ transform according to

$$\rho_0^{\otimes k} \otimes \rho^{\otimes p}$$

which decomposes into irreducibles in the form

$$\left(\rho_0^{\otimes k} \otimes \rho^{\otimes p}\right)(g) = \bigoplus_{\ell} \bigoplus_{m=1}^{\kappa(\ell)} C_m^\ell \cdot \rho^{(i)}(g) \cdot {C_m^\ell}^\dagger .$$

Let

$$\phi{\downarrow_{\ell,m}} = C_{\ell,m}^\dagger \sum_i (\boldsymbol{r}_i - \boldsymbol{r})^{\otimes k} \otimes |\psi_i\rangle^{\otimes p} \ .$$

Theorem. $|\psi\rangle$ is covariant if and only if the aggregation function is of the form

$$|\phi\rangle = \bigoplus_{\ell} \bigoplus_{m'=1}^{\kappa'(\ell)} w^{\ell}_{m,m'} \sum \phi{\downarrow_{\ell,m}}$$

4. Implementation

TensorFlow

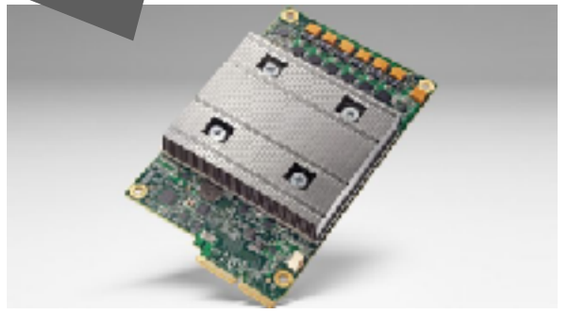
PYTORCH

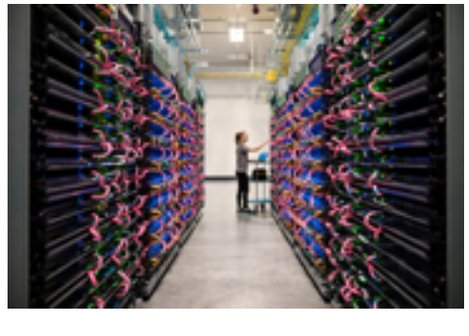
mxnet

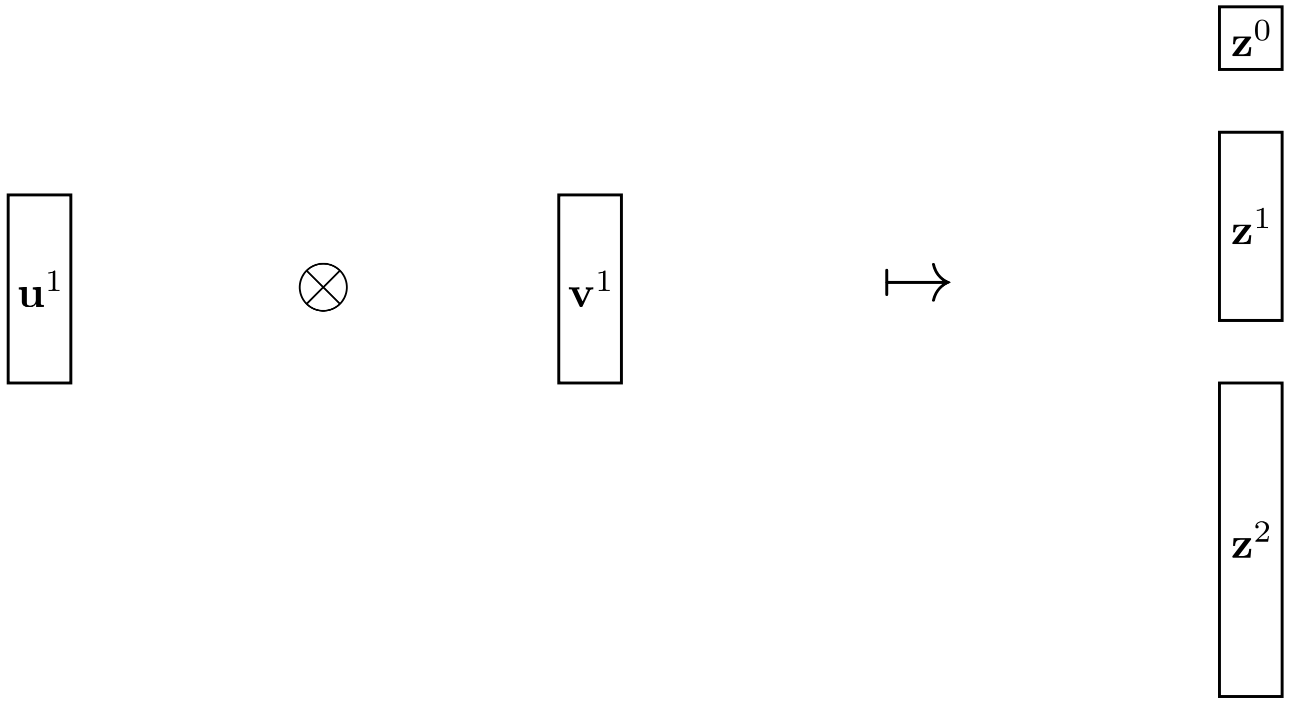
Caffe2

Highly optimized custom operator

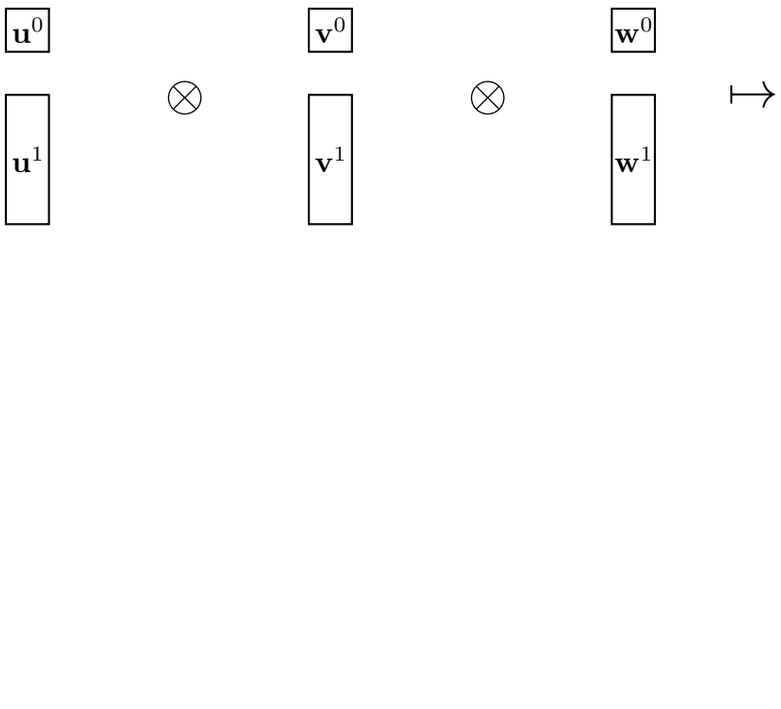
Computation engine

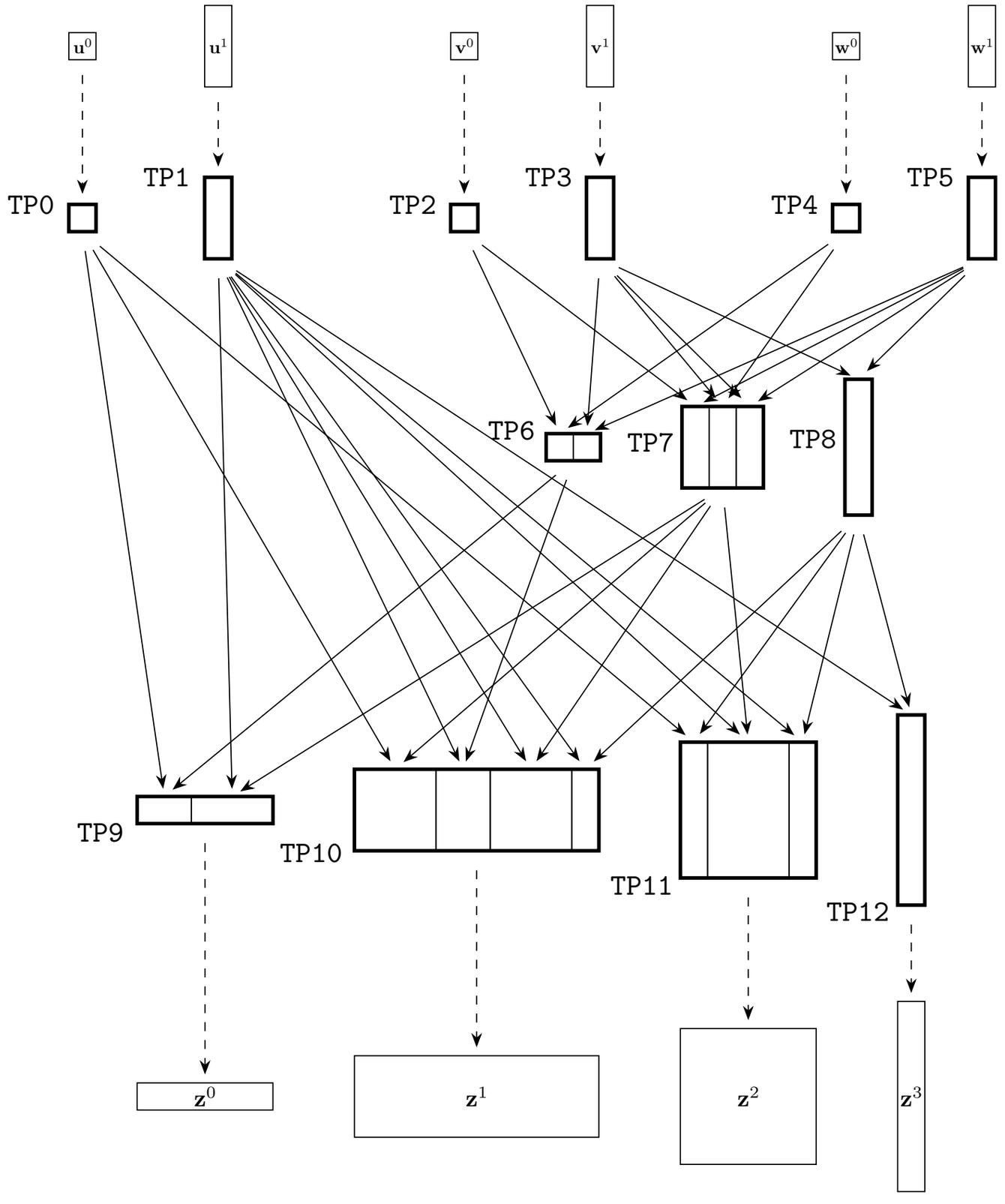




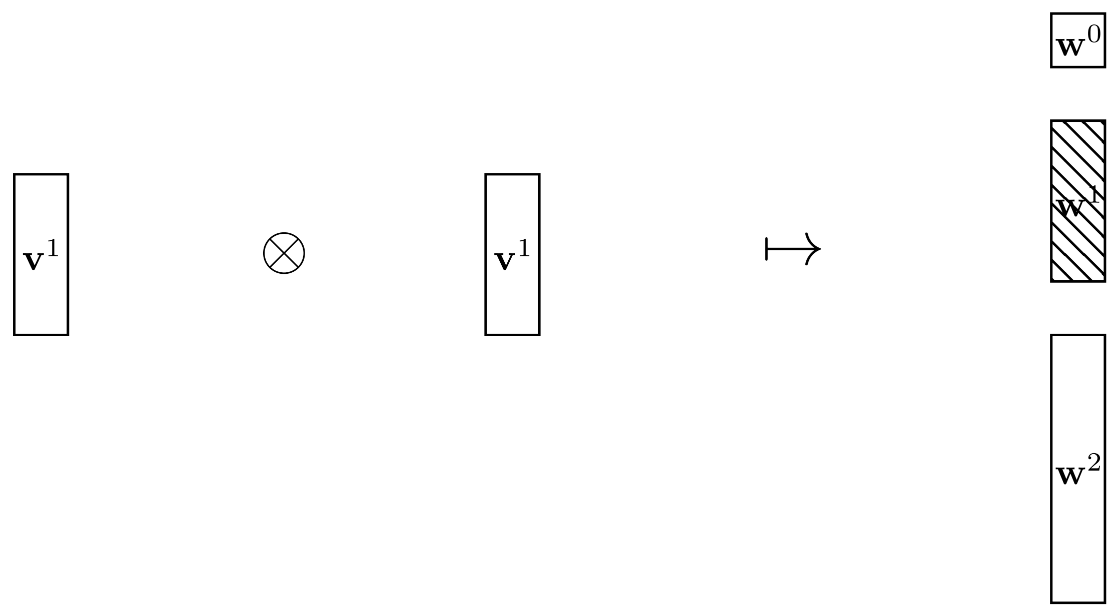


$\mathbf{u}^1$
$\otimes$
$\mathbf{v}^1$
$\mapsto$
$\mathbf{z}^0$
$\mathbf{z}^1$
$\mathbf{z}^2$

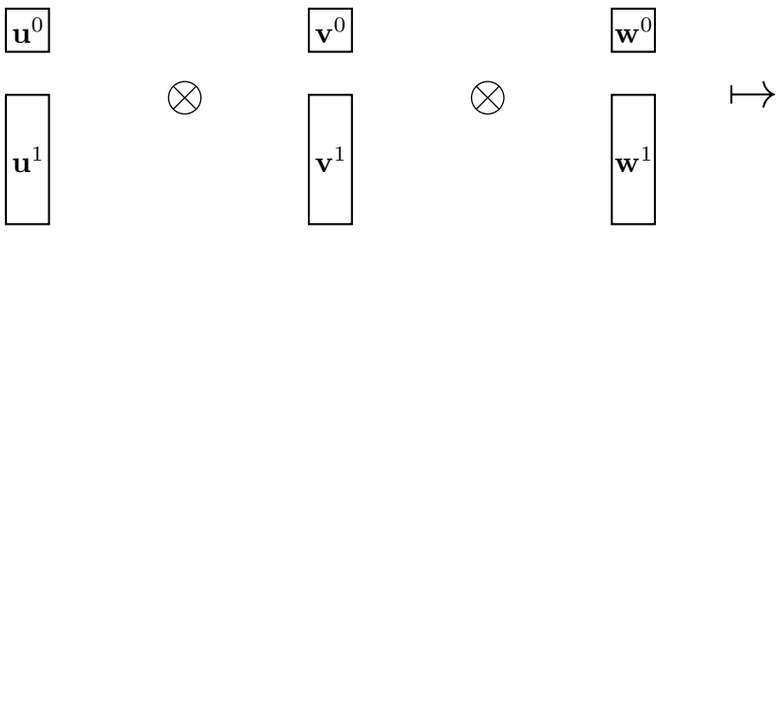

$\mathbf{u}^0$
$\mathbf{u}^1$
$\otimes$
$\mathbf{v}^0$
$\mathbf{v}^1$
$\otimes$
$\mathbf{w}^0$
$\mathbf{w}^1$
$\mapsto$
$\mathbf{u}^0 \otimes \mathbf{v}^0 \otimes \mathbf{w}^0$
$\mathbf{u}^0 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^0 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^0$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^0 \otimes \mathbf{v}^0 \otimes \mathbf{w}^1$
$\mathbf{u}^0 \otimes \mathbf{v}^1 \otimes \mathbf{w}^0$
$\mathbf{u}^1 \otimes \mathbf{v}^0 \otimes \mathbf{w}^0$
$\mathbf{u}^0 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^0 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^0$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$

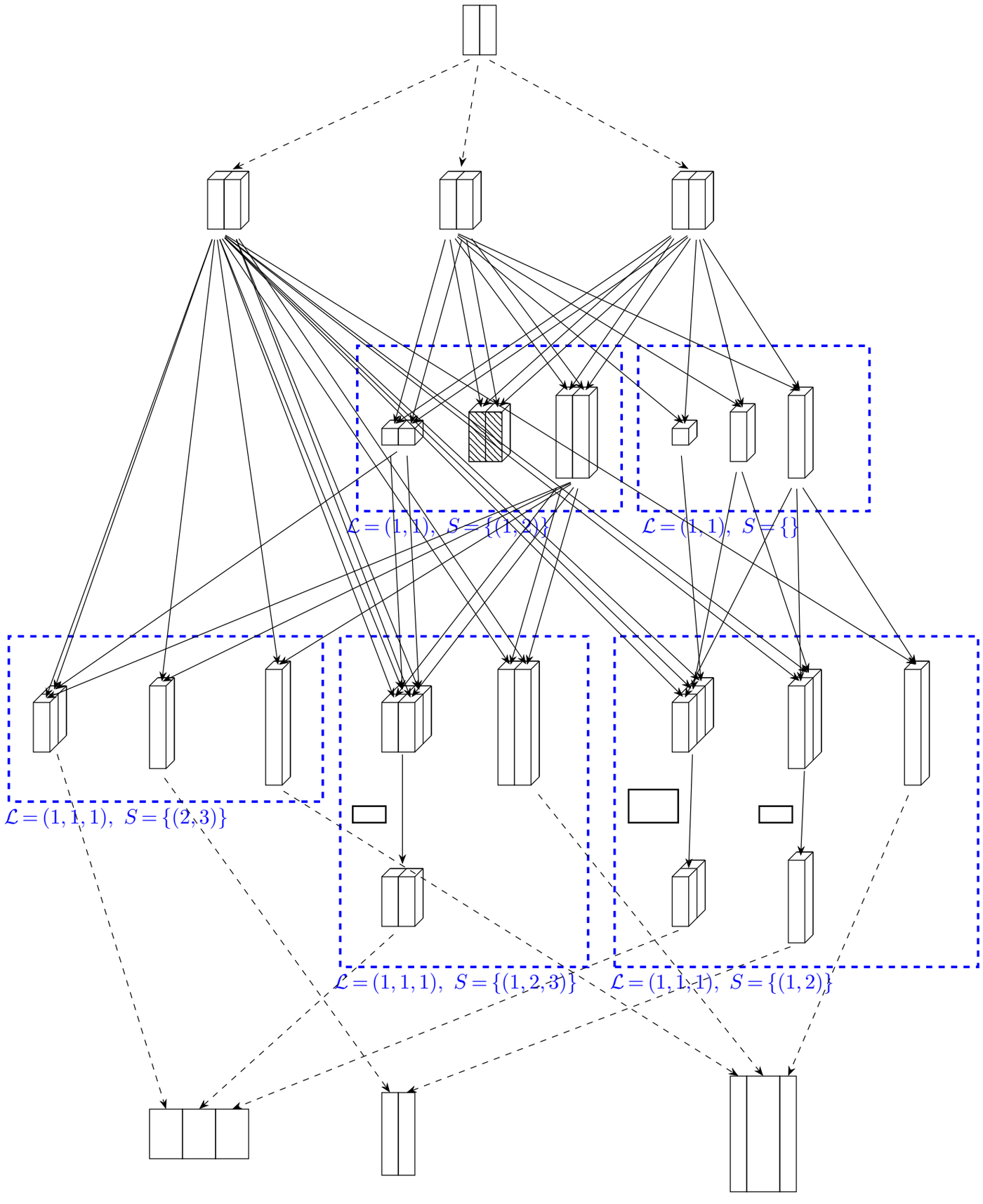

$\mathbf{u}^0$
$\mathbf{u}^1$
$\mathbf{v}^0$
$\mathbf{v}^1$
$\mathbf{w}^0$
$\mathbf{w}^1$
TP0
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
TP9
TP10
TP11
TP12
$\mathbf{z}^0$
$\mathbf{z}^1$
$\mathbf{z}^2$
$\mathbf{z}^3$

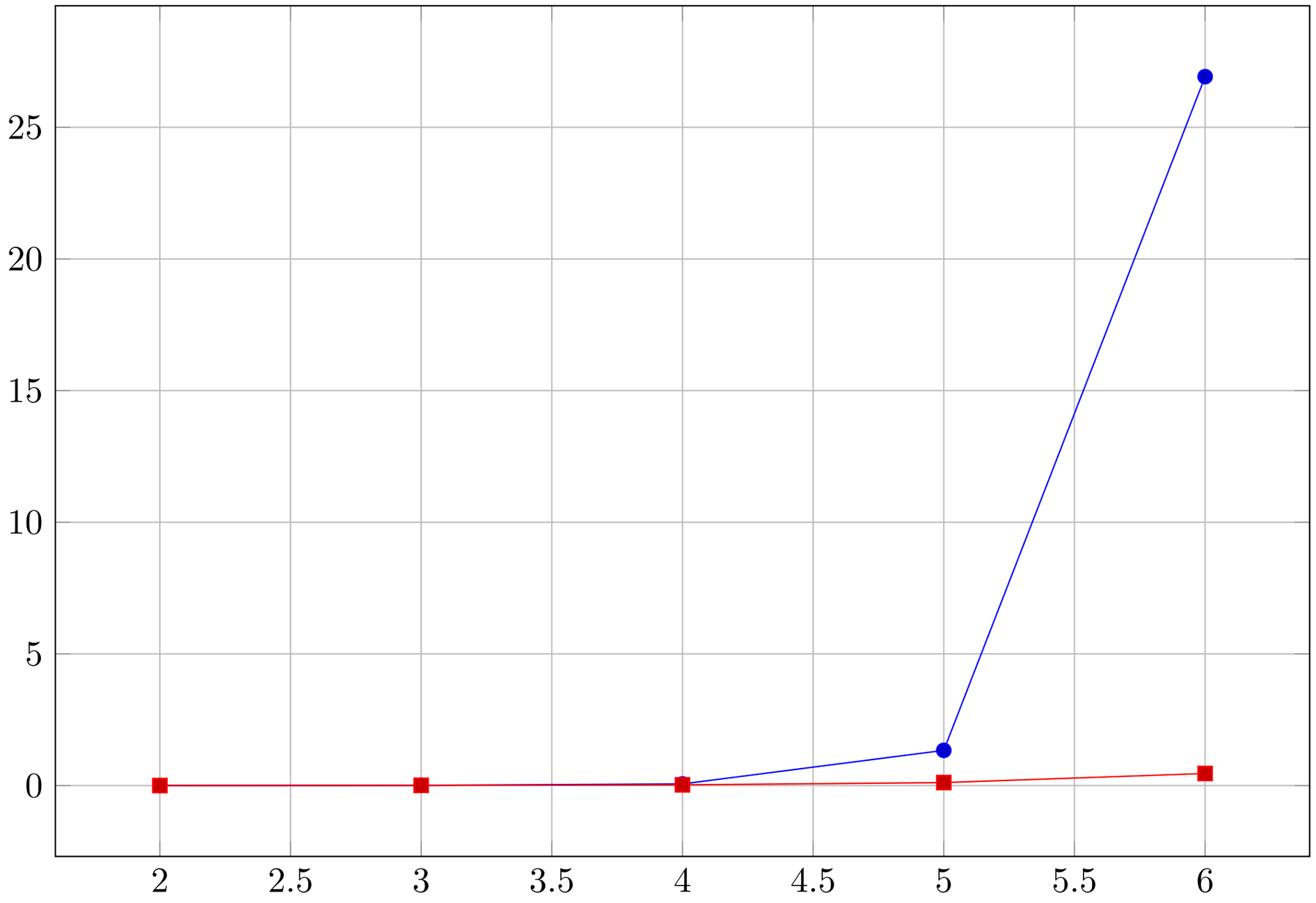
```
1   TPprogram  CGproduct(){
2     TPpart0 (l=0)[0m (n=1){
3       input(0,0);
4     }
5     TPpart1 (l=1) (n=1){
6       input(0,1);
7     }
8     TPpart2 (l=0) (n=1){
9       input(1,0);
10    }
11    TPpart3 (l=1) (n=1){
12      input(1,1);
13    }
14    TPpart4 (l=0) (n=1){
15      input(2,0);
16    }
17    TPpart5 (l=1) (n=1){
18      input(2,1);
19    }
20    TPpart6 (l=0) (n=2){
21      CG(2,4)[0];
22      CG(3,5)[1];
23    }
24    TPpart7 (l=1) (n=3){
25      CG(2,5)[0];
26      CG(3,4)[1];
27      CG(3,5)[2];
28    }
29    TPpart8 (l=2) (n=1){
30      CG(3,5)[0];
31    }
32    TPpart9 (l=0) (n=5){
33      output(0);
34      CG(0,6)[0];
35      CG(1,7)[2];
36    }
37    TPpart10 (l=1) (n=9){
38      output(1);
39      CG(0,7)[0];
40      CG(1,6)[3];
41      CG(1,7)[5];
```



$\mathbf{v}^1$
$\otimes$
$\mathbf{v}^1$
$\mapsto$
$\mathbf{w}^0$
$\mathbf{w}^1$
$\mathbf{w}^2$


$\mathbf{u}^0$
$\mathbf{u}^1$
$\otimes$
$\mathbf{v}^0$
$\mathbf{v}^1$
$\otimes$
$\mathbf{w}^0$
$\mathbf{w}^1$
$\mapsto$
$\mathbf{u}^0 \otimes \mathbf{v}^0 \otimes \mathbf{w}^0$
$\mathbf{u}^0 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^0 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^0$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^0 \otimes \mathbf{v}^0 \otimes \mathbf{w}^1$
$\mathbf{u}^0 \otimes \mathbf{v}^1 \otimes \mathbf{w}^0$
$\mathbf{u}^1 \otimes \mathbf{v}^0 \otimes \mathbf{w}^0$
$\mathbf{u}^0 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^0 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^0$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$
$\mathbf{u}^1 \otimes \mathbf{v}^1 \otimes \mathbf{w}^1$


$\mathcal{L} = (1,1),\ S = \{(1,2)\}$
$\mathcal{L} = (1,1),\ S = \{\}$
$\mathcal{L} = (1,1,1),\ S = \{(2,3)\}$
$\mathcal{L} = (1,1,1),\ S = \{(1,2,3)\}$
$\mathcal{L} = (1,1,1),\ S = \{(1,2)\}$


25
20
15
10
5
0
2
2.5
3
3.5
4
4.5
5
5.5
6

MD-17

	Us (10k)	Deep (95k)	SchNet (1k)	GDML (1k)	GDML (50k)	s-GDML (1k)	DTNN (50k)
aspirin	0.349	0.201	0.250	0.270	**0.129**	0.189	-
benzene	**0.036**	0.065	0.080	0.069	0.074	0.099	0.039
ethanol	0.062	0.055	0.070	0.150	**0.053**	0.069	-
malonaldehyde	0.107	0.092	0.130	0.159	**0.076**	0.099	0.189
naphthalene	**0.069**	0.095	0.200	0.120	0.118	0.120	-
salicylic acid	0.221	**0.106**	0.250	0.120	0.111	0.120	0.500
toluene	**0.079**	0.085	0.160	0.120	0.095	0.099	0.180
uracil	**0.073**	0.085	0.130	0.111	0.074	0.099	-

[Zhang, Han, Wang, Car, Weinan, 2017]

[Schütt, Sauceda, Kindermands, Tkatchenko, Müller, 2017]

[Schütt, Arbabzadah, Chmiela, Müller, Tkatchenko, 2017]

[Chmiela, Tkatchenko, Sauceda, Poltravsky, Schütt, Müller, 2017]

Conclusions

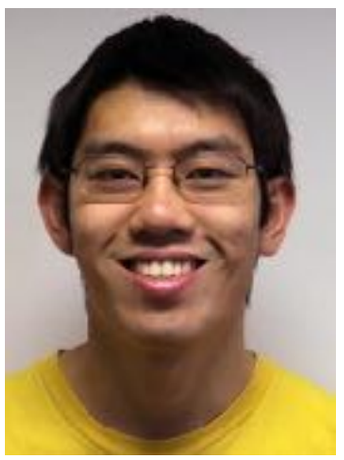
1. Compositional structure

2. Covariance

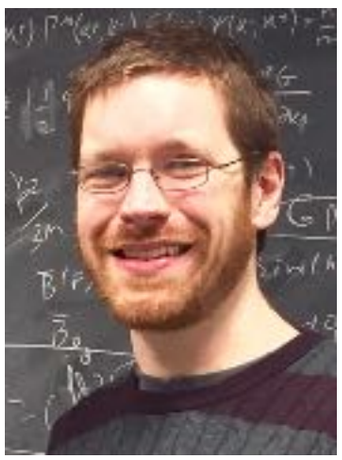
   ➔ Fourier space activations

Shubhendu Trivedi

Hy Trong Son



Horace Pan



Brandon Anderson