May 6, 2020 **Physics ∩ ML**

Qualcomm

Natural Graph Convolutions

Taco Cohen

Sr. Staff Engineer, Qualcomm AI Research,
Qualcomm Technologies Netherlands B.V.

Outline

- Equivariant Networks: the story so far
  - General concepts
  - Overview of equivariant networks
- Natural Graph Networks
  - Graph CNNs intro
  - Limitations
  - Graph Symmetries
  - Equivariant Message Passing
- Category theoretic formulation

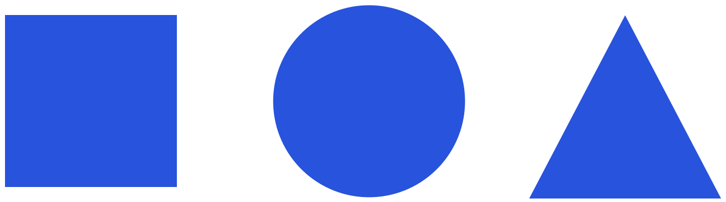
Equivariant Networks

The story so far

Symmetry

Definition

“A transformation of an object that leaves the object invariant”



- In ML: symmetries of distributions, label functions, parameter spaces
- Knowledge of symmetry provides a strong inductive bias
  - Example: Laws of physics are almost completely determined by a handful of symmetries

Invariance vs Equivariance

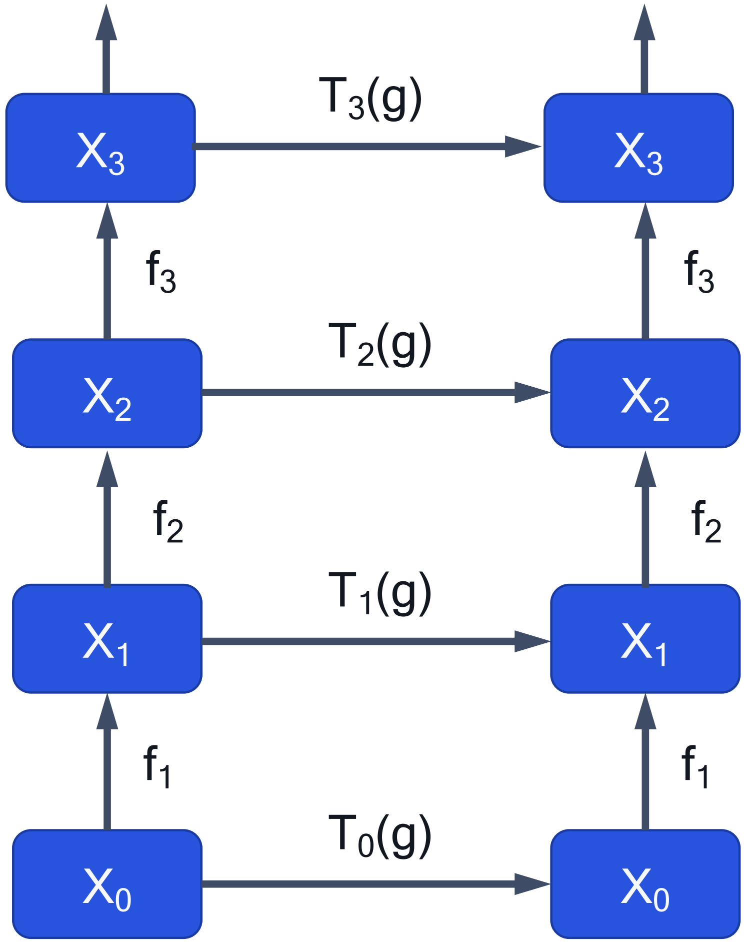
The "Picasso Problem":

Why invariance is not enough in DL

Image credit: Ninara on flickr
https://www.flickr.com/photos/ninara/35077385225

Equivariant Networks

General setup



Ingredients:

- Feature spaces X_i
- Maps f_i between them ("Layers")
- A group G
- Group representations ("Transformation laws") T_i of G for each feature space X_i

Equivariance

$$f_i \circ T_{i-1}(g) = T_i(g) \circ f_i$$

A Design Principle for Neural Network Architectures

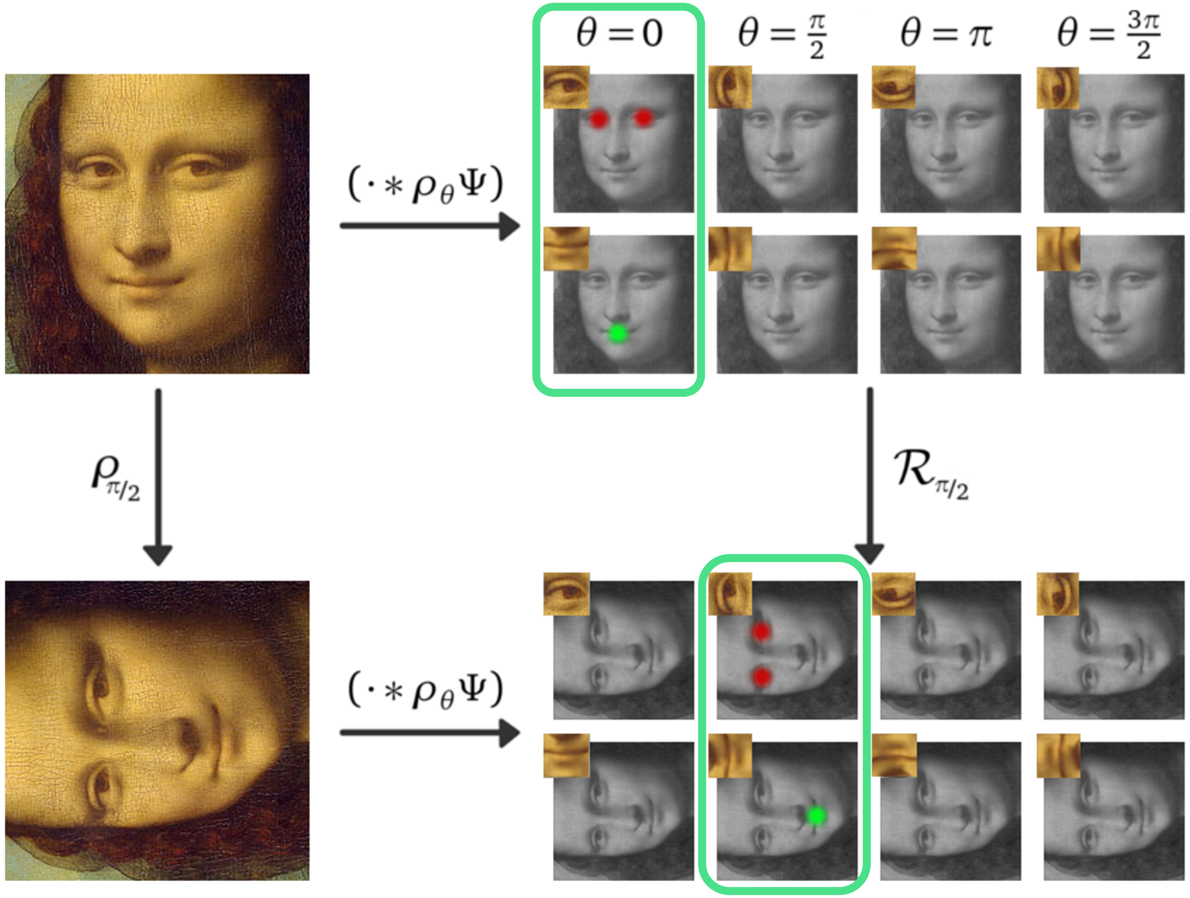
Equivariance to Symmetry Transformations

Examples

Data	Images, Audio, …	Signals on Graphs, Point Clouds	Signals on homogeneous space	Signals on manifolds / meshes
Symmetries	Translations; Rotations	Permutations	Global symmetries G	Structure group G & Gauge group Aut(P)
Architecture	CNNs; G-CNNs	Graph NNs, PointNet	Group-equivariant nets (G-CNNs)	Gauge CNNs

Overview of Equivariant Nets

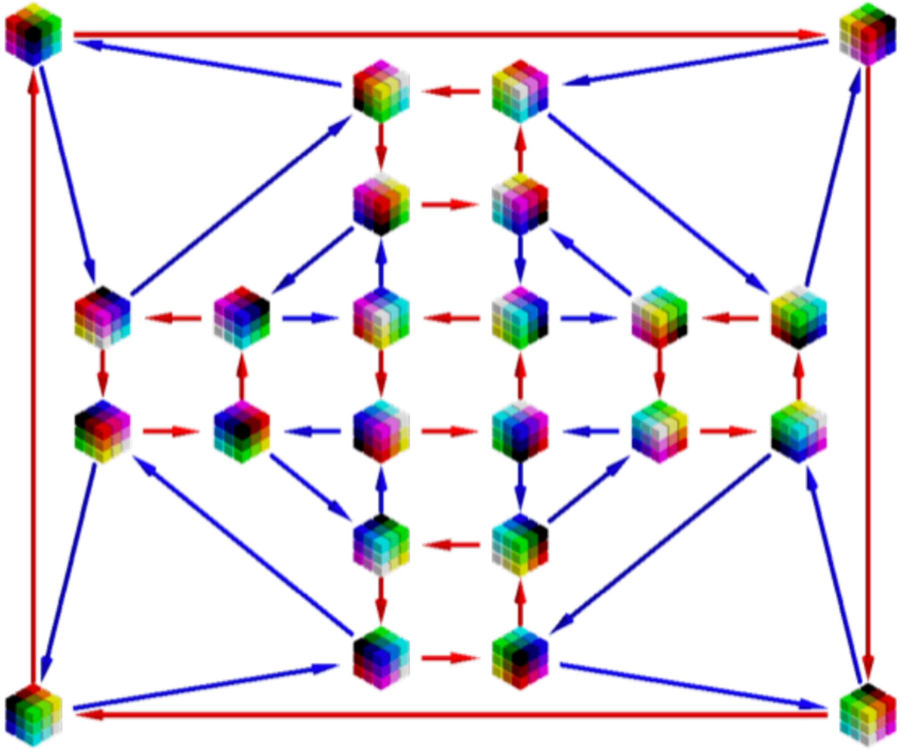
Regular G-CNNs

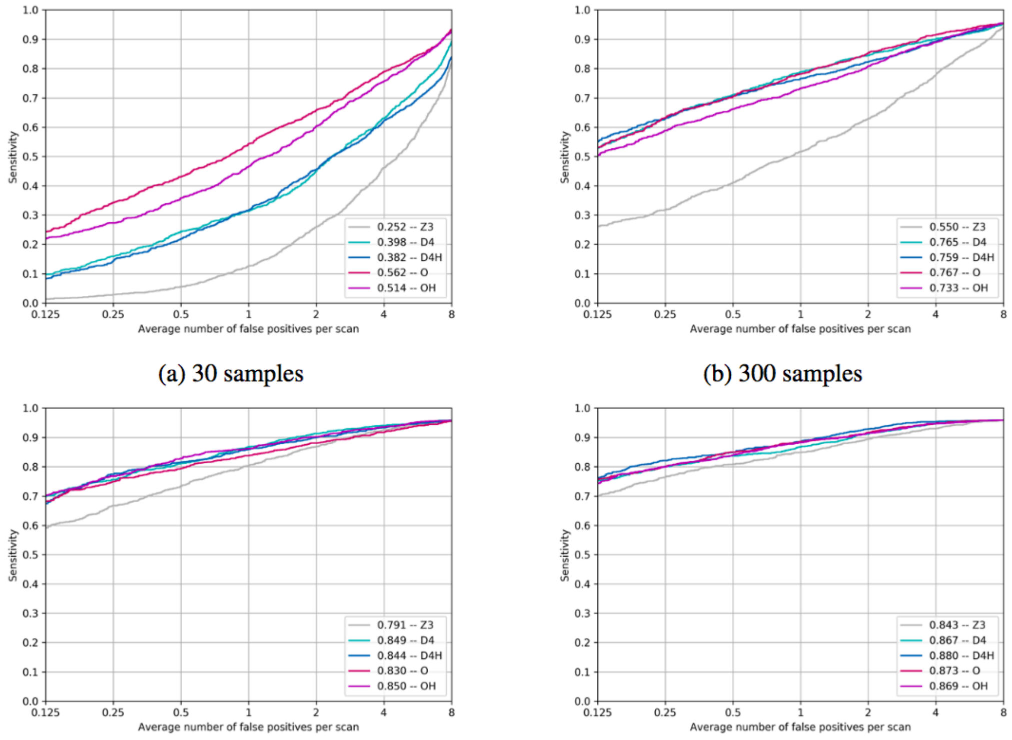


T.S. Cohen & M. Welling, *Group Equivariant Convolutional Networks.* ICML 2016

Regular G-CNNs in 3D

Application to pulmonary nodule detection in CT scans





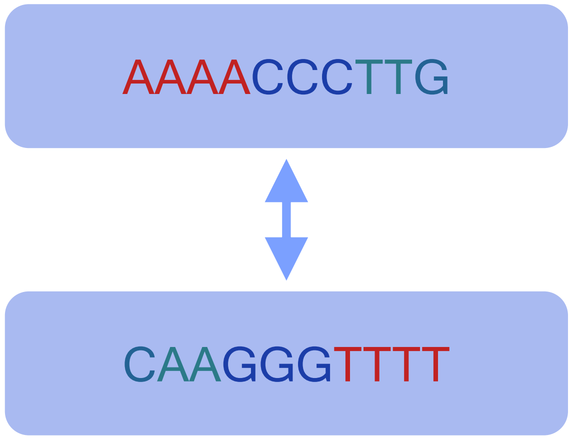
(a) 30 samples (b) 300 samples (c) 3,000 samples (d) 30,000 samples

Figure 4: FROC curves for all groups per training set size.

M. Winkels, T.S. Cohen, *Pulmonary Nodule Detection in CT Scans with Equivariant CNNs,* Medical Image Analysis, 2019
M. Winkels, T.S. Cohen, *3D G-CNNs for Pulmonary Nodule Detection*. MIDL 2018.
D. Worrall, G. Brostow, *CubeNet: Equivariance to 3D Rotation and Translation.* ECCV 2018

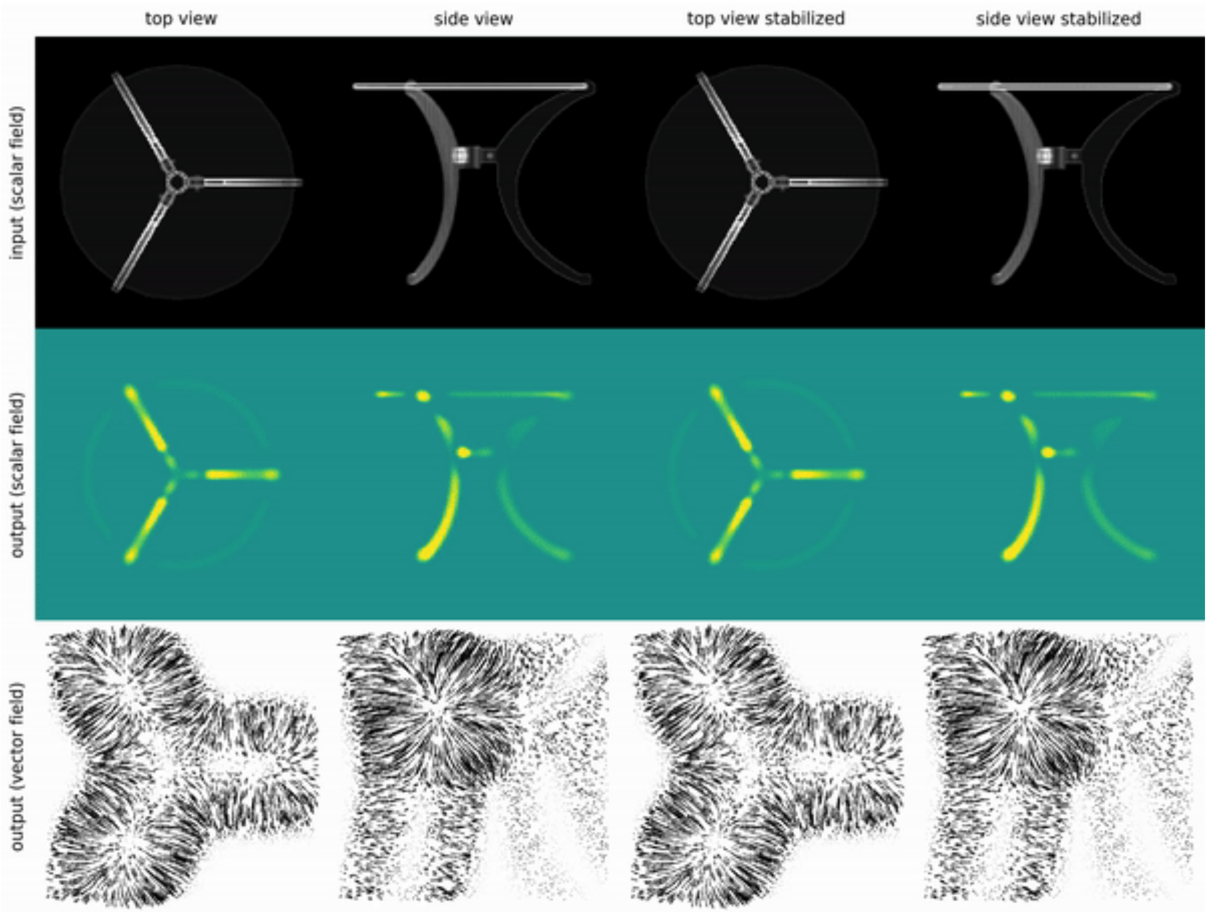
DNA Sequences

- Reverse-complement symmetry



- $\mathbb{Z} \rtimes C_2$ Equivariant CNN
  - Lunter, G., & Brown, R. *An Equivariant Bayesian Convolutional Network predicts recombination hotspots and accurately resolves binding motifs*, *Bioinformatics*, Volume 35, Issue 13, 2019

Steerable CNNs, Harmonic & Tensor Field Networks

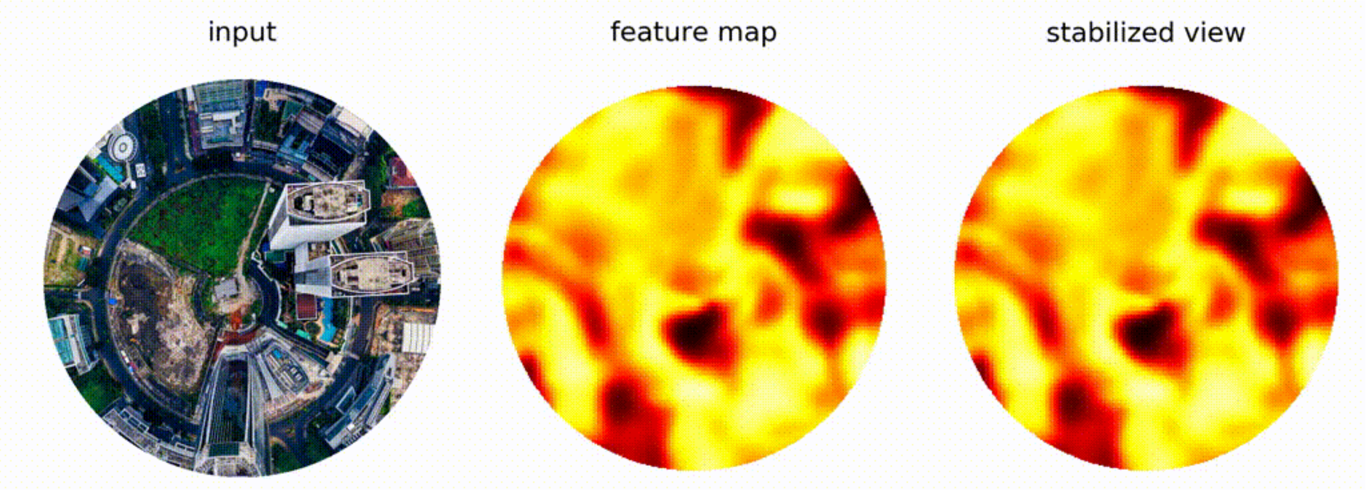


https://www.youtube.com/watch?v=ENLJACPHSEA

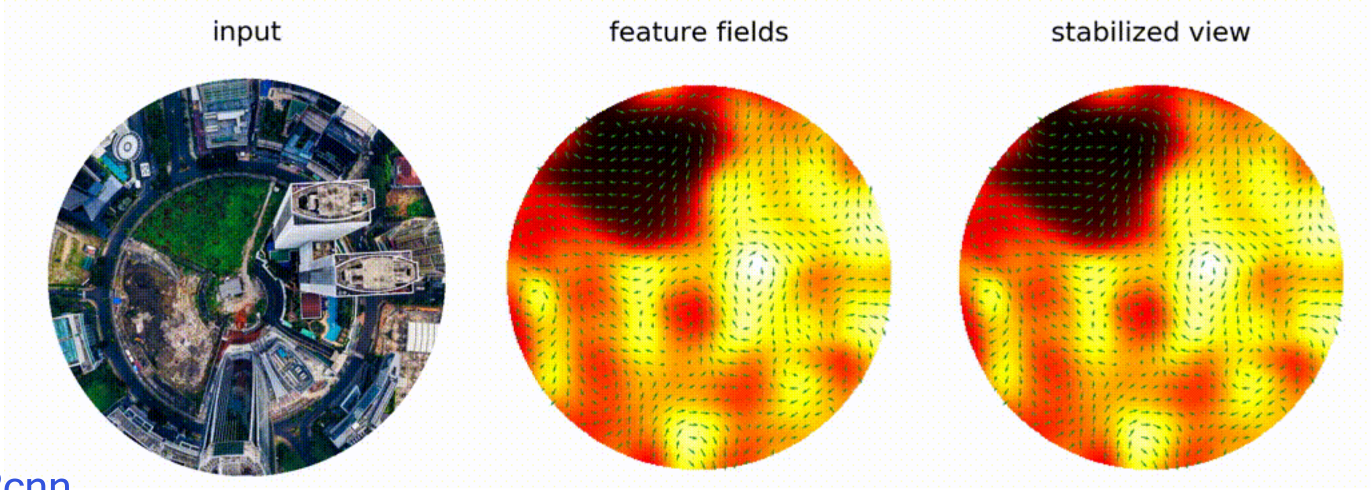
Cohen, T. S., & Welling, M. (2017). Steerable CNNs. In *ICLR*.
Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. (2017). Harmonic Networks: Deep Translation and Rotation Equivariance. In *CVPR*.
M. Weiler, W. Boomsma, M. Geiger, M. Welling, T.S. Cohen, *3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data*, NIPS, 2018
Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., & Riley, P. (2018). *Tensor Field Networks: Rotation- and Translation-Equivariant Neural Networks for 3D Point Clouds*.
Kondor, R. (2018). *N-body Networks: a Covariant Hierarchical Neural Network Architecture for Learning Atomic Potentials. arXiv.*
T. Son Hy, S. Trivedi, B.M. Anderson, R. Kondor (2018). *Predicting Molecular Properties with Covariant Compositional Networks,* JCP special issue on data enabled theoretical chemistry.
https://atomicarchitects.github.io

E(2) & E(3) Steerable CNN

CNN



G-CNN

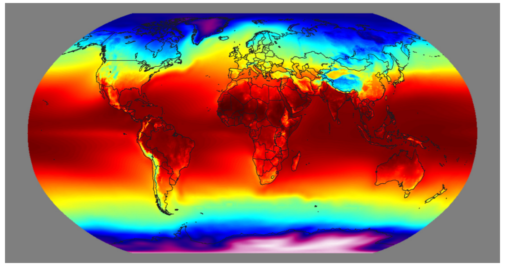


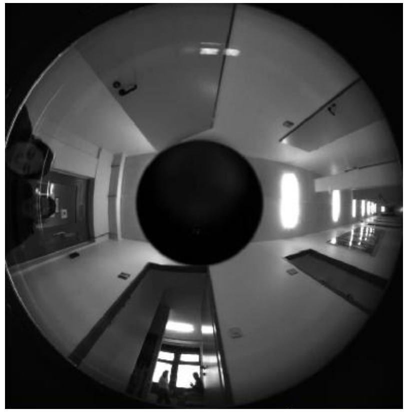
https://github.com/QUVA-Lab/e2cnn
https://github.com/e3nn/e3nn

Spherical CNNs

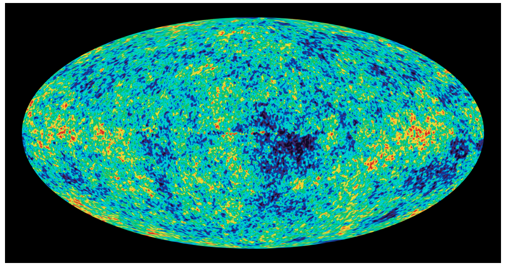
Omnidirectional vision

Earth sciences[1]





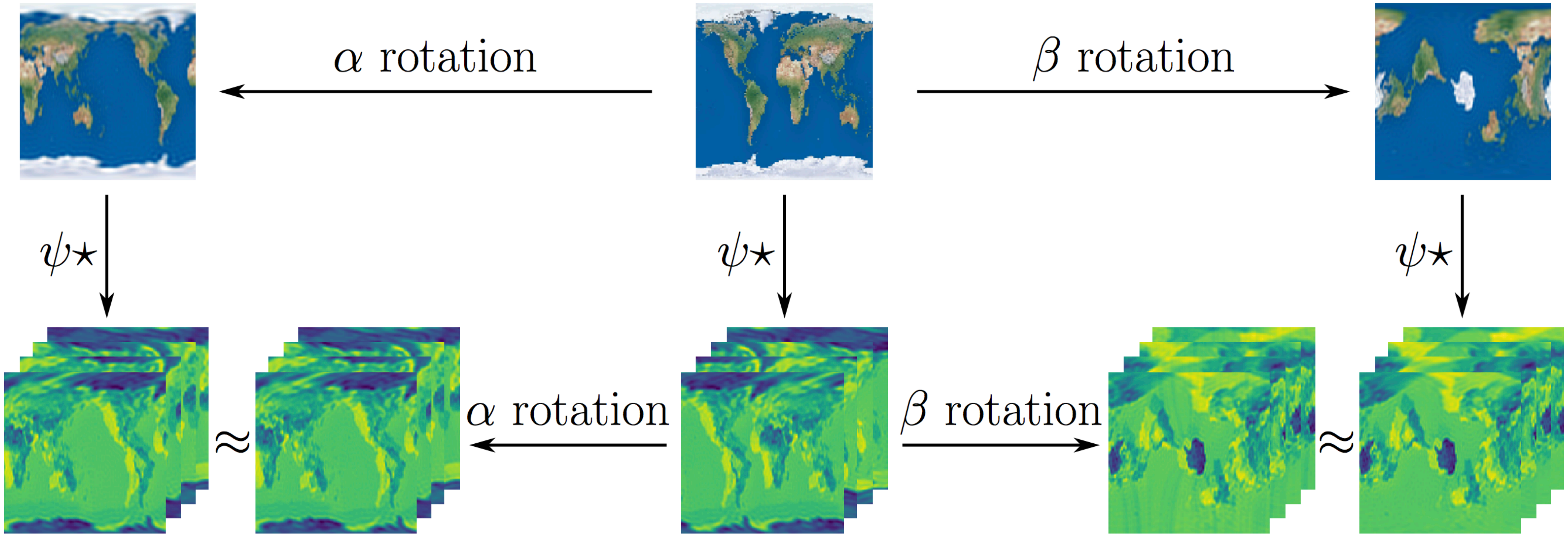
Cosmology



T.S. Cohen, M. Geiger, J. Koehler, M. Welling, Spherical CNNs. ICLR 2018.
Esteves, C., Allen-Blanchette, C., Makadia, A., & Daniilidis, K. Learning SO(3) Equivariant Representations with Spherical CNNs, ECCV 2018.
Kondor, R., Lin, Z., & Trivedi, S. Clebsch-Gordan Nets: A Fully Fourier Space Spherical Convolutional Neural Network. NeurIPS 2018
… and many more

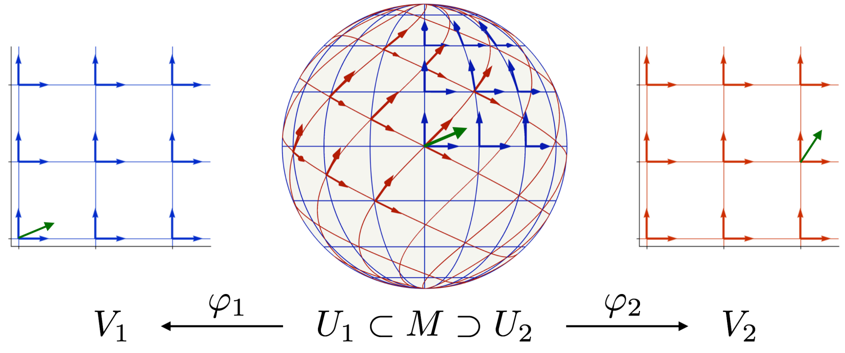
[1]By User Dragons flight (Wikimedia Commons, based on) [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

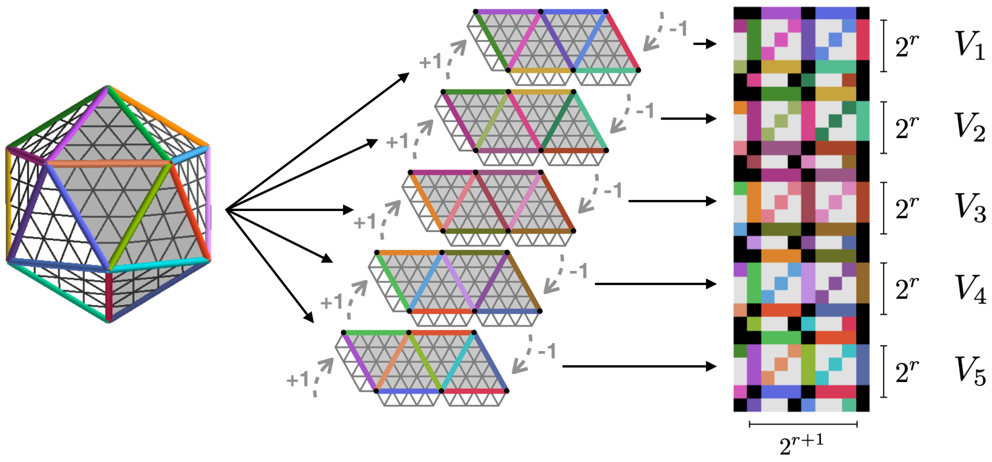
Equivariance of Spherical CNNs

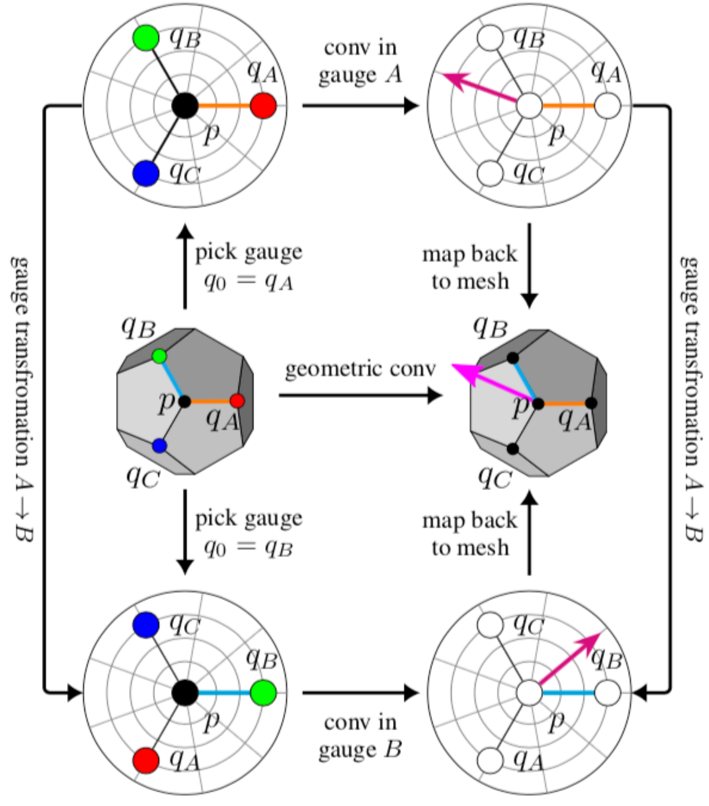


T.S. Cohen, M. Geiger, J. Koehler, M. Welling, Spherical CNNs. ICLR 2018.

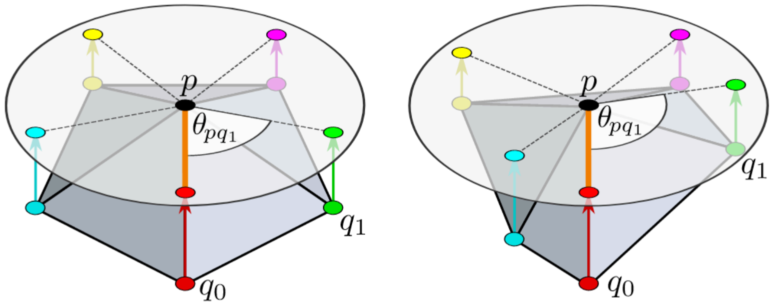
Gauge, manifold & mesh CNNs







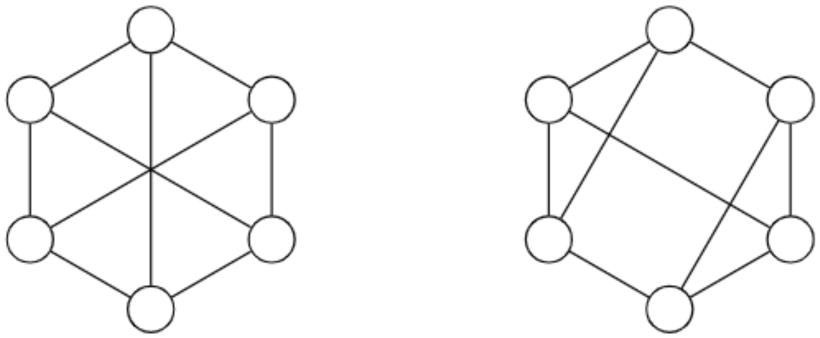
(b) Convolution from scalar to vector features.



D. Boscaini, J. Masci, S. Melzi, M.M. Bronstein, U. Castellani, and P. Vandergheynst, *Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks.* CGF 2015
J. Masci, D. Boscaini, M.M. Bronstein, and P. Vandergheynst, *Geodesic convolutional neural networks on riemannian manifolds.* ICCVW, 2015
T.S. Cohen, M. Weiler, B. Kicanaoglu, M. Welling, *Gauge Equivariant Convolutional Networks and the Icosahedral CNN,* ICML 2019
P. de Haan, M. Weiler, T. Cohen, M. Welling, *Gauge Equivariant Mesh CNNs: Anisotropic convolutions on geometric graphs,* 2020
B. Kicanaoglu, P. de Haan, T. Cohen, *Gauge Equivariant Spherical CNNs,* 2020

Graphs & Point Clouds

- Point clouds are *sets* of points, so ordering of points is not meaningful
  - Point Nets are in/equivariant to permutations
- Graphs can be defined by a *set* of nodes and a *set* of edges, so again order is not meaningful
  - Many graph nets represent graph as a *linear structure,* i.e. adjacency matrix which can be added / scaled
  - Layers are permutation equivariant and **linear** in the node features & adjacency matrix



Qi, C. R., Yi, L., Su, H., & Guibas, L. J. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. NIPS 2017
Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., & Smola, A. *Deep Sets*. NIPS 2017
Maron, H., Ben-Hamu, H., Shamir, N., & Lipman, Y. *Invariant and Equivariant Graph Networks*. ICLR 2018

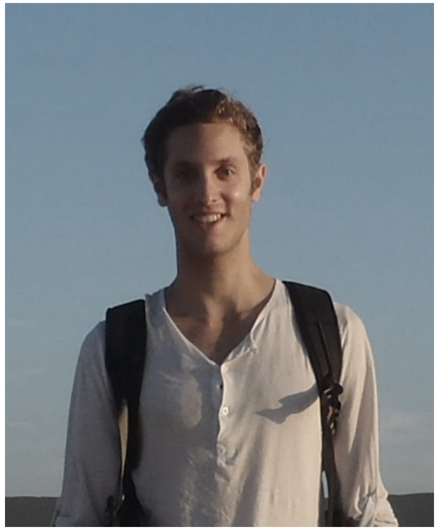
General Theories

- Key questions:
  - Classification of equivariant linear maps
  - Universal approximation theorems
- Homogeneous spaces:
  - Kondor, R., & Trivedi, S. *On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups*. ICML 2018
  - Cohen, T., Geiger, M., & Weiler, M. *A General Theory of Equivariant CNNs on Homogeneous Spaces.* NeurIPS 2019
  - Mackey, G. W. (1968). *Induced Representations of Groups and Quantum Mechanics*.
- General manifolds / Gauge CNNs:
  - Coming soon to an ArXiv near you
- Graphs, sets & other discrete structures
  - Maron, H., Fetaya, E., Segol, N., & Lipman, Y. On the Universality of Invariant Networks. ICML 2019
  - Segol, N., & Lipman, Y. (2019). On Universal Equivariant Set Networks. *ArXiv:1910.02421*.
  - Keriven, N., & Peyré, G. (2019). Universal Invariant and Equivariant Graph Neural Networks. NeurIPS 2019
  - Ravanbakhsh, S. (2020). Universal Equivariant Multilayer Perceptrons. *ArXiv:2002.02912*
  - Thiede, E. H., Hy, T. S., & Kondor, R. (2020). The general theory of permutation equivariant neural networks and higher order graph variational encoders. *ArXiv:2004.03990*.

Natural Graph Networks

Collaboration

Pim de Haan
Qualcomm AI Research
Qualcomm Technologies Netherlands B.V.
University of Amsterdam



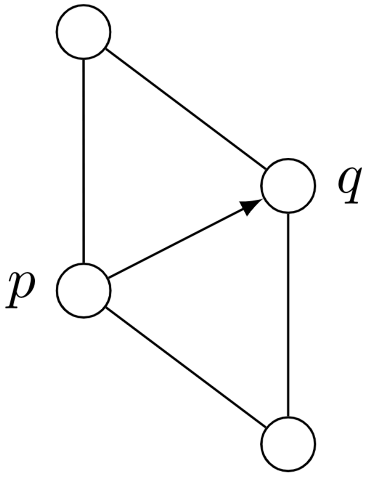
Taco Cohen
Qualcomm AI Research
Qualcomm Technologies Netherlands B.V.

Max Welling
Qualcomm AI Research
QUVA, University of Amsterdam
CIFAR
Qualcomm Technologies Netherlands B.V.

Graph Neural Networks

- Graphs are everywhere:
  - World wide web
  - Telecommunication networks
  - Social networks
  - Molecular graphs
  - Knowledge graphs
  - Road maps
  - Protein interaction networks
  - ….
- Fully-connected Neural networks are good at processing *vectors* (no symmetry)
- (G-)CNNs are good at processing spatial *signals* (geometrical symmetries)
- For graphs, we need graph networks that respect the relevant symmetries

Graph Convolutional Neural Networks

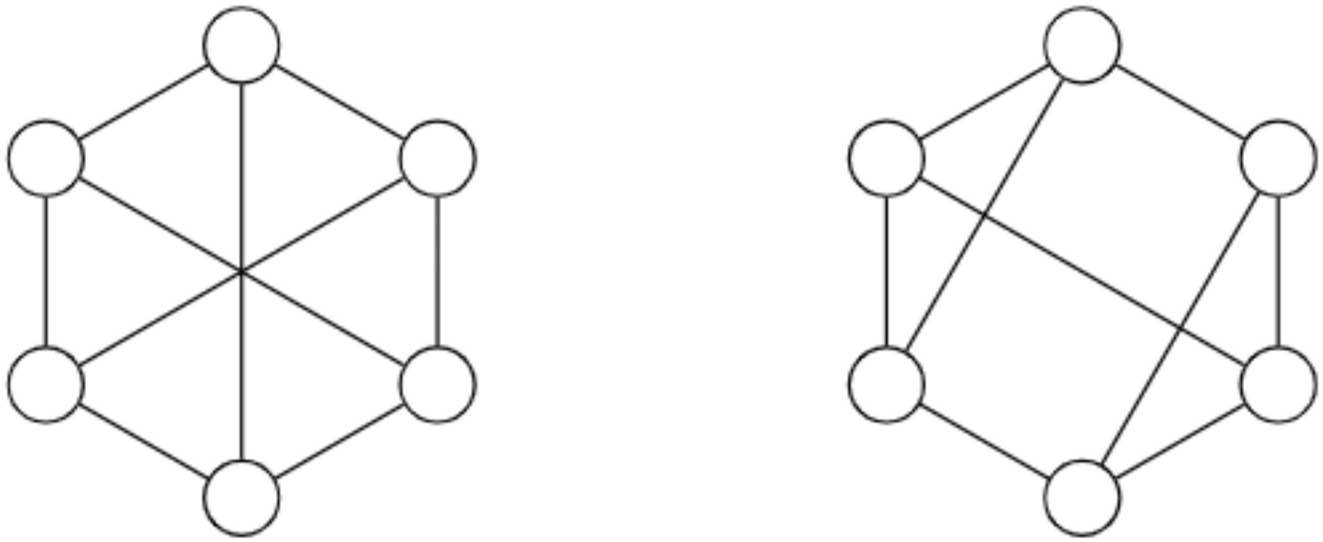


$$y_q = \sum_{(p,q)\in E} W x_p$$

- Pass messages to neighbours on the graph
- Linear function

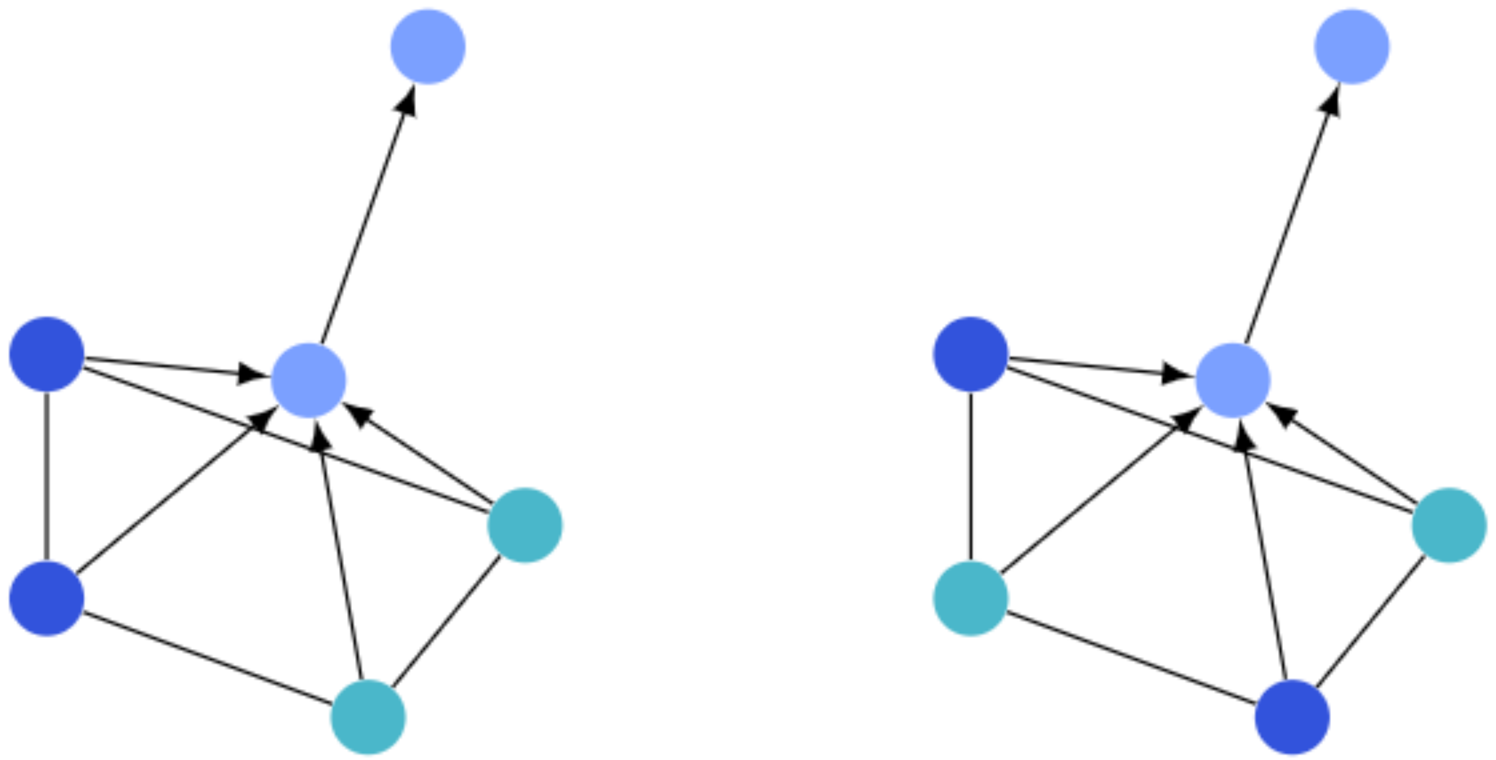
Limits of conventional Graph CNNs

Fail on regular graphs



Xu et al: How Powerful are Graph Neural Networks? (2018)

Source-aware features



- Detecting difference requires feature to remember where message came from

Conventional Graph CNNs

$$y_q = \sum_{pq \in E} W \, x_p$$

Same kernel on each edge

Invariant under permutation of neighbours

Kernel independent on graph

Kernel restricted by permutation group

Limited expressivity

Natural Graph Networks

$$y_q = \sum_{pq \in E} K_{pq}^{\mathcal{G}} \, x_p$$

Different kernel on different edges

Sensitive to permutations of neighbours

Kernel depends on graph

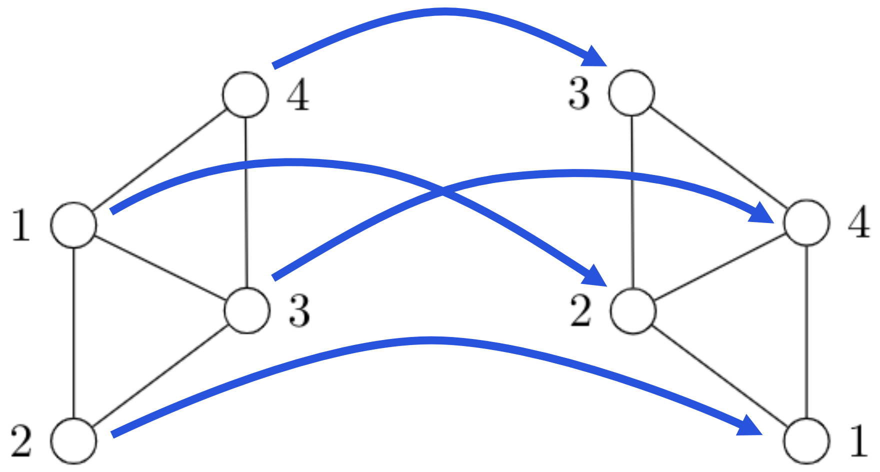
Kernel restricted by symmetry of graph

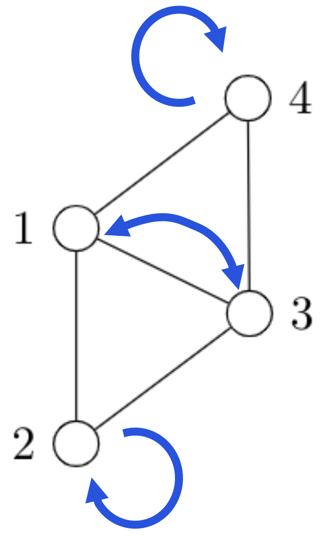
Most general convolution

Graph Equivalences & Symmetries

- Graph isomorphism

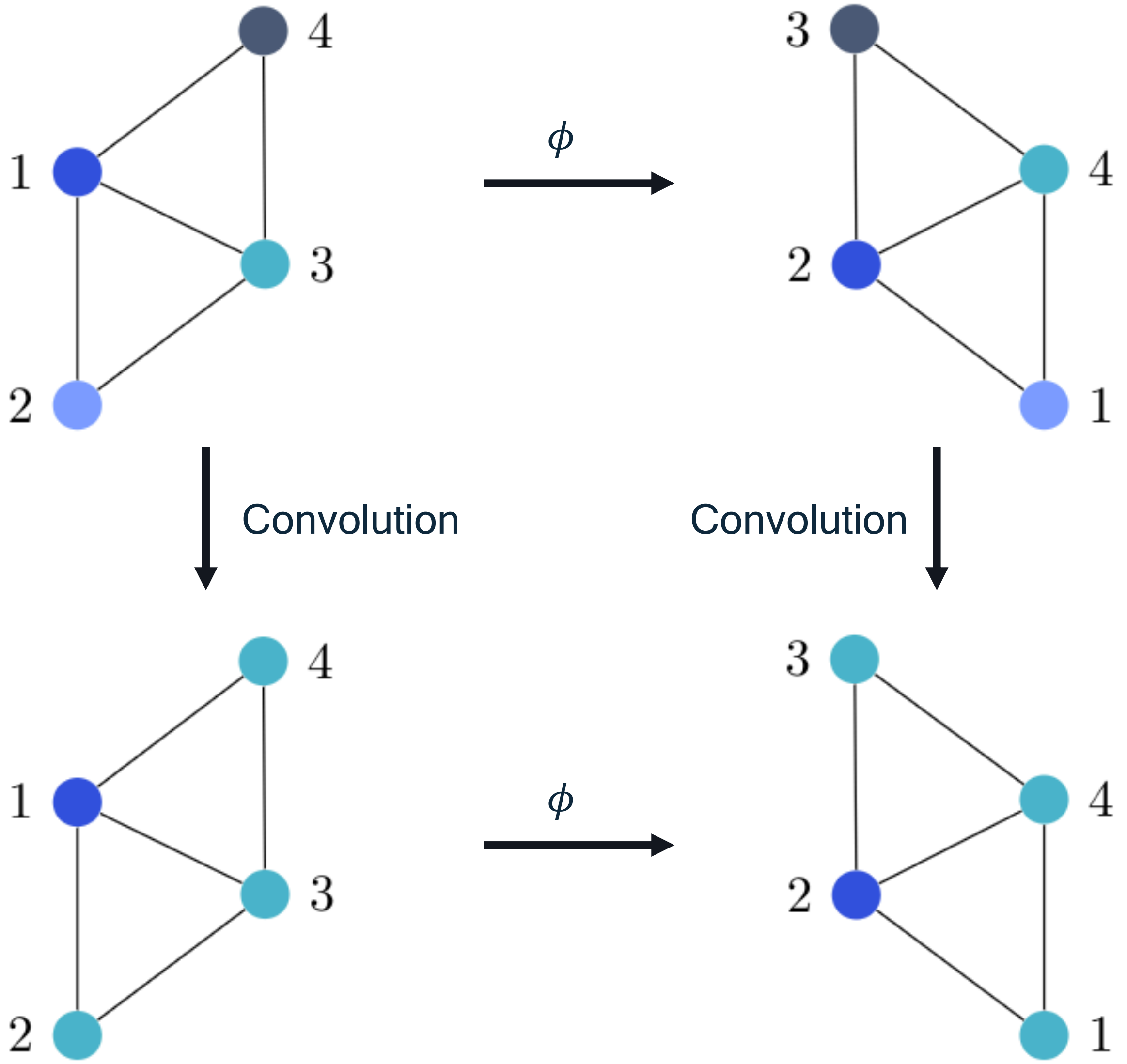
$$\phi : \mathcal{G} \to \mathcal{G}', \ \phi : V \xrightarrow{\sim} V' \text{ s.t. } (p,q) \in E \Leftrightarrow (\phi(p), \phi(q)) \in E'$$



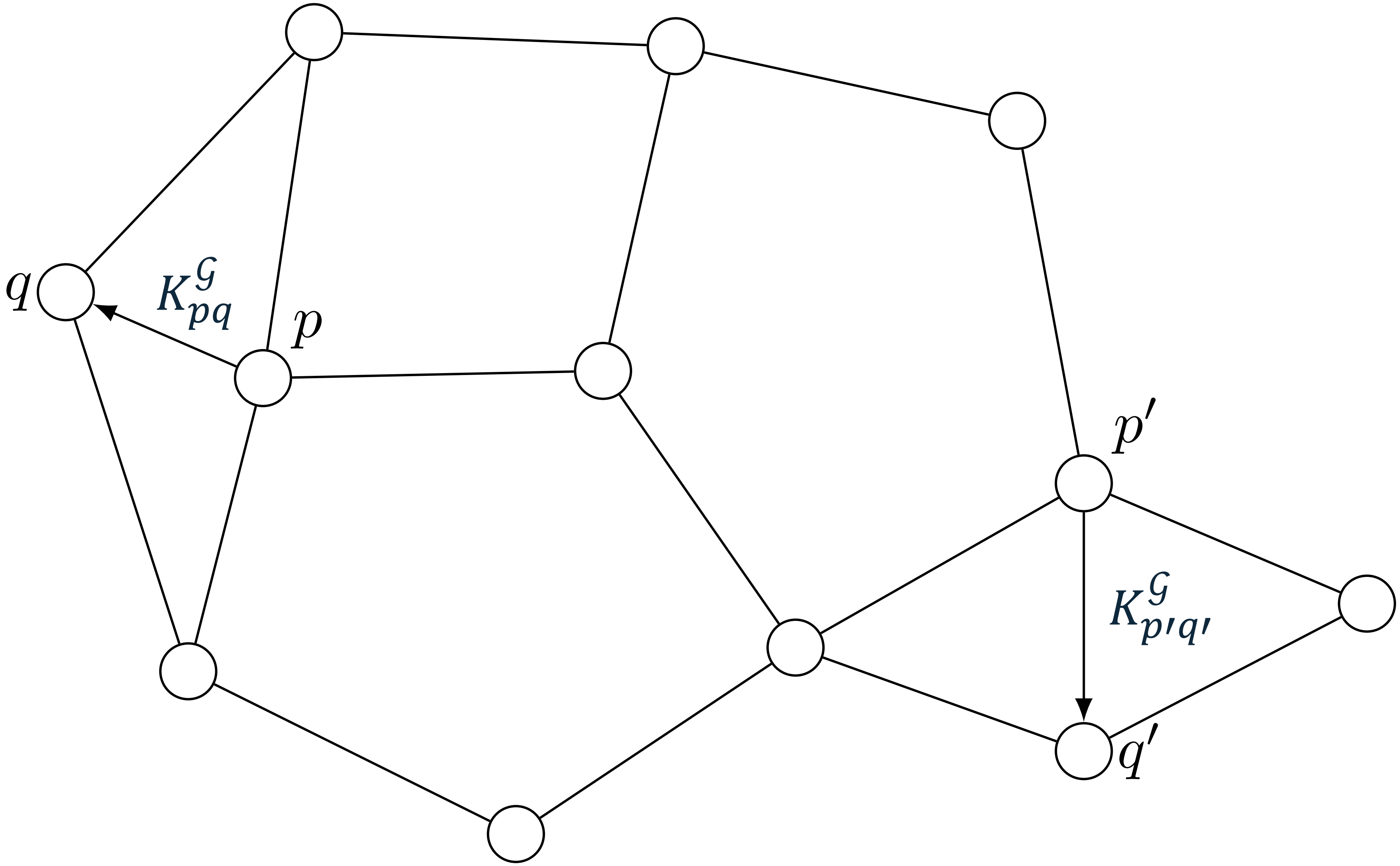


Automorphism = Symmetry

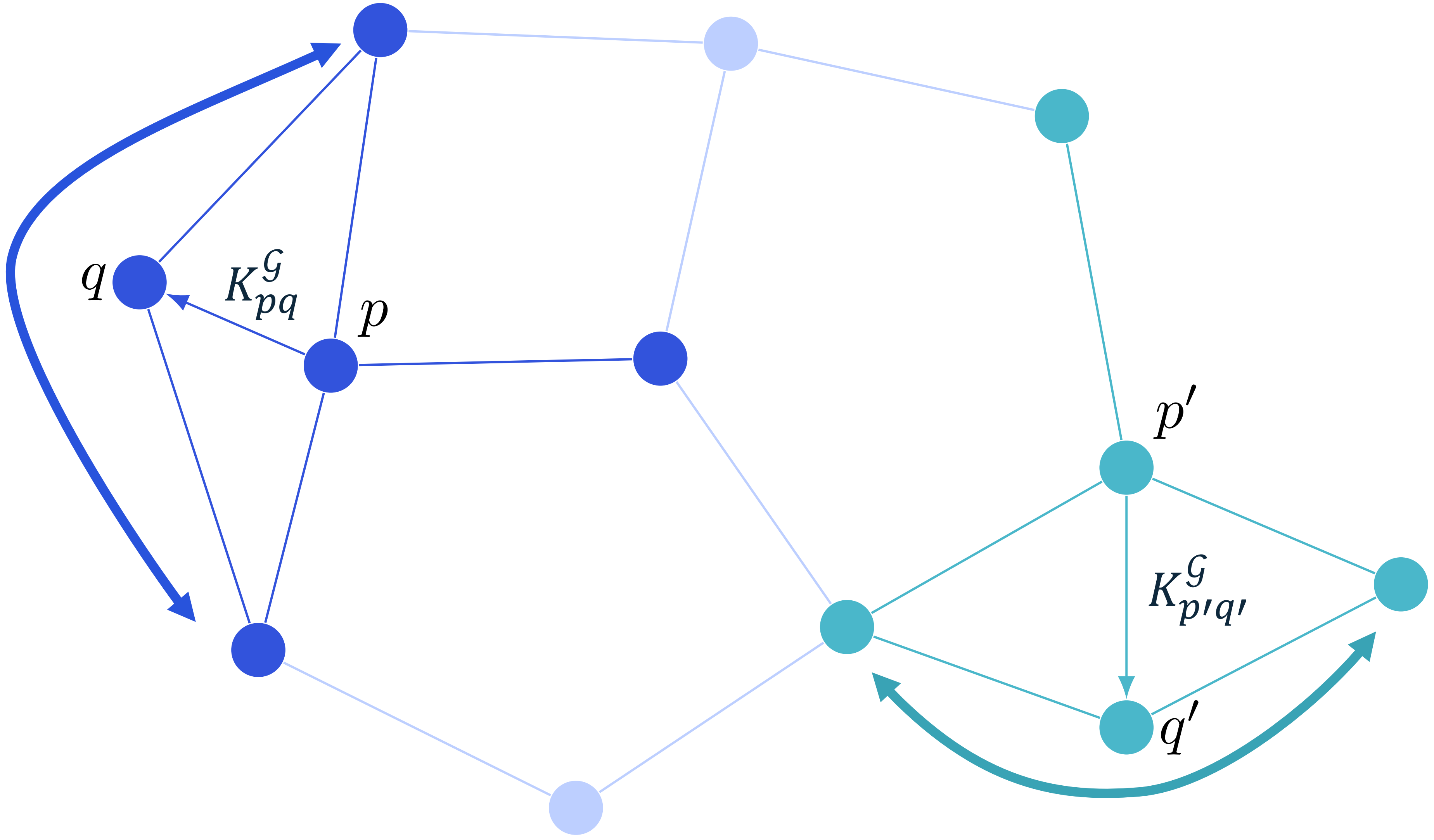
Equivariance



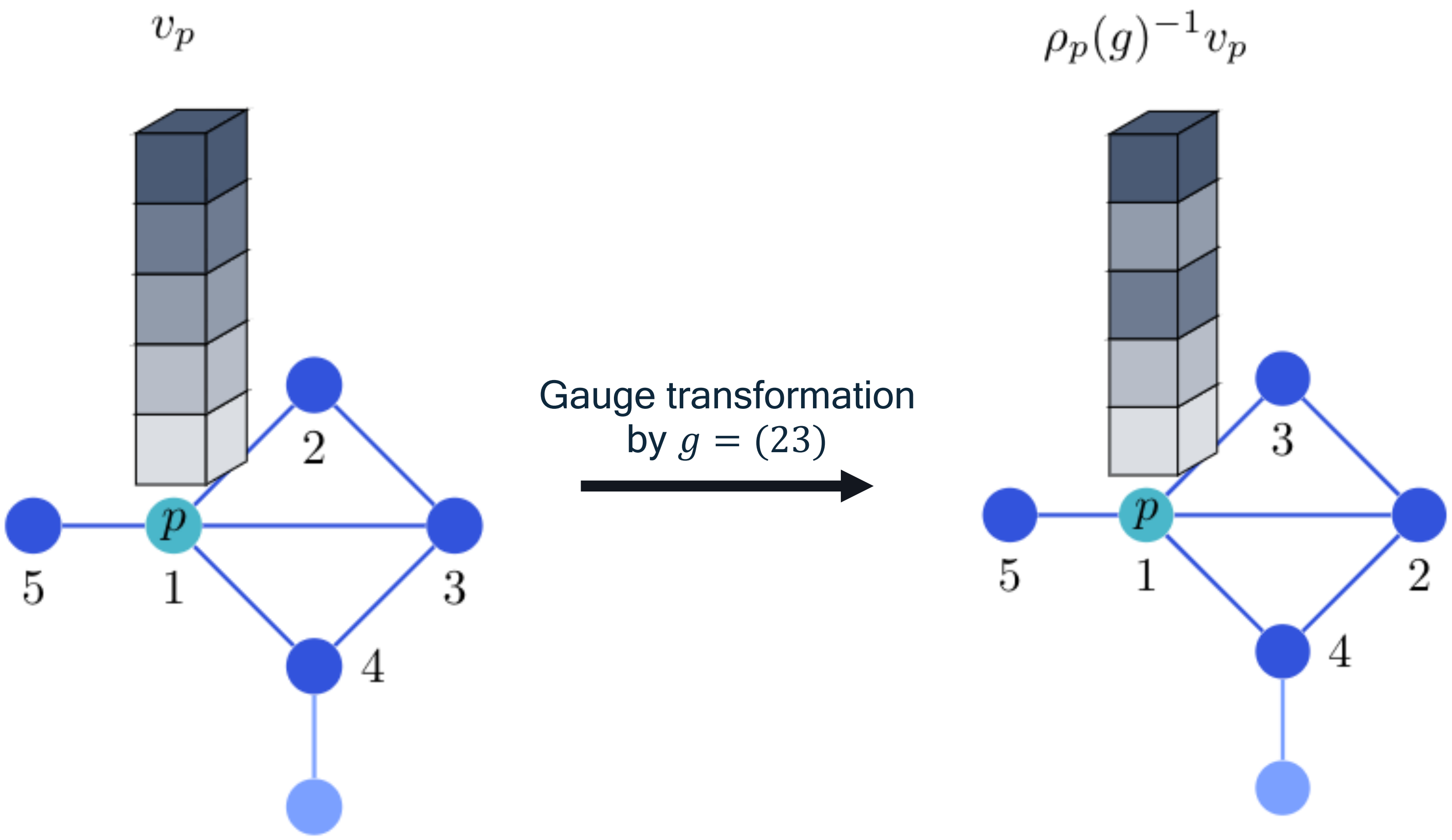
Equivariant Message Passing



Equivariant Message Passing with Local Symmetries



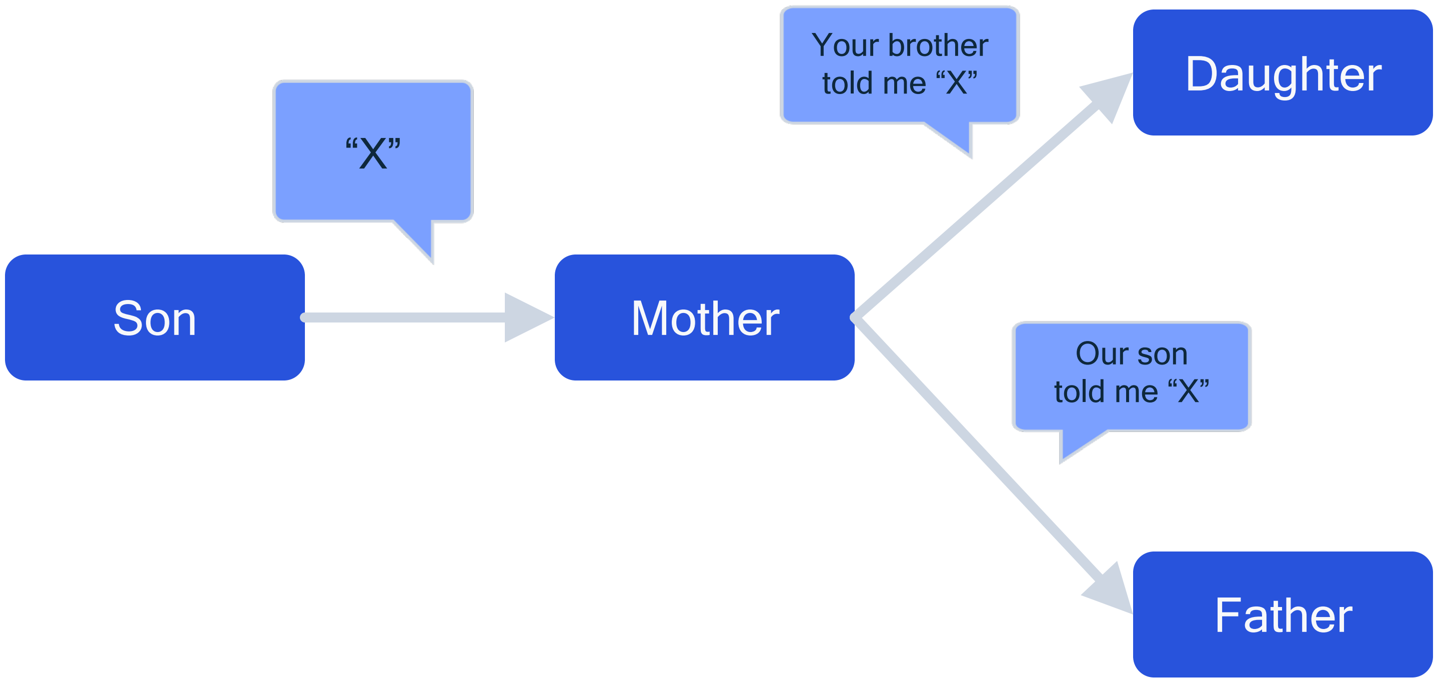
Feature space: neighbourhood gauges



Kondor et al: Covariant Compositional Networks For Learning Graphs (2018)

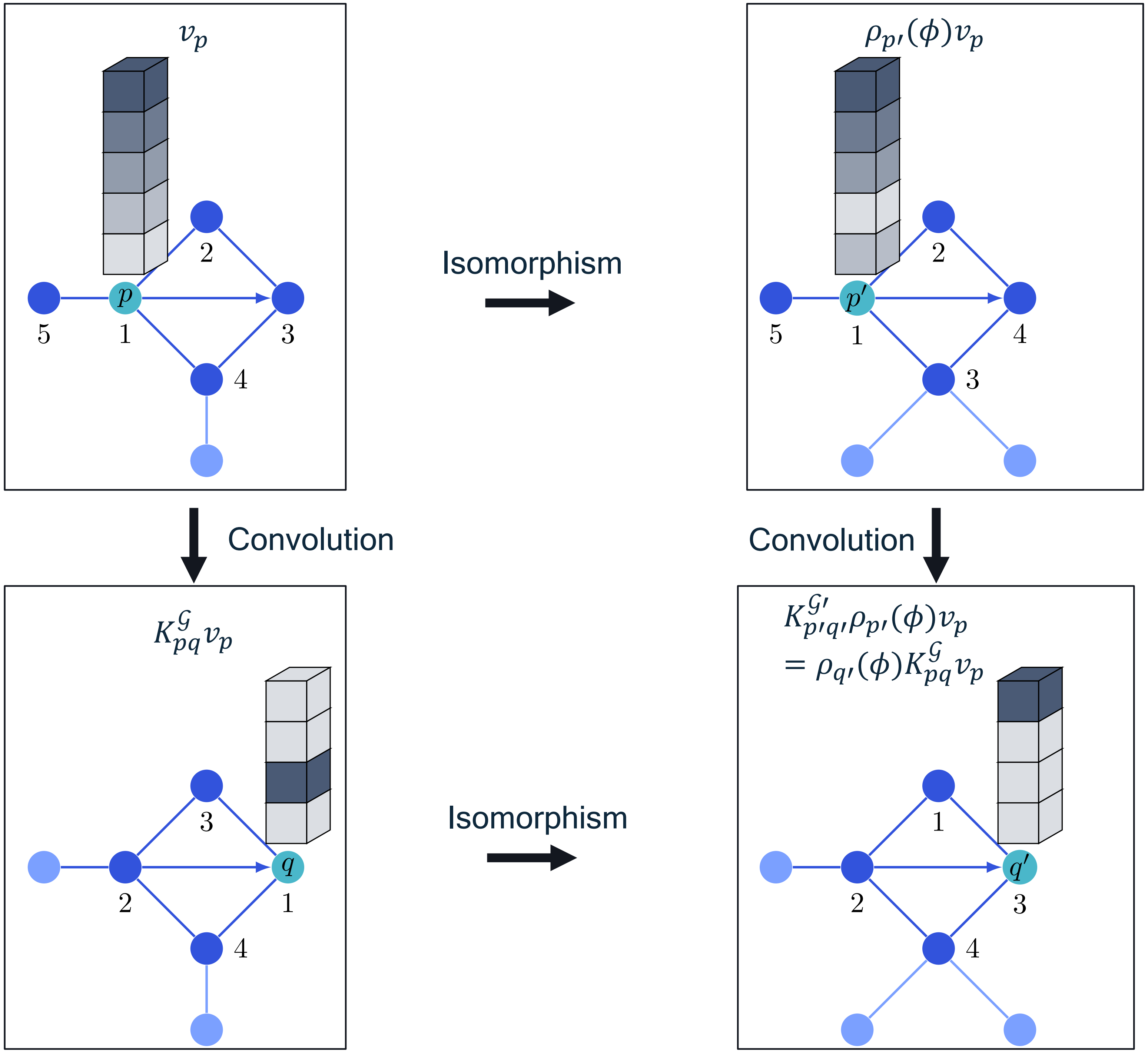
Cohen et al: Gauge Equivariant CNNs (2019)

Analogy



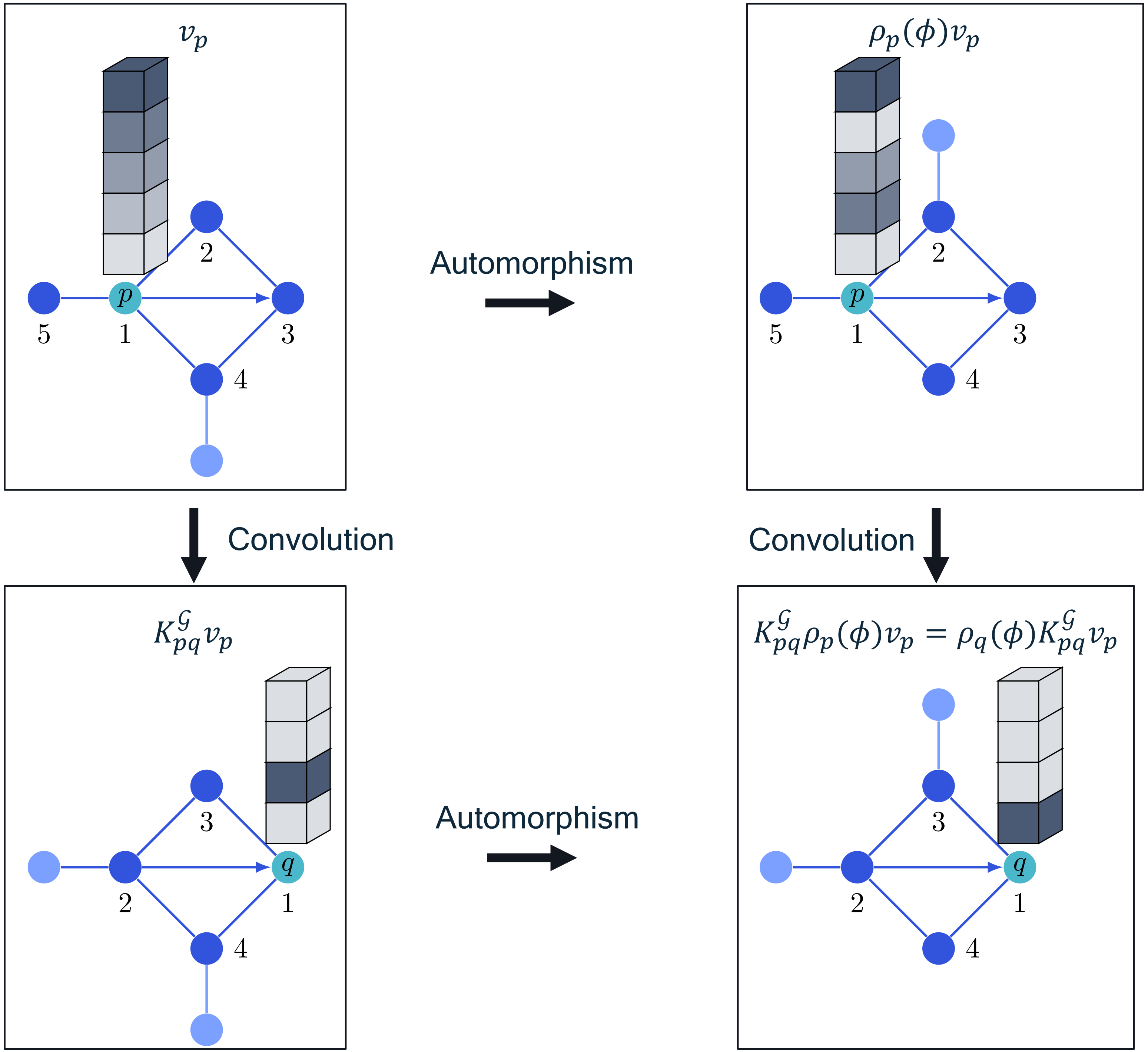
Son
"X"
Mother
Your brother told me "X"
Daughter
Our son told me "X"
Father

Isomorphism: Weight sharing



$$K^{\mathcal{G}'}_{p'q'} = \rho_{q'}(\phi)K^{\mathcal{G}}_{pq}\rho_{p'}(\phi)^{-1}$$

Automorphism: Kernel Constraint



$$K^{\mathcal{G}}_{pq} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & z \\ \cdot & x & \cdot & y & z \\ \cdot & y & \cdot & x & z \\ \cdot & \cdot & \cdot & \cdot & z \end{pmatrix}$$

Algorithm

- Precompute:

1. Define node and edge neighbourhoods
2. Classify edge neighbourhood isomorphism classes
3. Compute edge automorphisms
4. Solve kernel constraint, initialise params

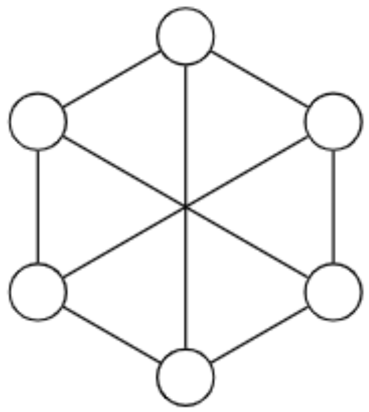
- During training:

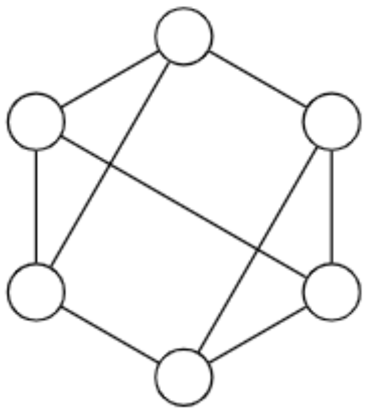
1. Linearly combine kernel solutions using parameters
2. Transport kernels by isomorphisms
3. Compute convolution

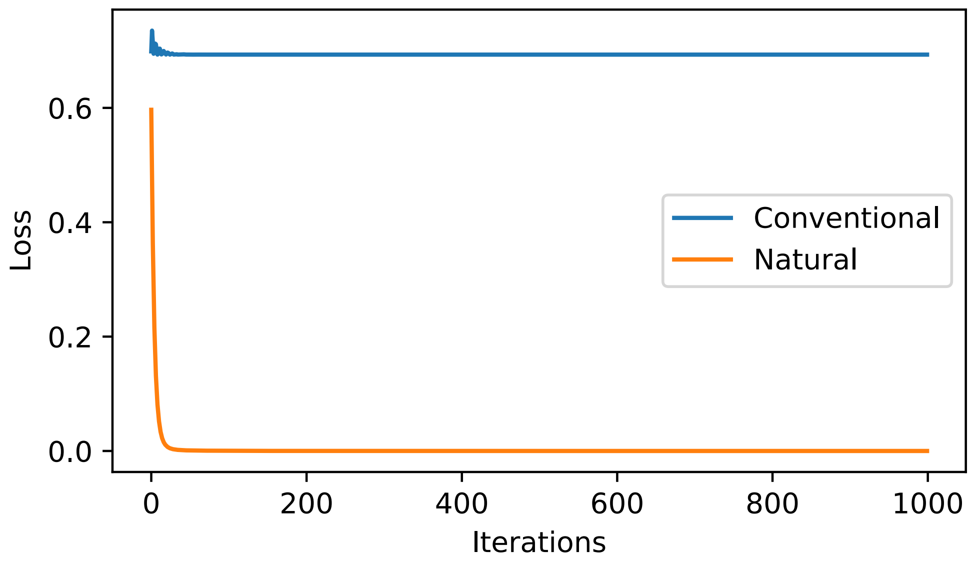
Relation to prior work

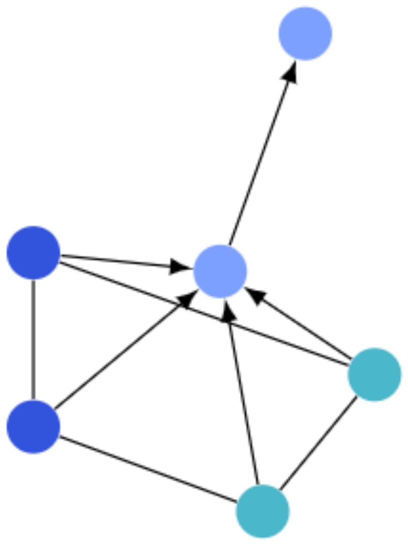
- Node neighbourhood trivial ⇒ graph CNN
- Rectangular grid / icosahedral graph ⇒ planar / icosahedral equivariant CNN
  - Cohen & Welling: Group Equivariant Convolutional Networks (2016)
  - Cohen et al: Gauge Equivariant Convolutional Networks and the Icosahedral CNN (2019)
- Kondor et al: Covariant Compositional Networks For Learning Graphs (2018)
  - Node neighbourhood size increases by depth
  - Kernel constrained by permutation group, instead of automorphism group
- Maron et al: Invariant and Equivariant Graph Networks (2019)
  - Represents entire graph as linear structure with permutation equivariance
  - Not message passing algorithm

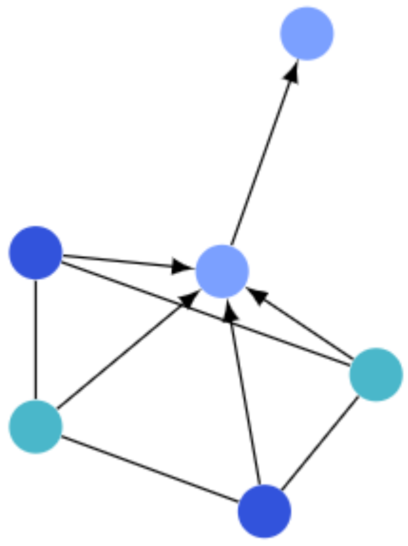
Synthetic experiments

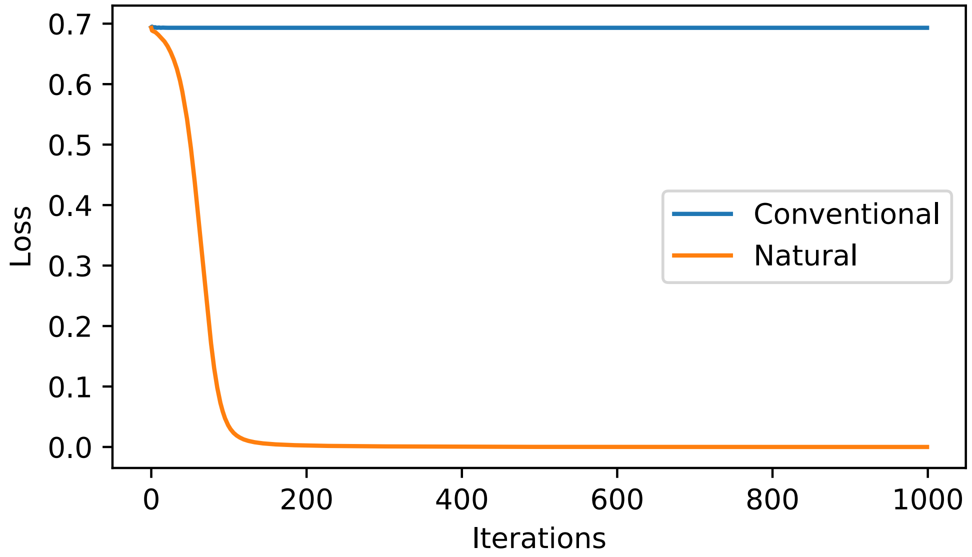












Preliminary Experiment: QM9 molecule predictions

	ENN-S2S Gilmer et al (2017)	CCN Kondor et al (2018)	IncidenceNet Albooyeh et al (2019)	Ours
CV	0.040	0.23	0.019	0.027
G	0.019	0.29	0.001	0.010
gap	0.069	0.54	0.073	0.07
H	0.017	0.30	0.001	0.012
LUMO	0.037	0.53	0.049	0.05
R2	0.180	0.19	0.010	0.040
U	0.019	0.29	0.001	0.012
U0	0.019	0.29	0.001	0.009
ZPVE	0.0015	0.39	0.006	0.0075
Average rank	2.3	4	1.4	2.2

Natural Graph Networks: Summary

- Graph networks must respect graph symmetries
- Graph symmetries = autmomorphisms ≠ permutation of nodes
- Exploting local symmetries leads to more powerful graph networks

Mathematical Theory

Category Theory: The Future of Deep Learning & AI

Category Theoretic Formulation

- General description of:
  - **Natural Graph Networks**
  - Homogeneous G-CNNs
  - Gauge CNNs

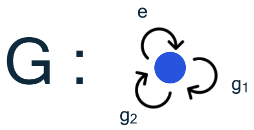
Category

- Objects Ob(C)
- Morphisms / arrows f : a -> b
- Associative composition rule
- Identity morphisms

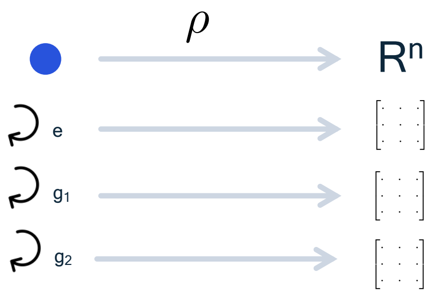
- Basic ingredients:
  - Category C of node neighbourhoods
    - Objects: “points with associated data”, arrows: “ways of transporting data between (some) points”
  - Category D of edge neighbourhoods
    - Objects: “messages”, arrows: “local symmetry / weight sharing”
  - Functors F0, F1 : D -> C, that map edge to source/target
    - Maps message to source/target, maps arrows in D to arrows in C
  - Principal groupoid P and category A of associated feature spaces on nodes
    - Functor T: C -> P (equivariant path lifting)
    - Functor R : P -> A (associated vector bundle functor; defines representation space)
  - Network layer is a natural transformation $K : R \circ T \circ F_0 \Rightarrow R \circ T \circ F_1$

Equivariance: it's only natural

- Group: a category with one object in which each morphism is an isomorphism



- Linear representation: a functor from group G to the category of vector spaces



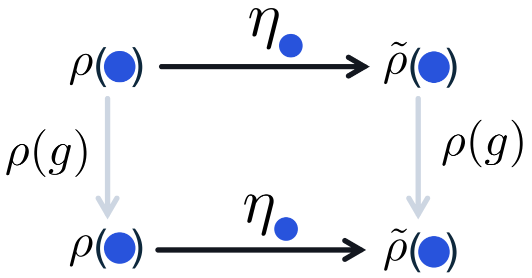
Associates to each object of G an object of Vec

Associates to each arrow of G an arrow of Vec, such that composition is preserved:

$$\rho(g)\rho(g') = \rho(gg')$$

- Equivariant linear map: natural transformation between functors (representations)

Associates to each object of G an arrow of Vec, such that for every morphism g in G, we have:



The category of node neighbourhoods

- Given graph G = (V, E)
- Define a category C:
  - Ob(C) = V
    - With each node, associate a neighbourhood (chosen in a consistent manner)
  - Introduce an arrow x -> y for each isomorphism from the neighbourhood of x to the neighbourhood of y that maps x to y.
    - “Rooted isomorphisms”
    - For given objects x, y, there could be n >= 0 such arrows.
- Note:
  - Arrows must be composable; in this case this is defined as composition of graph isomorphisms.
  - The category includes at least one arrow x -> x for each node x (the identity).
    - Additionally, when the neighbourhood has symmetries, it includes arrows x -> x for each neighbourhood automorphism
  - The category C is a groupoid, because all arrows are isomorphisms

The category of edge neighbourhoods

- Given graph G = (V, E)
- Define a category D:
  - Ob(D) = E
    - With each edge, associate a neighbourhood (chosen in a consistent manner)
  - Introduce an arrow e -> e’ for each isomorphism from the neighbourhood of e to the neighbourhood of e’.
    - For given e, e’, there could be n >= 0 such arrows.
- Note:
  - Arrows must be composable; in this case this is defined as composition of graph isomorphisms.
  - The category includes at least one arrow e -> e for each edge e (the identity).
    - Additionally, when the neighbourhood has symmetries, it includes arrows e -> e for each neighbourhood automorphism
  - The category D is a groupoid, because all arrows are isomorphisms

Source & Target Functors

- Define two functors $F_0, F_1 : D \to C$
- Functor maps both objects and arrows:
  - Let $e : x \to y$ and $e' : x' \to y'$ be isomorphic objects in D (i.e. edges with iso neighbourhoods)

	F_0	F_1
Map on objects	$F_0(e) = x$	$F_1(e) = y$
Map on arrows	$F_0(e \to e') = x \to x'$	$F_1(e \to e') = y \to y'$

- Check functor axioms:
  - Maps objects of D to objects of C (edge neighbourhoods to node neighbourhoods)
  - Maps morphisms of D to morphisms of C (edge isos tot node isos), such that:
    - F(f o g) = F(f) o F(g)
    - F(id_e) = id_F(e)
- Node neighbourhood should be a subset of edge neighbourhood, so that there is a natural definition of F_0, F_1 by restriction of graph isomorphism.

Principal and Associated Bundles

- Principal bundle groupoid P:
  - Ob(P) = (P_x, G_x) for x in ob(C). P_x contains all neighbourhood labelings. G_x permutes labels.
  - Morphisms are equivariant maps P_x -> $P_{x'}$ plus group homomorphisms G_x -> $G_{x'}$
- Transport functor T : C -> P
  - Lifts edges to equivariant maps
- Associated bundle A:
  - Associates with each node x in ob(C) a feature space V_x, acted on by a representation rho of G_x
- Representation functor R : P -> A

Network Layer: Natural Transformations

- Kernel is a natural transformation between functors:
  - Source feature space: $Q_0 = R \circ T \circ F_0$
  - Target feature space: $Q_1 = R \circ T \circ F_1$

  - Natural transformation: $K : Q_0 \Rightarrow Q_1$
- Definition of natural transformation:
  - K assigns to each object e = (x, y) of D a morphism (linear map) $K_e : Q_0(e) \to Q_1(e)$
  - Such that the following diagram commutes (naturality)

$$\begin{array}{ccc} Q_0(e) & \xrightarrow{K_e} & Q_1(e) \\ \downarrow{\scriptstyle Q_0\xi} & & \downarrow{\scriptstyle Q_1\xi} \\ Q_0(e') & \xrightarrow{K_{e'}} & Q_1(e') \end{array}$$

  - For all edge isomorphisms $\xi : e \to e'$

Application to Homogeneous & Gauge CNNs

- The same framework describes homogeneous G-CNNs and Gauge CNNs
  - Choice of C determines where the data lives
  - Choice of D determines how messages are passed (objects) and how weights are shared (morphisms)
- Homogeneous case:
  - C : points x in a manifold with morphisms x -> gx for g in G
  - D : pairs (x, y), with morphisms (x, y) -> (gx, gy)
  - R : induced representation functor
  - K : natural transformation = intertwiner between induced representation (= conv layer)
- Gauge CNN case:
  - C : points x in a manifold with morphisms x -> y paths
  - D : geodesics x -> y
  - R, T : associated vector bundle, parallel transport
  - K : natural transformation = gauge invariant linear map

Conclusions

- Equivariance is a natural design principle for neural networks
  - Applicable to planar images, signals on homogeneous spaces & manifolds, graphs, etc.
- New framework: natural graph networks
  - Fundamentally more flexible than invariant message passing methods
- Mathematical theory
  - Categorical formulation
  - Opens up a large design space for natural networks
  - Covers graphs, homogeneous spaces, general manifolds, and more in a uniform manner

Qualcomm

www: ta.co.nl

Twitter: @TacoCohen

Thank you

Follow us on:

For more information, visit us at:

www.qualcomm.com & www.qualcomm.com/blog