Equivariant Neural Networks: From Training Dynamics to Topology

Jan E. Gerken

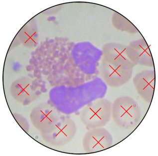
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

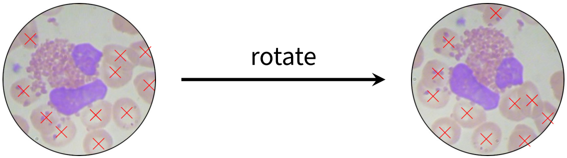
WASP
WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Symmetries in deep learning

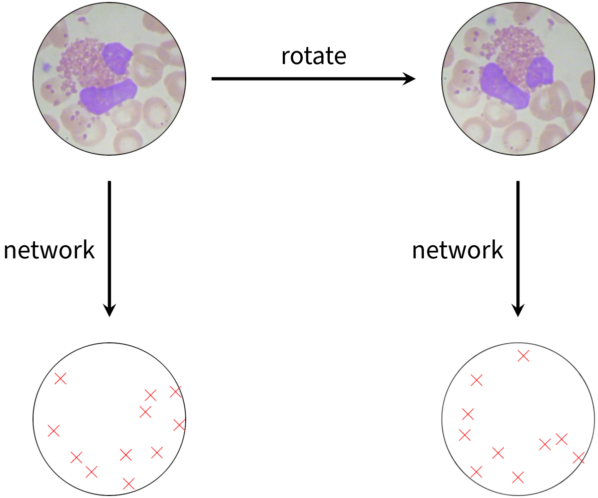
Symmetries in deep learning



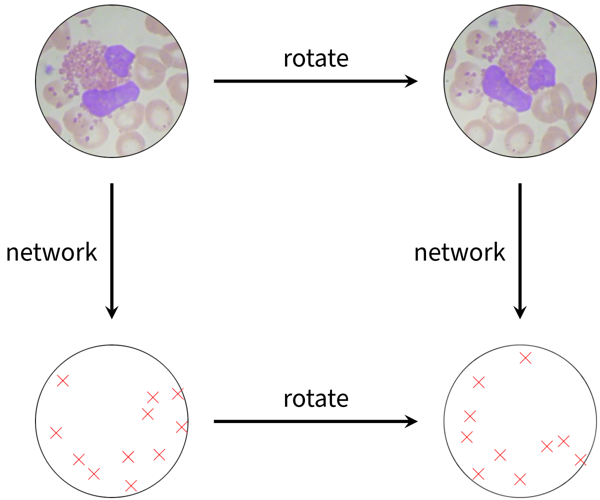
Symmetries in deep learning



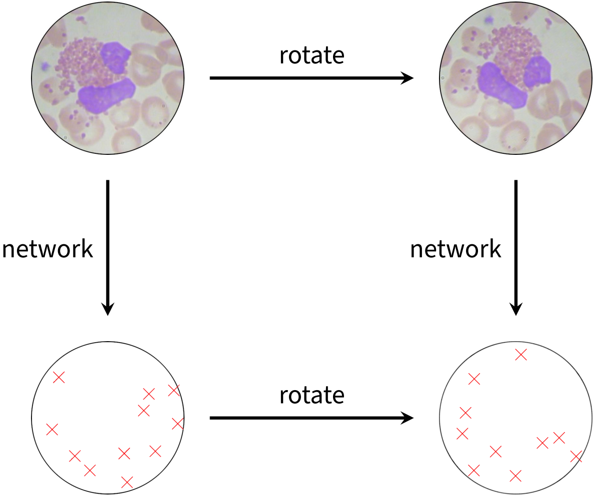
Symmetries in deep learning



Symmetries in deep learning

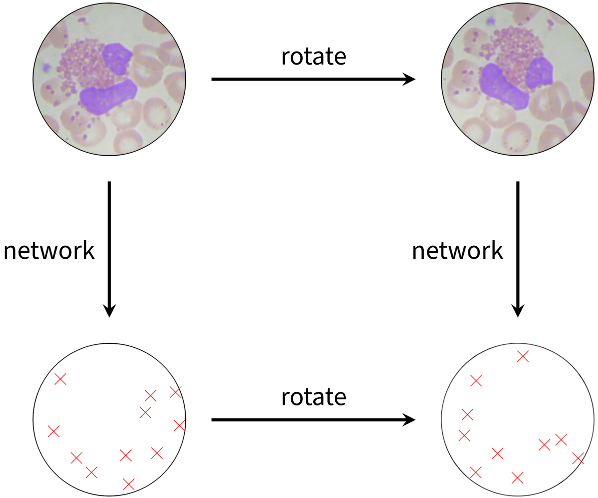


Symmetries in deep learning



$$\iff \mathcal{N}(\rho_{\text{in}}(g)x) = \rho_{\text{out}}(g)\mathcal{N}(x)$$

Symmetries in deep learning



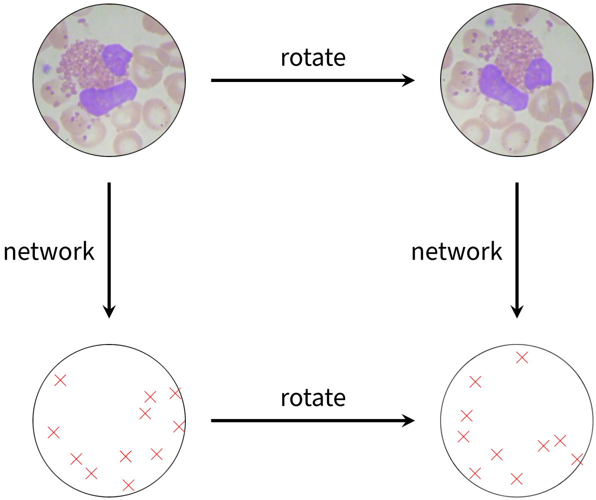
network

$$\Longleftrightarrow \quad \mathcal{N}(\rho_{\text{in}}(g)x) = \rho_{\text{out}}(g)\mathcal{N}(x)$$

rotation

image

Equivariance



network

$$\iff \mathcal{N}(\rho_{\text{in}}(g)x) = \rho_{\text{out}}(g)\mathcal{N}(x)$$

rotation

image

Equivariant neural networks

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam

Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

We introduce Group equivariant Convolutional Neural Networks (G-CNNs), a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symme-

Convolution layers can be used effectively in a *deep* network because all the layers in such a network are *translation equivariant*: shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects). In

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam

Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

We introduce Group equivariant Convolutional Neural Networks (G-CNNs), a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symme-

Convolution layers can be used effectively in a *deep* network because all the layers in such a network are *translation equivariant*: shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects). In

Equivariant Transformer Networks

Kai Sheng Tai [1] **Peter Bailis** [1] **Gregory Valiant** [1]

Abstract

How can prior knowledge on the transformation invariances of a domain be incorporated into the architecture of a neural network? We propose Equivariant Transformers (ETs), a family of differentiable image-to-image mappings that improve the robustness of models towards pre-defined continuous transformation groups. Through the use of specially-derived canonical coordinate systems, ETs incorporate functions that

scaling to each training image). While data augmentation typically helps reduce the test error of CNN-based models, there is no guarantee that transformation invariance will be enforced for data not seen during training.

In contrast to training time approaches like data augmentation, recent work on group equivariant CNNs (Cohen & Welling, 2016; Dieleman et al., 2016; Marcos et al., 2017; Worrall et al., 2017; Henriques & Vedaldi, 2017; Cohen et al., 2018) has explored new CNN architectures that are guaranteed to respond predictably to particular transforma-

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam
Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

We introduce Group equivariant Convolutional Neural Networks (G-CNNs), a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symme-

Convolution layers can be used effectively in a *deep* network because all the layers in such a network are *translation equivariant*: shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects). In

Equivariant Transformer Networks

Kai Sheng Tai [1] **Peter Bailis** [1] **Gregory Valiant** [1]

Abstract

How can prior knowledge on the transformation invariances of a domain be incorporated into the architecture of a neural network? We propose Equivariant Transformers (ETs), a family of differentiable image-to-image mappings that improve the robustness of models towards pre-defined continuous transformation groups. Through the use of specially-derived canonical co-

scaling to each training image). While data augmentation typically helps reduce the test error of CNN-based models, there is no guarantee that transformation invariance will be enforced for data not seen during training.

In contrast to training time approaches like data augmentation, recent work on group equivariant CNNs (Cohen & Welling, 2016; Dieleman et al., 2016; Marcos et al., 2017; Worrall et al., 2017; Henriques & Vedaldi, 2017; Cohen et al., 2018) has explored new CNN architectures that are

Theory for Equivariant Quantum Neural Networks

Quynh T. Nguyen,[1,2] Louis Schatzki,[3,4] Paolo Braccia,[1,5] Michael Ragone,[1,6] Patrick J. Coles,[1] Frédéric Sauvage,[1] Martín Larocca,[1,7] and M. Cerezo[3]
[1] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2] *School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*
[3] *Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[4] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[5] *Dipartimento di Fisica e Astronomia, Università di Firenze, Sesto Fiorentino (FI), 50019 , Italy*
[6] *Department of Mathematics, University of California Davis, Davis, California 95616, USA*
[7] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Quantum neural network architectures that have little-to-no inductive biases are known to face trainability and generalization issues. Inspired by a similar problem, recent breakthroughs in machine learning address this challenge by creating models encoding the symmetries of the learning task. This is materialized through the usage of equivariant neural networks whose action com-

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam

Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

We introduce Group equivariant Convolutional Neural Networks (G-CNNs), a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symme-

Convolution layers can be used effectively in a *deep* network because all the layers in such a network are *translation equivariant*: shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects). In

Equivariant Transformer Networks

Kai Sheng Tai [1] Peter Bailis [1] Gregory Valiant [1]

Abstract

How can prior knowledge on the transformation invariances of a domain be incorporated into the architecture of a neural network? We propose Equivariant Transformers (ETs), a family of differentiable image-to-image mappings that improve the robustness of models towards pre-defined continuous transformation groups. Through the use of specially-derived canonical coordinate systems, ETs incorporate functions that

scaling to each training image). While data augmentation typically helps reduce the test error of CNN-based models, there is no guarantee that transformation invariance will be enforced for data not seen during training.

In contrast to training time approaches like data augmentation, recent work on group equivariant CNNs (Cohen & Welling, 2016; Dieleman et al., 2016; Marcos et al., 2017; Worrall et al., 2017; Henriques & Vedaldi, 2017; Cohen et al., 2018) has explored new CNN architectures that are guaranteed to respond predictably to particular transforma-

Theory for Equivariant Quantum Neural Networks

Quynh T. Nguyen,[1,2] Louis Schatzki,[3,4] Paolo Braccia,[1,5] Michael Ragone,[1,6]
Patrick J. Coles,[1] Frédéric Sauvage,[1] Martín Larocca,[1,7] and M. Cerezo[3]

[1] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2] *School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*
[3] *Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[4] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[5] *Dipartimento di Fisica e Astronomia, Università di Firenze, Sesto Fiorentino (FI), 50019 , Italy*
[6] *Department of Mathematics, University of California Davis, Davis, California 95616, USA*
[7] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Quantum neural network architectures that have little-to-no inductive biases are known to face trainability and generalization issues. Inspired by a similar problem, recent breakthroughs in machine learning address this challenge by creating models encoding the symmetries of the learning task. This is materialized through the usage of equivariant neural networks whose action com-

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong[a,e,1] Qi Meng[b] Jue Zhang[b] Huilin Qu[c] Congqiao Li[d] Sitian Qian[d] Weitao Du[a] Zhi-Ming Ma[a] Tie-Yan Liu[b]

[a] *Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road, Beijing 100190, China*
[b] *Microsoft Research Asia, Danling Street, Beijing 100080, China*
[c] *CERN, EP Department, CH-1211 Geneva 23, Switzerland*
[d] *School of Physics, Peking University, Chengfu Road, Beijing 100871, China*

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam
Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

We introduce Group equivariant Convolutional Neural Networks (G-CNNs), a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symme-

Convolution layers can be used effectively in a *deep* network because all the layers in such a network are *translation equivariant*: shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects). In

Equivariant Transformer Networks

Kai Sheng Tai [1] Peter Bailis [1] Gregory Valiant [1]

Abstract

How can prior knowledge on the transformation invariances of a domain be incorporated into the architecture of a neural network? We propose Equivariant Transformers (ETs), a family of differentiable image-to-image mappings that improve the robustness of models towards pre-defined continuous transformation groups. Through the use of specially-derived canonical co-

scaling to each training image). While data augmentation typically helps reduce the test error of CNN-based models, there is no guarantee that transformation invariance will be enforced for data not seen during training.

In contrast to training time approaches like data augmentation, recent work on group equivariant CNNs (Cohen & Welling, 2016; Dieleman et al., 2016; Marcos et al., 2017; Worrall et al., 2017; Henriques & Vedaldi, 2017; Cohen et al., 2018) has explored new CNN architectures that are

Theory for Equivariant Quantum Neural Networks

Quynh T. Nguyen,[1,2] Louis Schatzki,[3,4] Paolo Braccia,[1,5] Michael Ragone,[1,6] Patrick J. Coles,[1] Frédéric Sauvage,[1] Martín Larocca,[1,7] and M. Cerezo[3]

[1] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2] *School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*
[3] *Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[4] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[5] *Dipartimento di Fisica e Astronomia, Università di Firenze, Sesto Fiorentino (FI), 50019 , Italy*
[6] *Department of Mathematics, University of California Davis, Davis, California 95616, USA*
[7] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Quantum neural network architectures that have little-to-no inductive biases are known to face trainability and generalization issues. Inspired by a similar problem, recent breakthroughs in machine learning address this challenge by creating models encoding the symmetries of the learning task. This is materialized through the usage of equivariant neural networks whose action com-

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong[a,e,1] **Qi Meng**[b] **Jue Zhang**[b] **Huilin Qu**[c] **Congqiao Li**[d] **Sitian Qian**[d] **Weitao Du**[a] **Zhi-Ming Ma**[a] **Tie-Yan Liu**[b]

[a] *Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road, Beijing 100190, China*
[b] *Microsoft Research Asia, Danling Street, Beijing 100080, China*
[c] *CERN, EP Department, CH-1211 Geneva 23, Switzerland*
[d] *School of Physics, Peking University, Chengfu Road, Beijing 100871, China*

E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials

Simon Batzner*,[1] Albert Musaelian,[1] Lixin Sun,[1] Mario Geiger,[2] Jonathan P. Mailoa,[3] Mordechai Kornbluth,[3] Nicola Molinari,[1] Tess E. Smidt,[4,5] and Boris Kozinsky*[1,3]

[1] *John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA*
[2] *École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland*
[3] *Robert Bosch Research and Technology Center, Cambridge, MA 02139, USA*
[4] *Computational Research Division and Center for Advanced Mathematics for Energy Research Applications, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[5] *Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02142, USA*

This work presents Neural Equivariant Interatomic Potentials (NequIP), an E(3)-equivariant neural network approach for learning interatomic potentials from *ab-initio* calculations for molecular dynamics simulations. While most contemporary symmetry-aware models use invariant convolutions and only act on scalars, NequIP employs E(3)-equivariant convolutions for interactions of geometric tensors, resulting in a more information-rich and faithful representation of atomic environments. The method achieves state-of-the-art accuracy on a challenging and diverse set of molecules and

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen T.S.COHEN@UVA.NL
University of Amsterdam

Max Welling M.WELLING@UVA.NL
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

Abstract

We introduce Group equivariant Convolutional Neural Networks (G-CNNs), a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symme-

Convolution layers can be used effectively in a *deep* network because all the layers in such a network are *translation equivariant*: shifting the image and then feeding it through a number of layers is the same as feeding the original image through the same layers and then shifting the resulting feature maps (at least up to edge-effects). In

Equivariant Transformer Networks

Kai Sheng Tai [1] **Peter Bailis** [1] **Gregory Valiant** [1]

Abstract

How can prior knowledge on the transformation invariances of a domain be incorporated into the architecture of a neural network? We propose Equivariant Transformers (ETs), a family of differentiable image-to-image mappings that improve the robustness of models towards pre-defined continuous transformation groups. Through the use of specially-derived canonical coordinate systems, ETs incorporate functions that

scaling to each training image). While data augmentation typically helps reduce the test error of CNN-based models, there is no guarantee that transformation invariance will be enforced for data not seen during training.

In contrast to training time approaches like data augmentation, recent work on group equivariant CNNs (Cohen & Welling, 2016; Dieleman et al., 2016; Marcos et al., 2017; Worrall et al., 2017; Henriques & Vedaldi, 2017; Cohen et al., 2018) has explored new CNN architectures that are guaranteed to respond predictably to particular transforma-

Theory for Equivariant Quantum Neural Networks

Quynh T. Nguyen,[1,2] Louis Schatzki,[3,4] Paolo Braccia,[1,5] Michael Ragone,[1,6]
Patrick J. Coles,[1] Frédéric Sauvage,[1] Martín Larocca,[1,7] and M. Cerezo[3]

[1] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2] *School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*
[3] *Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[4] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[5] *Dipartimento di Fisica e Astronomia, Università di Firenze, Sesto Fiorentino (FI), 50019, Italy*
[6] *Department of Mathematics, University of California Davis, Davis, California 95616, USA*
[7] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Quantum neural network architectures that have little-to-no inductive biases are known to face trainability and generalization issues. Inspired by a similar problem, recent breakthroughs in machine learning address this challenge by creating models encoding the symmetries of the learning task. This is materialized through the usage of equivariant neural networks whose action commutes with that of the symmetry. In this work, we import these ideas to the quantum realm by

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong[a,e,1] **Qi Meng**[b] **Jue Zhang**[b] **Huilin Qu**[c] **Congqiao Li**[d] **Sitian Qian**[d] **Weitao Du**[a] **Zhi-Ming Ma**[a] **Tie-Yan Liu**[b]

[a] *Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road, Beijing 100190, China*
[b] *Microsoft Research Asia, Danling Street, Beijing 100080, China*
[c] *CERN, EP Department, CH-1211 Geneva 23, Switzerland*
[d] *School of Physics, Peking University, Chengfu Road, Beijing 100871, China*

E(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials

Simon Batzner*,[1] Albert Musaelian,[1] Lixin Sun,[1] Mario Geiger,[2] Jonathan P. Mailoa,[3]
Mordechai Kornbluth,[3] Nicola Molinari,[1] Tess E. Smidt,[4,5] and Boris Kozinsky*[1,3]

[1] *John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA*
[2] *École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland*
[3] *Robert Bosch Research and Technology Center, Cambridge, MA 02139, USA*
[4] *Computational Research Division and Center for Advanced Mathematics for Energy Research Applications, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[5] *Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02142, USA*

This work presents Neural Equivariant Interatomic Potentials (NequIP), an E(3)-equivariant neural network approach for learning interatomic potentials from *ab-initio* calculations for molecular dynamics simulations. While most contemporary symmetry-aware models use invariant convolutions and only act on scalars, NequIP employs E(3)-equivariant convolutions for interactions of geometric tensors, resulting in a more information-rich and faithful representation of atomic environments. The method achieves state-of-the-art accuracy on a challenging and diverse set of molecules and

HIERARCHICAL, ROTATION-EQUIVARIANT NEURAL NETWORKS TO SELECT STRUCTURAL MODELS OF PROTEIN COMPLEXES

Stephan Eismann* Department of Applied Physics, Stanford University, seismann@stanford.edu

Raphael J.L. Townshend* Department of Computer Science, Stanford University, raphael@cs.stanford.edu

Nathaniel Thomas* Department of Physics, Stanford University, nthomas103@gmail.com

Milind Jagota Department of Electrical Engineering, Stanford University, mjagota@stanford.edu

Bowen Jing Department of Computer Science, Stanford University, bjing@stanford.edu

Ron O. Dror Department of Computer Science, Stanford University, rondror@cs.stanford.edu

ABSTRACT

Predicting the structure of multi-protein complexes is a grand challenge in biochemistry, with major implications for basic science and drug discovery. Computational structure prediction methods generally leverage pre-defined structural features to distinguish accurate structural models from less accurate ones. This raises the question of whether it is possible to learn characteristics of accurate models directly from atomic coordinates of protein complexes, with no prior assumptions. Here we introduce a machine learning method that learns directly from the 3D positions of all atoms to

Training Dynamics

✓ A lot of work on equivariant architectures

Training Dynamics

- ✓ A lot of work on equivariant architectures
- ? Training dynamics much less studied

Empirical NTK

Training dynamics under continuous gradient descent:

$$\frac{\mathrm{d}\mathcal{N}_\theta(x)}{\mathrm{d}t} = -\underbrace{\frac{\eta}{N}}_{\text{learning rate}} \sum_{i=1}^{N} \Theta_\theta(x, \underbrace{x_i}_{\text{training sample}}) \frac{\partial \overbrace{L}^{\text{loss}}}{\partial \mathcal{N}(x_i)}$$

Empirical NTK

Training dynamics under continuous gradient descent:

learning rate loss

$$\frac{\mathrm{d}\mathcal{N}_\theta(x)}{\mathrm{d}t} = -\frac{\eta}{N}\sum_{i=1}^{N} \Theta_\theta(x, x_i)\frac{\partial L}{\partial \mathcal{N}(x_i)}$$

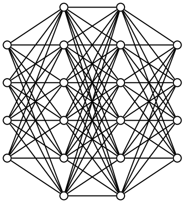
training sample

with the **empirical neural tangent kernel (NTK)**

$$\Theta_\theta(x, x') = \sum_\mu \frac{\partial \mathcal{N}(x)}{\partial \theta_\mu}\frac{\partial \mathcal{N}(x')}{\partial \theta_\mu}$$

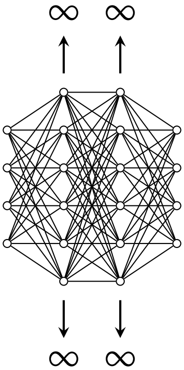
Infinite width limit

[Jacot et al. 2018]



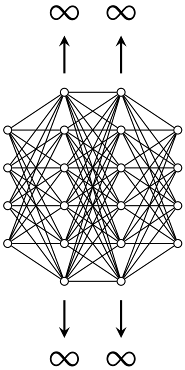
Infinite width limit

[Jacot et al. 2018]



Infinite width limit

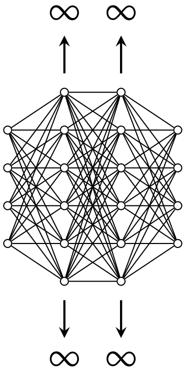
[Jacot et al. 2018]



👍 NTK becomes independent of initialization

Infinite width limit

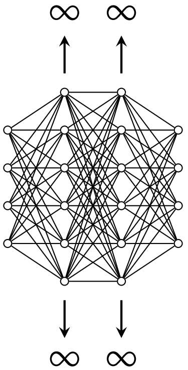
[Jacot et al. 2018]



- 👍 NTK becomes independent of initialization
- 👍 NTK becomes constant in training

Infinite width limit

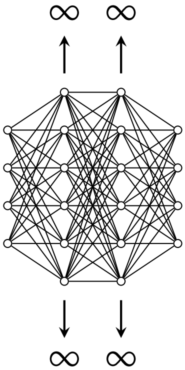
[Jacot et al. 2018]



- NTK becomes independent of initialization
- NTK becomes constant in training
- NTK can be computed for most networks

Infinite width limit

[Jacot et al. 2018]



- NTK becomes independent of initialization
- NTK becomes constant in training
- NTK can be computed for most networks
- ✓ Training dynamics can be solved

Mean prediction from NTK

[Jacot et al. 2018]

At infinite width, the mean prediction is given by

$$\mu_t(x) = \Theta(x, X)\Theta(X, X)^{-1}(\mathbb{I} - e^{-\eta\Theta(X,X)t})Y$$

Mean prediction from NTK

[Jacot et al. 2018]

ⓘ At infinite width, the mean prediction is given by

neural tangent kernel

$$\mu_t(x) = \Theta(x, X)\Theta(X, X)^{-1}(\mathbb{I} - e^{-\eta\Theta(X,X)t})Y$$

Mean prediction from NTK

[Jacot et al. 2018]

① At infinite width, the mean prediction is given by

neural tangent kernel

$$\mu_t(x) = \Theta(x, X)\Theta(X, X)^{-1}(\mathbb{I} - e^{-\eta\Theta(X,X)t})Y$$

train data

Mean prediction from NTK

[Jacot et al. 2018]

ⓘ At infinite width, the mean prediction is given by

neural tangent kernel

$$\mu_t(x) = \Theta(x, X)\Theta(X, X)^{-1}(\mathbb{I} - e^{-\eta\Theta(X,X)t})Y$$

learning rate

train data

Mean prediction from NTK

[Jacot et al. 2018]

ⓘ At infinite width, the mean prediction is given by

neural tangent kernel   train labels

$$\mu_t(x) = \Theta(x, X)\Theta(X, X)^{-1}\left(\mathbb{I} - e^{-\eta\Theta(X,X)t}\right)Y$$

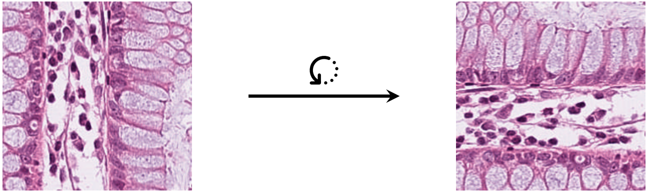
learning rate

train data

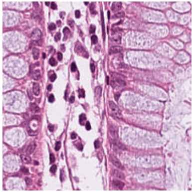
Equivariant Neural Tangent Kernels

Rotating images

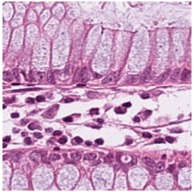
Rotating images



Rotating images



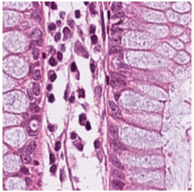
$$\xrightarrow{\circlearrowleft}$$

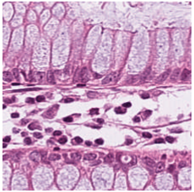


$f(x)$

f : pixels $\rightarrow$ colors

Rotating images





$$f(x)$$

f : pixels $\rightarrow$ colors

$$f(\rho(g^{-1})x)$$

$$= [\rho_{\text{reg}}(g)f](x)$$

Group convolutions

[Cohen, Welling 2016]

Group convolutions

[Cohen, Welling 2016]

Group conv's are the (unique) linear layers equivariant wrt ρ_{reg}

Group convolutions

[Cohen, Welling 2016]

Group conv's are the (unique) linear layers equivariant wrt ρ_{reg}

- Ordinary convolutions

$$f'(y) = \int_X \mathrm{d}x \, \kappa(x - y) \, f(x)$$

Group convolutions

[Cohen, Welling 2016]

Group conv's are the (unique) linear layers equivariant wrt ρ_{reg}

- Ordinary convolutions

$$f'(y) = \int_X \mathrm{d}x\, \kappa(x - y)\, f(x)$$

- Group convolutions

$$f'(g) = \int_X \mathrm{d}x\, \kappa(\rho(g^{-1})x)\, f(x) \qquad \text{lifting}$$

Group convolutions

[Cohen, Welling 2016]

Group conv's are the (unique) linear layers equivariant wrt ρ_{reg}

- Ordinary convolutions

$$f'(y) = \int_X \mathrm{d}x\, \kappa(x - y)\, f(x)$$

- Group convolutions

$$f'(g) = \int_X \mathrm{d}x\, \kappa(\rho(g^{-1})x)\, f(x) \qquad \text{lifting}$$

$$f'(g) = \int_G \mathrm{d}g\, \kappa(g^{-1}h)\, f(h) \qquad \text{group convolution}$$

Group convolutions

[Cohen, Welling 2016]

Group conv's are the (unique) linear layers equivariant wrt ρ_{reg}

- Ordinary convolutions

$$f'(y) = \int_X \mathrm{d}x\, \kappa(x - y)\, f(x)$$

- Group convolutions

$$f'(g) = \int_X \mathrm{d}x\, \kappa(\rho(g^{-1})x)\, f(x) \qquad \text{lifting}$$

$$f'(g) = \int_G \mathrm{d}g\, \kappa(g^{-1}h)\, f(h) \qquad \text{group convolution}$$

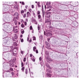
$$f' = \frac{1}{\mathrm{vol}(G)} \int_G \mathrm{d}g\, f(g) \qquad \text{group pooling}$$

GCNNs

Stack GConv-layers to obtain an invariant network

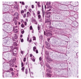
GCNNs

Stack GConv-layers to obtain an invariant network



GCNNs

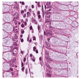
Stack GConv-layers to obtain an invariant network



lifting

GCNNs

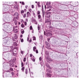
Stack GConv-layers to obtain an invariant network



$\longrightarrow$ lifting $\longrightarrow$ σ

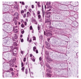
GCNNs

Stack GConv-layers to obtain an invariant network

 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv

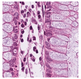
GCNNs

Stack GConv-layers to obtain an invariant network

 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ

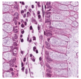
GCNNs

Stack GConv-layers to obtain an invariant network

 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv

GCNNs

Stack GConv-layers to obtain an invariant network

 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

NTKs for GCNNs

For GCNN-layers, define the NNGP and NTK via

$$K^{(\ell)}_{g,g'}(f,f') = \mathbb{E}\left[[\mathcal{N}^{(\ell)}(f)](g)\left([\mathcal{N}^{(\ell)}(f')](g')\right)^\top\right]$$

NTKs for GCNNs

For GCNN-layers, define the NNGP and NTK via

$$K^{(\ell)}_{g,g'}(f,f') = \mathbb{E}\left[[\mathcal{N}^{(\ell)}(f)](g)\left([\mathcal{N}^{(\ell)}(f')](g')\right)^\top\right]$$

$$\Theta^{(\ell)}_{g,g'}(f,f') = \mathbb{E}\left[\sum_{\ell'=1}^{\ell} \frac{\partial[\mathcal{N}^{(\ell)}(f)](g)}{\partial\theta^{(\ell')}}\left(\frac{\partial[\mathcal{N}^{(\ell)}(f')](g')}{\partial\theta^{(\ell')}}\right)^\top\right]$$

NTKs for GCNNs

$$[\mathcal{N}^{(\ell)}(f)](g) = \int_G \mathrm{d}g\, \kappa(g^{-1}h)\, [\mathcal{N}^{(\ell-1)}(f)](h)$$

The layer-recursion for a GCNN-layer is given by

$$K^{(\ell+1)}_{g,g'}(f,f') = \frac{1}{|S_\kappa|} \int_{S_\kappa} \mathrm{d}h\, K^{(\ell)}_{gh,g'h}(f,f')$$

NTKs for GCNNs

$$\boxed{[\mathcal{N}^{(\ell)}(f)](g) = \int_G \mathrm{d}g\, \kappa(g^{-1}h)\, [\mathcal{N}^{(\ell-1)}(f)](h)}$$

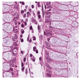
The layer-recursion for a GCNN-layer is given by

$$K^{(\ell+1)}_{g,g'}(f,f') = \frac{1}{|S_\kappa|} \int_{S_\kappa} \mathrm{d}h\, K^{(\ell)}_{gh,g'h}(f,f')$$

$$\Theta^{(\ell+1)}_{g,g'}(f,f') = K^{(\ell+1)}_{g,g'}(f,f') + \frac{1}{|S_\kappa|} \int_{S_\kappa} \mathrm{d}h\, \Theta^{(\ell)}_{gh,g'h}(f,f')$$

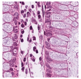
GCNNs

Stack GConv-layers to obtain an invariant network

 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

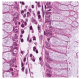


$\longrightarrow$ lifting $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

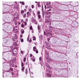
 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

0

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

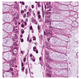
 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

$$0 \longrightarrow \Theta^{(1)}_{g,g'}(f,f')$$

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

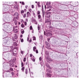
 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

$$0 \longrightarrow \Theta^{(1)}_{g,g'}(f,f') \longrightarrow \Theta^{(2)}_{g,g'}(f,f')$$

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network



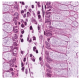
$$\longrightarrow \text{lifting} \longrightarrow \sigma \longrightarrow \text{GConv} \longrightarrow \sigma \longrightarrow \text{GConv} \longrightarrow \text{GPool}$$

Compute NTK with layer-wise recursion

$$0 \longrightarrow \Theta^{(1)}_{g,g'}(f,f') \longrightarrow \Theta^{(2)}_{g,g'}(f,f') \longrightarrow \Theta^{(3)}_{g,g'}(f,f')$$

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network



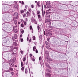
$$\longrightarrow \text{lifting} \longrightarrow \sigma \longrightarrow \text{GConv} \longrightarrow \sigma \longrightarrow \text{GConv} \longrightarrow \text{GPool}$$

Compute NTK with layer-wise recursion

$$0 \longrightarrow \Theta^{(1)}_{g,g'}(f,f') \longrightarrow \Theta^{(2)}_{g,g'}(f,f') \longrightarrow \Theta^{(3)}_{g,g'}(f,f') \longrightarrow \Theta^{(4)}_{g,g'}(f,f')$$

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

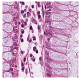
 $\longrightarrow$ lifting $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

$$0 \longrightarrow \Theta^{(1)}_{g,g'}(f,f') \longrightarrow \Theta^{(2)}_{g,g'}(f,f') \longrightarrow \Theta^{(3)}_{g,g'}(f,f') \longrightarrow \Theta^{(4)}_{g,g'}(f,f') \longrightarrow \Theta^{(5)}_{g,g'}(f,f')$$

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

 $\longrightarrow$ lifting $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

$$0 \longrightarrow \Theta^{(1)}_{g,g'}(f,f') \longrightarrow \Theta^{(2)}_{g,g'}(f,f') \longrightarrow \Theta^{(3)}_{g,g'}(f,f') \longrightarrow \Theta^{(4)}_{g,g'}(f,f') \longrightarrow \Theta^{(5)}_{g,g'}(f,f') \longrightarrow \Theta(f,f')$$

NTKs of MLPs and GCNNs

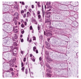
NTKs of MLPs and GCNNs

- Consider two neural networks

NTKs of MLPs and GCNNs

- Consider two neural networks

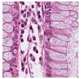
An MLP

 $\longrightarrow \text{FC} \longrightarrow \sigma \longrightarrow \text{FC} \longrightarrow \sigma \longrightarrow \text{FC}$

NTKs of MLPs and GCNNs

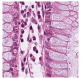
- Consider two neural networks

An MLP



$\longrightarrow \text{FC} \longrightarrow \sigma \longrightarrow \text{FC} \longrightarrow \sigma \longrightarrow \text{FC}$

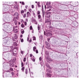
A GCNN



NTKs of MLPs and GCNNs

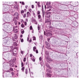
- Consider two neural networks

An MLP



$\longrightarrow \mathrm{FC} \longrightarrow \sigma \longrightarrow \mathrm{FC} \longrightarrow \sigma \longrightarrow \mathrm{FC}$

A GCNN

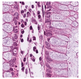


$\longrightarrow$ lifting

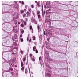
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN

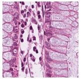


$$\begin{tikzcd}
{} \arrow[r] & \mathrm{FC} \arrow[r] \arrow[dd] & \sigma \arrow[r] \arrow[dd] & \mathrm{FC} \arrow[r] & \sigma \arrow[r] & \mathrm{FC} \\
 & & & & & \\
{} \arrow[r] & \text{lifting} \arrow[r] & \sigma & & &
\end{tikzcd}$$

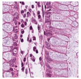
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



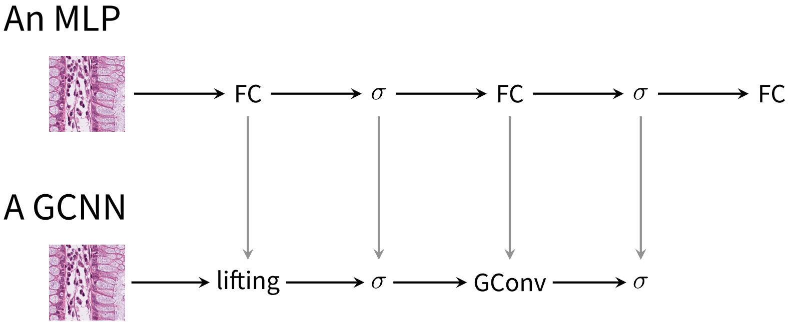
A GCNN



$$\begin{tikzcd}
\cdot \arrow[r] & \mathrm{FC} \arrow[r] \arrow[d] & \sigma \arrow[r] \arrow[d] & \mathrm{FC} \arrow[r] \arrow[d] & \sigma \arrow[r] & \mathrm{FC} \\
\cdot \arrow[r] & \text{lifting} \arrow[r] & \sigma \arrow[r] & \mathrm{GConv} & &
\end{tikzcd}$$

NTKs of MLPs and GCNNs

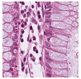
- Consider two neural networks



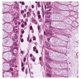
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN

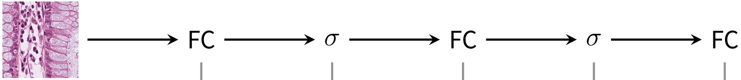


$$
\begin{tikzcd}
\cdot \arrow[r] & \mathrm{FC} \arrow[r] \arrow[d] & \sigma \arrow[r] \arrow[d] & \mathrm{FC} \arrow[r] \arrow[d] & \sigma \arrow[r] \arrow[d] & \mathrm{FC} \arrow[d] \\
\cdot \arrow[r] & \text{lifting} \arrow[r] & \sigma \arrow[r] & \mathrm{GConv} \arrow[r] & \sigma \arrow[r] & \mathrm{GConv}
\end{tikzcd}
$$

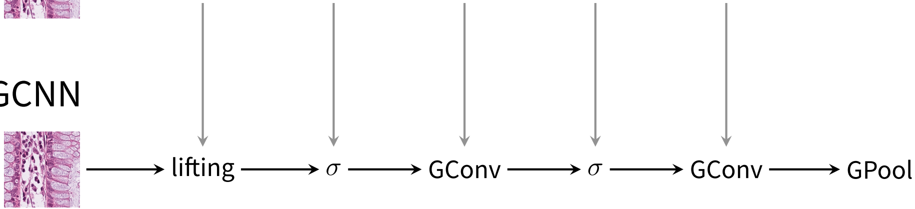
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN

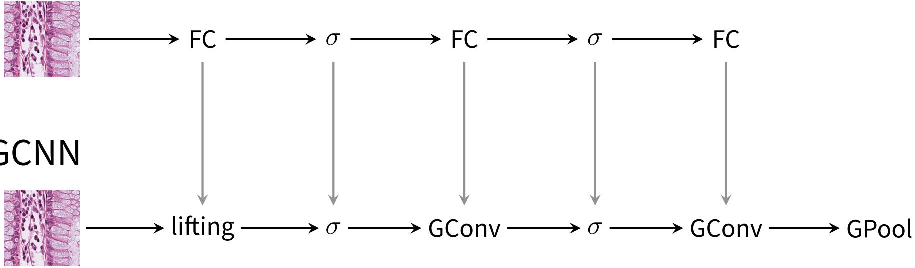


NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP

A GCNN



- Then

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

Data augmentation and NTKs

Data augmentation and NTKs

trained without data augmentation

trained with data augmentation

If

$$\Theta^{\text{non-aug}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\text{aug}}(f, \rho_{\text{reg}}(g) f')$$

Data augmentation and NTKs

trained without data augmentation

trained with data augmentation

If

$$\Theta^{\mathrm{non-aug}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{aug}}(f, \rho_{\mathrm{reg}}(g) f')$$

Then

$$\mu_t^{\mathrm{non-aug}}(x) = \mu_t^{\mathrm{aug}}(x)$$

at infinite width. ⇨Predictions of infinite ensembles agree.

Data augmentation and NTKs

trained without data augmentation

trained with data augmentation

If

$$\Theta^{\text{non-aug}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\text{aug}}(f, \rho_{\text{reg}}(g) f')$$

Then

$$\mu_t^{\text{non-aug}}(x) = \mu_t^{\text{aug}}(x) \quad \forall t$$

at infinite width. ⇨ Predictions of infinite ensembles agree.

Data augmentation and NTKs

trained without data augmentation

trained with data augmentation

If

$$\Theta^{\text{non-aug}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\text{aug}}(f, \rho_{\text{reg}}(g) f')$$

Then

$$\mu_t^{\text{non-aug}}(x) = \mu_t^{\text{aug}}(x) \quad \forall t \quad \forall x$$

at infinite width. ⇨Predictions of infinite ensembles agree.

Data augmentation of MLPs

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

Data augmentation of MLPs

before: non-aug

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

Data augmentation of MLPs

before: non-aug

before: aug

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

Data augmentation of MLPs

before: non-aug

before: aug

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

⇨ training the MLP on G-augmented data

Data augmentation of MLPs

before: non-aug

before: aug

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

⇨ training the MLP on G-augmented data = training the GCNN on unaugmented data

Data augmentation of MLPs

before: non-aug

before: aug

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\mathrm{MLP}}(f, \rho_{\mathrm{reg}}(g) f')$$

⇨ training the MLP on G-augmented data = training the GCNN on unaugmented data

in the ensemble mean

Data augmentation of MLPs

before: non-aug

before: aug

$$\Theta^{\text{GCNN}}(f, f') = \frac{1}{|G|} \sum_{g \in G} \Theta^{\text{MLP}}(f, \rho_{\text{reg}}(g) f')$$

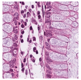
⇨ training the MLP on G-augmented data = training the GCNN on unaugmented data

in the ensemble mean, $\forall t$, $\forall x$

Data augmentation of CNNs

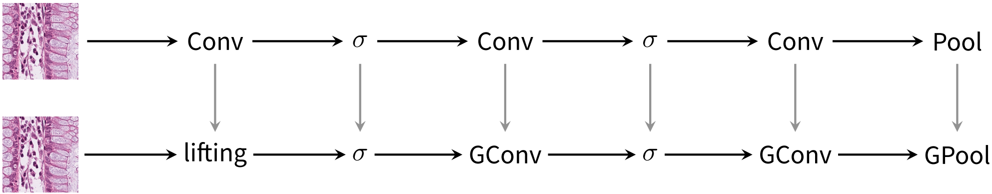
Data augmentation of CNNs

- Consider a CNN

 $\longrightarrow$ Conv $\longrightarrow$ σ $\longrightarrow$ Conv $\longrightarrow$ σ $\longrightarrow$ Conv $\longrightarrow$ Pool

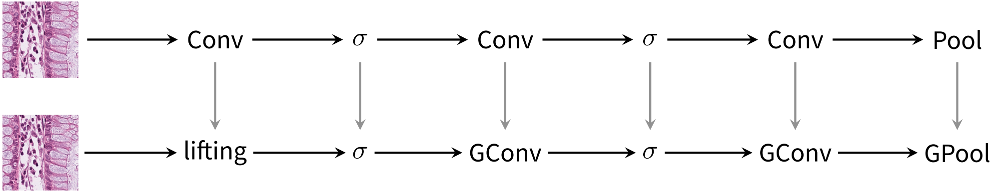
Data augmentation of CNNs

- Consider a CNN and a GCNN invariant wrt. roto-translations



Data augmentation of CNNs

- Consider a CNN and a GCNN invariant wrt. roto-translations

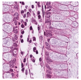


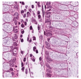
- Then

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{n} \sum_{r \in C_n} \Theta^{\mathrm{CNN}}(f, \rho_{\mathrm{reg}}(r) f')$$

Data augmentation of CNNs

- Consider a CNN and a GCNN invariant wrt. roto-translations





$$
\begin{tikzcd}
{} \arrow[r] & \text{Conv} \arrow[r] \arrow[d] & \sigma \arrow[r] \arrow[d] & \text{Conv} \arrow[r] \arrow[d] & \sigma \arrow[r] \arrow[d] & \text{Conv} \arrow[r] \arrow[d] & \text{Pool} \arrow[d] \\
{} \arrow[r] & \text{lifting} \arrow[r] & \sigma \arrow[r] & \text{GConv} \arrow[r] & \sigma \arrow[r] & \text{GConv} \arrow[r] & \text{GPool}
\end{tikzcd}
$$

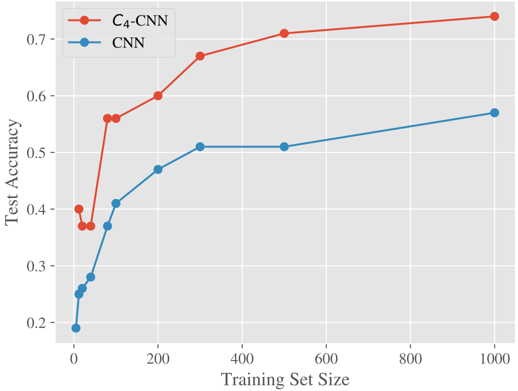
- Then

$$\Theta^{\mathrm{GCNN}}(f, f') = \frac{1}{n} \sum_{r \in C_n} \Theta^{\mathrm{CNN}}(f, \rho_{\mathrm{reg}}(r) f')$$

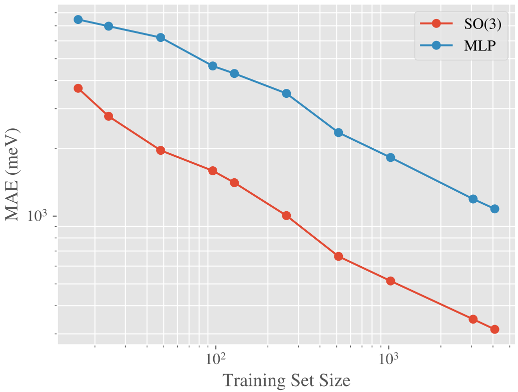
⇨ By training the CNN on rotated images, one obtains a roto-translation invariant GCNN

Experiments

Equivariant NTKs for medical image classification

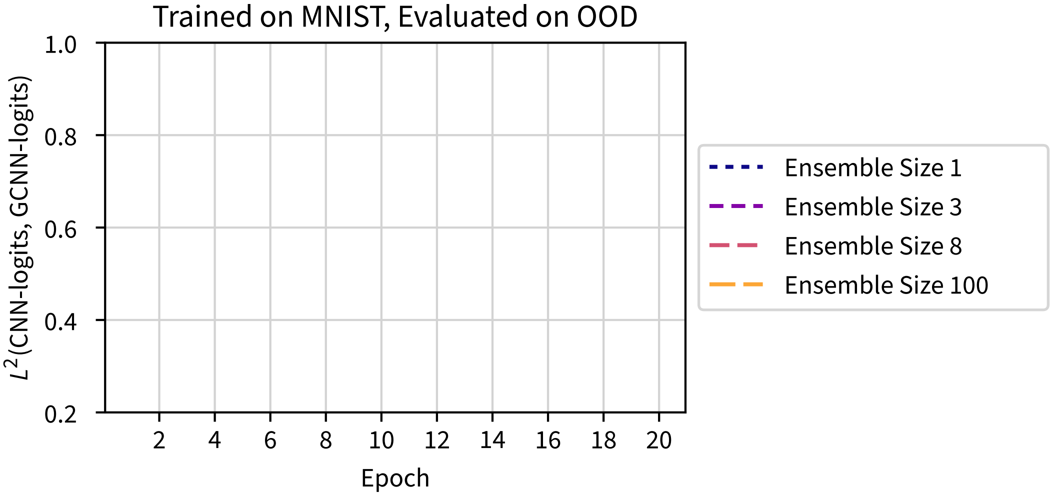


Equivariant NTKs for molecular property regression

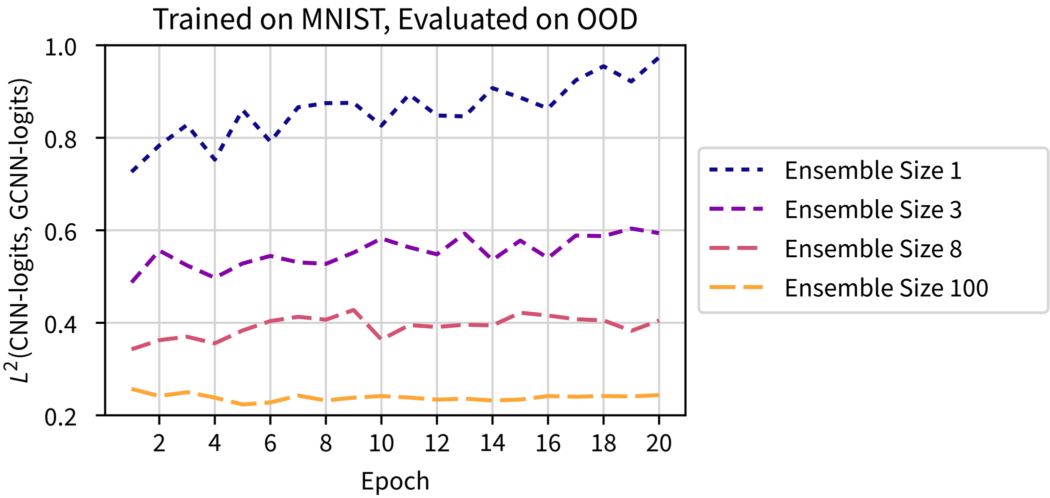


Convergence of augmented CNNs to GCNNs

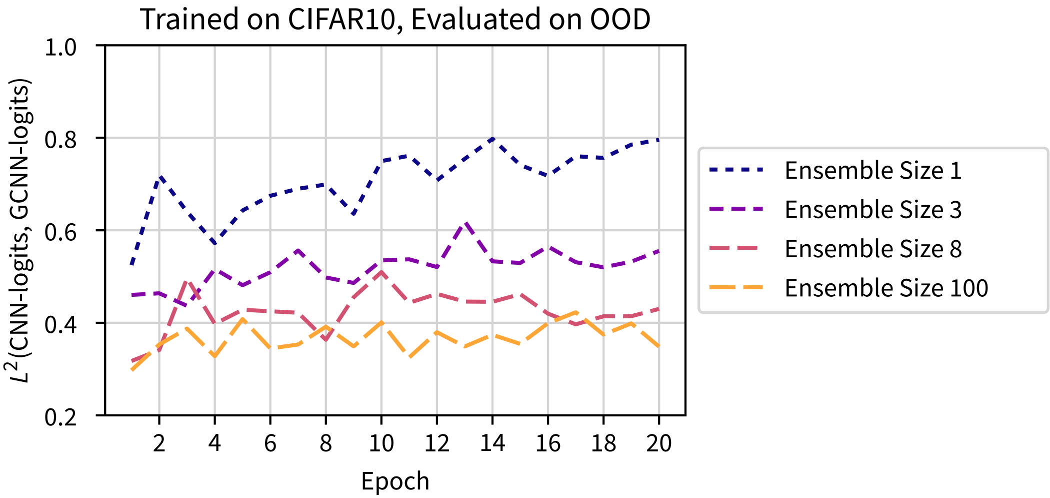
Convergence of augmented CNNs to GCNNs



Convergence of augmented CNNs to GCNNs



Convergence of augmented CNNs to GCNNs



Learning Topology
with
Gauge Equivariant Neural Networks

Gauge symmetry

- So far, we discussed global symmetries:

$$[\rho_{\text{reg}}(g)f](x) = f(\rho(g^{-1})x) \quad g \in G$$

Gauge symmetry

- So far, we discussed global symmetries:

$$[\rho_{\text{reg}}(g)f](x) = f(\rho(g^{-1})x) \quad g \in G$$

- Now, extend to local (gauge) symmetries:

$$[\rho_{\text{gauge}}(g)f](x) = f(\rho(g_x^{-1})x) \quad g_x \in G$$

Gauge symmetry

- So far, we discussed global symmetries:

$$[\rho_{\text{reg}}(g)f](x) = f(\rho(g^{-1})x) \quad g \in G$$

- Now, extend to local (gauge) symmetries:

$$[\rho_{\text{gauge}}(g)f](x) = f(\rho(g_x^{-1})x) \quad g_x \in G$$

- For instance, choose basis in tangent space at each point

Gauge symmetry

- So far, we discussed global symmetries:

$$[\rho_{\text{reg}}(g)f](x) = f(\rho(g^{-1})x) \quad g \in G$$

- Now, extend to local (gauge) symmetries:

$$[\rho_{\text{gauge}}(g)f](x) = f(\rho(g_x^{-1})x) \quad g_x \in G$$

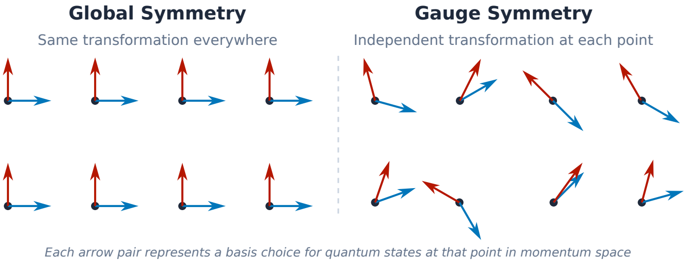
- For instance, choose basis in tangent space at each point

ⓘ This symmetry is exponentially larger!

$$G_{\text{gauge}} = G^{|X|}$$

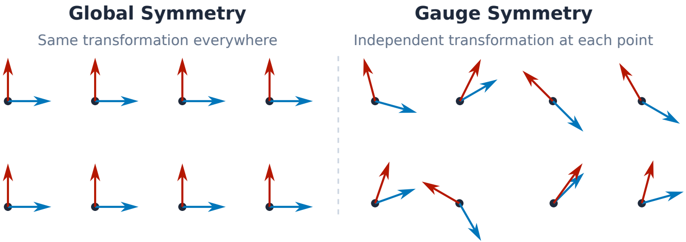
Topological materials

- Need to choose an arbitrary basis for quantum states at each point in momentum space



Topological materials

- Need to choose an arbitrary basis for quantum states at each point in momentum space

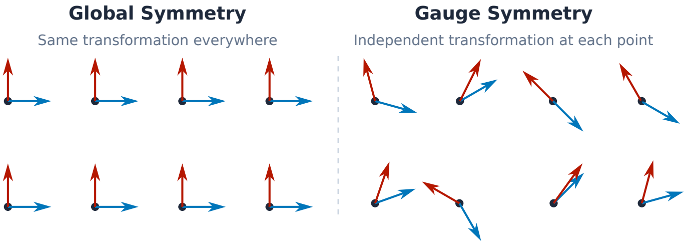


Each arrow pair represents a basis choice for quantum states at that point in momentum space

⇨ $U(N)$ gauge symmetry at each point

Topological materials

- Need to choose an arbitrary basis for quantum states at each point in momentum space



Each arrow pair represents a basis choice for quantum states at that point in momentum space

➪ $U(N)$ gauge symmetry at each point

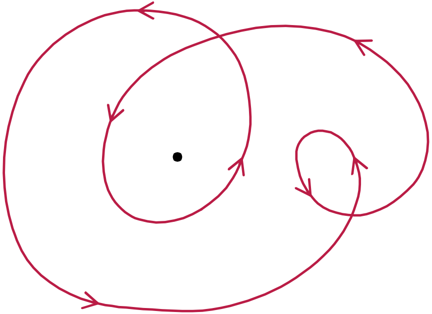
ⓘ Topology of states determines physical properties and is independent of choice of basis

The Chern number

- The topology is characterized by the **Chern number**

$$C = \frac{1}{2\pi i}\int \mathrm{Tr}(F(k))\mathrm{d}^2k$$

- Higher-dimensional analog of winding number



$$w = \frac{1}{2\pi}\oint_\gamma \mathrm{d}\theta$$

The Chern number

- On a discrete lattice in 2d, it is given by

Wilson loop, $N \times N$ matrix

$$C = \sum_{i,j} \mathrm{Im}(\mathrm{Tr}(\log W_{ij})) \in \mathbb{Z}$$

The Chern number

- On a discrete lattice in 2d, it is given by

Wilson loop, $N \times N$ matrix

$$C = \sum_{i,j} \mathrm{Im}(\mathrm{Tr}(\log W_{ij})) \in \mathbb{Z}$$

- The Chern number is invariant under gauge transformations

$$W_{ij} \to \Omega_{ij}^\dagger W_{ij} \Omega_{ij} \quad \text{with} \quad \Omega_{ij} \in U(N)$$

The Chern number

- On a discrete lattice in 2d, it is given by

Wilson loop, $N \times N$ matrix

$$C = \sum_{i,j} \mathrm{Im}(\mathrm{Tr}(\log W_{ij})) \in \mathbb{Z}$$

- The Chern number is invariant under gauge transformations

$$W_{ij} \to \Omega_{ij}^{\dagger} W_{ij} \Omega_{ij} \quad \text{with} \quad \Omega_{ij} \in U(N)$$

The Chern number

- On a discrete lattice in 2d, it is given by

Wilson loop, $N \times N$ matrix

$$C = \sum_{i,j} \mathrm{Im}(\mathrm{Tr}(\log W_{ij})) \in \mathbb{Z}$$

- The Chern number is invariant under gauge transformations

$$W_{ij} \to \Omega_{ij}^{\dagger} W_{ij} \Omega_{ij} \quad \text{with} \quad \Omega_{ij} \in U(N)$$

Learning problem: Predict C given $\{W_{ij} \,|\, (i,j) \in \text{lattice}\}$

Learning Determinants

- Since $\mathrm{Tr}(\log X) = \log(\det X)$, try learning the determinant as a toy problem

Learning Determinants

- Since $\mathrm{Tr}(\log X) = \log(\det X)$, try learning the determinant as a toy problem

ⓘ MLPs cannot learn determinants

Learning Determinants

- The determinant of a 4×4 matrix is

$$\det A = \sum_{i,j,k,l=1}^{4} \varepsilon^{ijkl} A_{1i} A_{2j} A_{3k} A_{4l}$$

Learning Determinants

- The determinant of a 4×4 matrix is

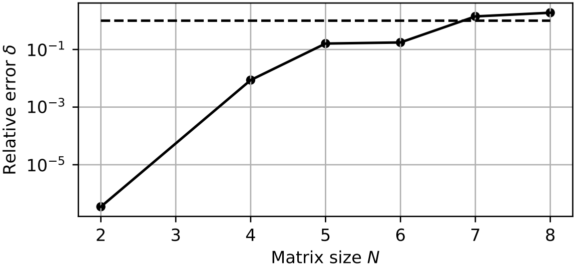
$$\det A = \sum_{i,j,k,l=1}^{4} \varepsilon^{ijkl} A_{1i} A_{2j} A_{3k} A_{4l}$$

- Build a ResNet with polynomial layers

$$A_{ij}^{\text{out}} = \sum_{k_1,\ldots,k_{2R}} \theta_{ij}^{k_1 \cdots k_{2R}} A_{k_1 k_2}^{\text{in}} \cdots A_{k_{2R-1} k_{2R}}^{\text{in}}$$

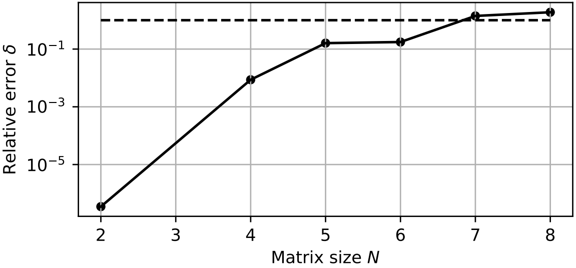
Learning Determinants

- Even deep models with layers of order $R \leq 4$ cannot learn determinants of larger matrices



Learning Determinants

- Even deep models with layers of order $R \leq 4$ cannot learn determinants of larger matrices



- Use gauge equivariant model to learn Chern number

Gauge Equivariant Network

[Favoni et al. 2022]

- Gauge equivariant network for lattice QCD

Gauge Equivariant Network

[Favoni et al. 2022]

- Gauge equivariant network for lattice QCD
- Gauge Equivariant Bilinear Layer (GEBL)

channel index

$$W'^{\gamma}_{ij} = \sum_{\mu\nu} \alpha_{\gamma\mu\nu} W^{\mu}_{ij} W^{\nu}_{ij}$$

trainable parameters

Gauge Equivariant Network

[Favoni et al. 2022]

- Gauge equivariant network for lattice QCD
- Gauge Equivariant Bilinear Layer (GEBL)

channel index

$$W'^{\gamma}_{ij} = \sum_{\mu\nu} \alpha_{\gamma\mu\nu} W^{\mu}_{ij} W^{\nu}_{ij}$$

trainable parameters

- Gauge Equivariant Activation Layer (GEAct)

$$W'^{\gamma}_{ij} = \sigma\left(\mathrm{Re}\left(\mathrm{Tr}\, W^{\gamma}_{ij}\right)\right) W^{\gamma}_{ij}$$

Gauge Equivariant Network

[Favoni et al. 2022]

- Gauge equivariant network for lattice QCD
- Gauge Equivariant Bilinear Layer (GEBL)

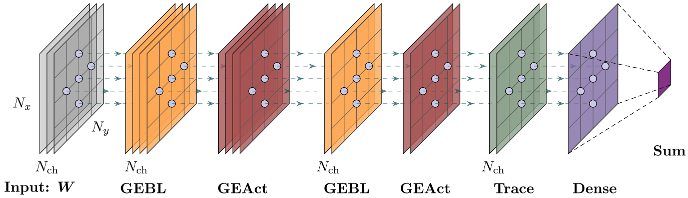
channel index

$$W'^{\gamma}_{ij} = \sum_{\mu\nu} \alpha_{\gamma\mu\nu} W^{\mu}_{ij} W^{\nu}_{ij}$$

trainable parameters

- Gauge Equivariant Activation Layer (GEAct)

$$W'^{\gamma}_{ij} = \sigma(\mathrm{Re}(\mathrm{Tr}\, W^{\gamma}_{ij}))W^{\gamma}_{ij}$$



Universal Approximation Theorem

[Huang et al. 2025]

For a compact Lie group G and bounded and non-decreasing activation function σ, GEBLNets can approximate any gauge invariant function.

Training

- Train on randomly sampled Wilson loops W_{ij}, generate labels from analytical expression

Training

- Train on randomly sampled Wilson loops W_{ij}, generate labels from analytical expression
- Training is unstable due to bilinear layers

Training

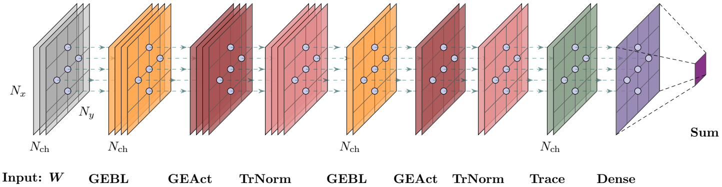
- Train on randomly sampled Wilson loops W_{ij}, generate labels from analytical expression
- Training is unstable due to bilinear layers
- Introduce gauge equivariant normalization layer (TrNorm)

$$W'^{\gamma}_{ij} = \frac{1}{|\text{mean}_{\gamma}(\text{Tr}\, W^{\gamma}_{ij})|} W^{\gamma}_{ij}$$

Training

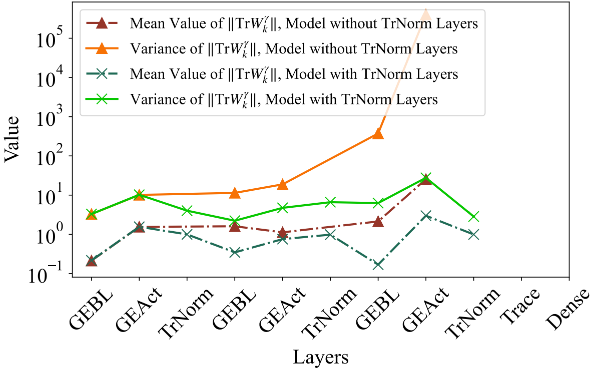
- Train on randomly sampled Wilson loops W_{ij}, generate labels from analytical expression
- Training is unstable due to bilinear layers
- Introduce gauge equivariant normalization layer (TrNorm)

$$W'^{\gamma}_{ij} = \frac{1}{|\mathrm{mean}_\gamma(\mathrm{Tr}\, W^\gamma_{ij})|} W^\gamma_{ij}$$



Training

The TrNorm layers stabilize the training statistics



Results

The normalized network can predict Chern numbers for higher bands

Bands	4	5	6	7	8
Accuracy	95.9%	94.0%	93.8%	91.7%	52.5%

Learning from trivial Chern numbers

- The entire network architecture before the final sum is per-site

Learning from trivial Chern numbers

- The entire network architecture before the final sum is per-site
- Can learn local features just from topologically trivial ($C = 0$) samples: No training labels necessary!

Learning from trivial Chern numbers

- The entire network architecture before the final sum is per-site
- Can learn local features just from topologically trivial ($C = 0$) samples: No training labels necessary!
- Network generalizes to topologically non-trivial ($C > 0$) samples

Learning from trivial Chern numbers

- Problem: Collapse to zero-solution

Learning from trivial Chern numbers

- Problem: Collapse to zero-solution

- Solution: Add loss L_{std} to encourage non-zero std across sites

minimum target std

$$L_{std} = \left\| \min(\mathrm{std}(g(W_{ij})), \delta) - \delta \right\|_1$$

per-site output of final dense layer

Learning from trivial Chern numbers

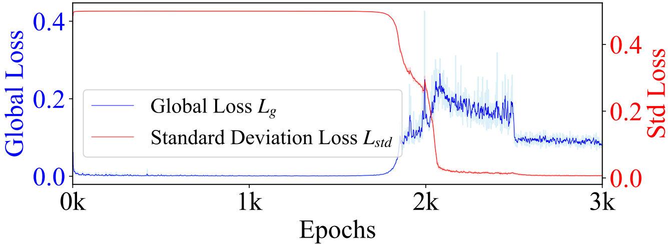
- Problem: Collapse to zero-solution

- Solution: Add loss L_{std} to encourage non-zero std across sites

minimum target std

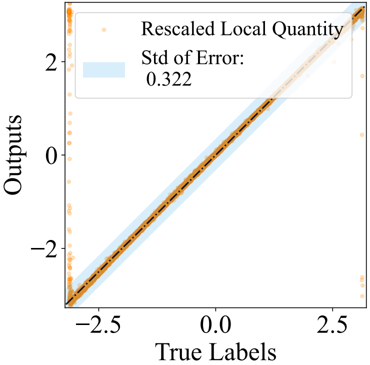
$$L_{std} = \| \min(\mathrm{std}(g(W_{ij})), \delta) - \delta \|_1$$

per-site output of final dense layer



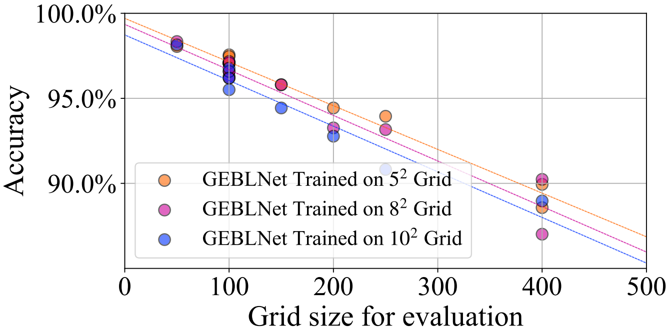
Learning from trivial Chern numbers

Network trained on trivial topology alone can predict non-trivial Chern numbers up to global factor



Generalization to larger grids

Since the learned features are local, network generalizes to larger grids



Key takeaways

① Training dynamics of equivariant networks can be studied using equivariant NTKs

Key takeaways

① Training dynamics of equivariant networks can be studied using equivariant NTKs

② In the infinite-width limit, MLPs trained with data augmentation are equivalent to GCNNs

Key takeaways

① Training dynamics of equivariant networks can be studied using equivariant NTKs

② In the infinite-width limit, MLPs trained with data augmentation are equivalent to GCNNs

③ Gauge equivariant networks are required to learn Chern numbers of topological materials

Key takeaways

① Training dynamics of equivariant networks can be studied using equivariant NTKs

② In the infinite-width limit, MLPs trained with data augmentation are equivalent to GCNNs

③ Gauge equivariant networks are required to learn Chern numbers of topological materials

④ Networks trained on trivial topologies generalize to non-trivial topologies

Papers

- **Equivariant Neural Tangent Kernels**
  Philipp Misof, Pan Kessel, Jan E. Gerken
  ICML 2025
- **Learning Chern Numbers of Topological Insulators with Gauge Equivariant Neural Networks**
  Longde Huang, Oleksandr Balabanov, Hampus Linander, Mats Granath, Daniel Persson, Jan E. Gerken
  NeurIPS 2025

Thank you