Neural Tangent Kernels:

Data augmentation and Feynman diagrams

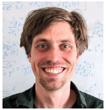
Jan E. Gerken

CHALMERS
UNIVERSITY OF TECHNOLOGY

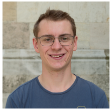
UNIVERSITY OF GOTHENBURG

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

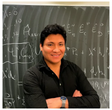
in collaboration with



Pan Kessel



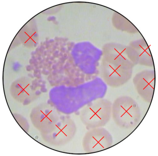
Philipp Misof



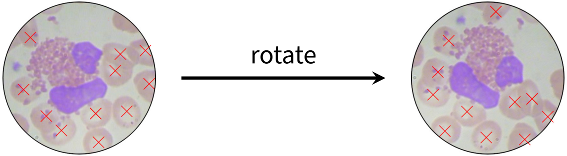
Max Guillen

Symmetries in deep learning

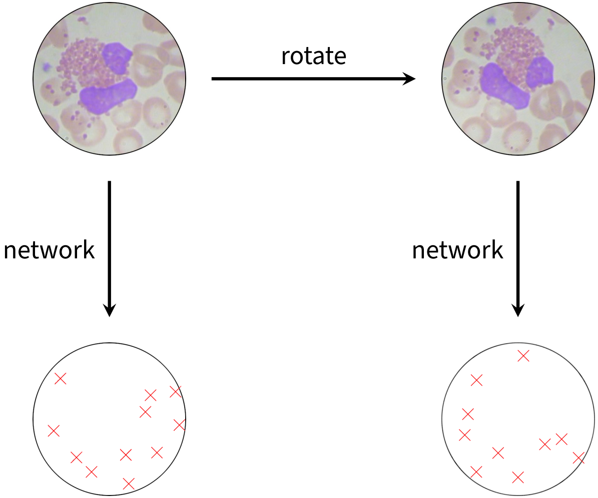
Symmetries in deep learning



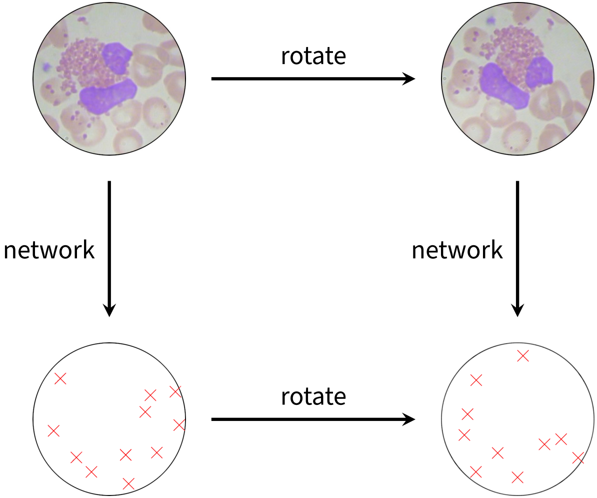
Symmetries in deep learning



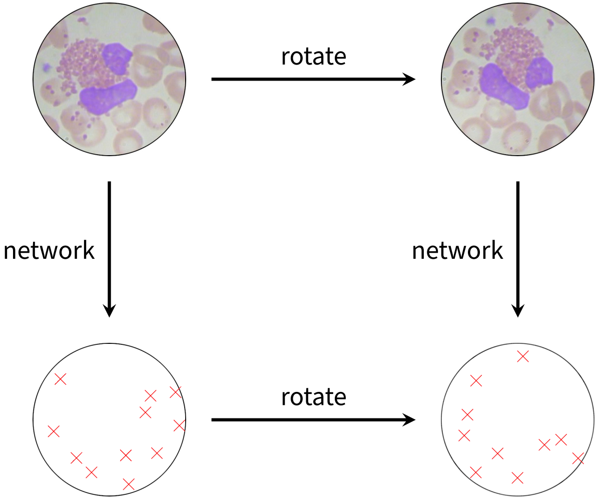
Symmetries in deep learning



Symmetries in deep learning

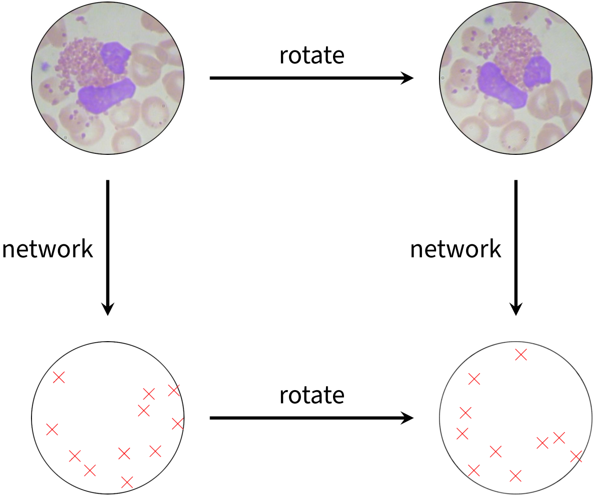


Symmetries in deep learning



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

Symmetries in deep learning



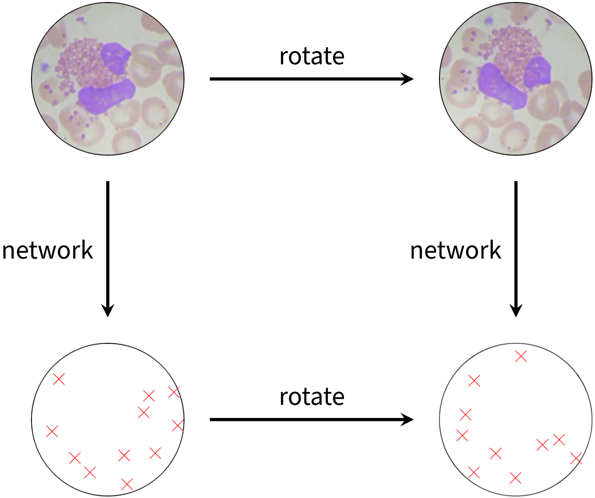
network

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

rotation

image

Equivariance



network

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

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 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-

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 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

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

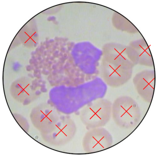
Ron O. Dror
Department of Computer Science
Stanford University
rondror@cs.stanford.edu

ABSTRACT

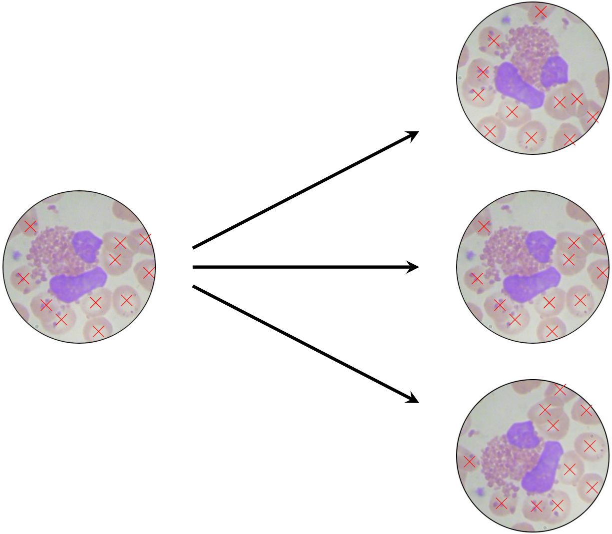
Predicting the structure of multi-protein complexes is a grand challenge in biochemistry, with major implications for basic science and drug discovery. Computational structure prediction methods generally leverage pre-defined structural features to distinguish accurate structural models from less accurate 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

Data augmentation

Data augmentation



Data augmentation



Data augmentation

Data augmentation

Article

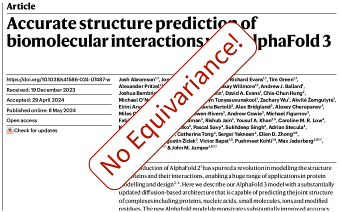
Accurate structure prediction of biomolecular interactions with AlphaFold 3

https://doi.org/10.1038/s41586-024-07487-w
Received: 19 December 2023
Accepted: 29 April 2024
Published online: 8 May 2024
Open access
Check for updates

Josh Abramson[1,7], Jonas Adler[1,7], Jack Dunger[1,7], Richard Evans[1,7], Tim Green[1,7], Alexander Pritzel[1,7], Olaf Ronneberger[1,7], Lindsay Willmore[1,7], Andrew J. Ballard[1], Joshua Bambrick[2], Sebastian W. Bodenstein[1], David A. Evans[1], Chia-Chun Hung[2], Michael O'Neill[1], David Reiman[1], Kathryn Tunyasuvunakool[1], Zachary Wu[1], Akvilė Žemgulytė[1], Eirini Arvaniti[3], Charles Beattie[3], Ottavia Bertolli[3], Alex Bridgland[3], Alexey Cherepanov[4], Miles Congreve[4], Alexander I. Cowen-Rivers[3], Andrew Cowie[3], Michael Figurnov[3], Fabian B. Fuchs[3], Hannah Gladman[3], Rishub Jain[3], Yousuf A. Khan[3,5], Caroline M. R. Low[4], Kuba Perlin[3], Anna Potapenko[3], Pascal Savy[4], Sukhdeep Singh[3], Adrian Stecula[4], Ashok Thillaisundaram[3], Catherine Tong[4], Sergei Yakneen[4], Ellen D. Zhong[3,6], Michal Zielinski[3], Augustin Žídek[3], Victor Bapst[1,8], Pushmeet Kohli[1,8], Max Jaderberg[2,8] ✉, Demis Hassabis[1,2,8] ✉ & John M. Jumper[1,8] ✉

The introduction of AlphaFold 2[1] has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein modelling and design[2–6]. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy

Data augmentation



Data augmentation

Article

Accurate structure predictio[illegible] of biomolecular interactions [illegible]lphaFold 3

https://doi.org/10.1038/s41586-024-07487-w
Received: 19 December 2023
Accepted: 29 April 2024
Published online: 8 May 2024
Open access
Check for updates

Josh Abramson[1,7], Jo[illegible] Richard Evans[1,7], Tim Green[1,7], Alexander Pritzel[1,7] [illegible]say Willmore[1,7], Andrew J. Ballard[1], Joshua Bambri[illegible]in[1], David A. Evans[1], Chia-Chun Hung[2], Michael O'N[illegible]yn Tunyasuvunakool[1], Zachary Wu[1], Akvilė Žemgulytė[1], Eirini Arv[illegible]avia Bertolli[3], Alex Bridgland[3], Alexey Cherepanov[4], Miles [illegible]owen-Rivers[3], Andrew Cowie[3], Michael Figurnov[3], Fab[illegible]dman[3], Rishub Jain[3], Yousuf A. Khan[3,5], Caroline M. R. Low[4], [illegible]nko[3], Pascal Savy[4], Sukhdeep Singh[3], Adrian Stecula[4], [illegible], Catherine Tong[4], Sergei Yakneen[4], Ellen D. Zhong[3,6], [illegible]gustin Žídek[3], Victor Bapst[1,8], Pushmeet Kohli[1,8], Max Jaderberg[2,8], [illegible] & John M. Jumper[1,8]

[illegible]duction of AlphaFold 2[1] has spurred a revolution in modelling the structure [illegible]teins and their interactions, enabling a huge range of applications in protein [illegible]odelling and design[2–6]. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy

No Equivariance!

The Importance of Being Scalable: Improving the Speed and Accuracy of Neural Network Interatomic Potentials Across Chemical Domains

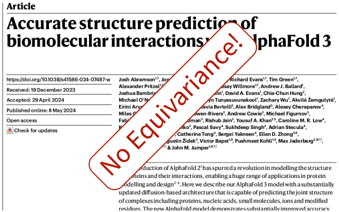
Eric Qu
UC Berkeley
ericqu@berkeley.edu

Aditi S. Krishnapriyan
UC Berkeley, LBNL
aditik1@berkeley.edu

Abstract

Scaling has been a critical factor in improving model performance and generalization across various fields of machine learning. It involves how a model's performance changes with increases in model size or input data, as well as how efficiently computational resources are utilized to support this growth. Despite successes in scaling other types of machine learning models, the study of scaling in Neural Network Interatomic Potentials (NNIPs) remains limited. NNIPs act as

Data augmentation



Article

Accurate structure predictio[n] of biomolecular interactions w[ith Al]phaFold 3

https://doi.org/10.1038/s41586-024-07487-w

Received: 19 December 2023

Accepted: 29 April 2024

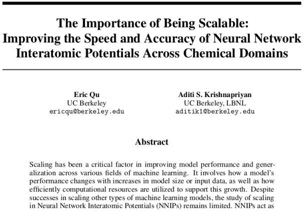
Published online: 8 May 2024

Open access

Check for updates

Josh Abramson[1,7], Jon… Richard Evans[1,7], Tim Green[1,7],
Alexander Pritzel[1,7] …dsay Willmore[1,7], Andrew J. Ballard[1],
Joshua Bambric… …ein[1], David A. Evans[1], Chia-Chun Hung[2],
Michael O'N… …yn Tunyasuvunakool[1], Zachary Wu[1], Akvilė Žemgulytė[1],
Eirini Arv… …avia Bertolli[3], Alex Bridgland[3], Alexey Cherepanov[4],
Miles C… …owen-Rivers[3], Andrew Cowie[3], Michael Figurnov[3],
Fabi… …dman[3], Rishub Jain[3], Yousuf A. Khan[3,5], Caroline M. R. Low[4],
K… …nko[3], Pascal Savy[4], Sukhdeep Singh[3], Adrian Stecula[4],
…, Catherine Tong[4], Sergei Yakneen[4], Ellen D. Zhong[3,6],
…gustin Žídek[3], Victor Bapst[1,8], Pushmeet Kohli[1,8], Max Jaderberg[2,8],
… & John M. Jumper[1,8]

…oduction of AlphaFold 2[1] has spurred a revolution in modelling the structure …oteins and their interactions, enabling a huge range of applications in protein …odelling and design[2–6]. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy



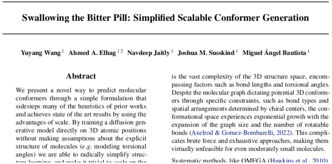
The Importance of Being Scalable: Improving the Speed and Accuracy of Neural Network Interatomic Potentials Across Chemical Domains

Eric Qu
UC Berkeley
ericqu@berkeley.edu

Aditi S. Krishnapriyan
UC Berkeley, LBNL
aditik1@berkeley.edu

Abstract

Scaling has been a critical factor in improving model performance and generalization across various fields of machine learning. It involves how a model's performance changes with increases in model size or input data, as well as how efficiently computational resources are utilized to support this growth. Despite successes in scaling other types of machine learning models, the study of scaling in Neural Network Interatomic Potentials (NNIPs) remains limited. NNIPs act as



Swallowing the Bitter Pill: Simplified Scalable Conformer Generation

Yuyang Wang[1] Ahmed A. Elhag[1 2] Navdeep Jaitly[1] Joshua M. Susskind[1] Miguel Ángel Bautista[1]

Abstract

We present a novel way to predict molecular conformers through a simple formulation that sidesteps many of the heuristics of prior works and achieves state of the art results by using the advantages of scale. By training a diffusion generative model directly on 3D atomic positions without making assumptions about the explicit structure of molecules (*e.g.* modeling torsional angles) we are able to radically simplify struc-

is the vast complexity of the 3D structure space, encompassing factors such as bond lengths and torsional angles. Despite the molecular graph dictating potential 3D conformers through specific constraints, such as bond types and spatial arrangements determined by chiral centers, the conformational space experiences exponential growth with the expansion of the graph size and the number of rotatable bonds (Axelrod & Gomez-Bombarelli, 2022). This complicates brute force and exhaustive approaches, making them virtually unfeasible for even moderately small molecules.

Systematic methods, like OMEGA (Hawkins et al., 2010),

Data augmentation

Article

Accurate structure prediction of biomolecular interactions ... AlphaFold 3

No Equivariance!

https://doi.org/10.1038/s41586-024-07487-w
Received: 19 December 2023
Accepted: 29 April 2024
Published online: 8 May 2024
Open access
Check for updates

Josh Abramson[1,7], Jon... Richard Evans[1,7], Tim Green[1,7], Alexander Pritzel[1,7], ...dsay Willmore[1,7], Andrew J. Ballard[1], Joshua Bambric... ...ein[1], David A. Evans[1], Chia-Chun Hung[2], Michael O'N... ...yn Tunyasuvunakool[1], Zachary Wu[1], Akvilė Žemgulytė[1], Eirini Arva... ...avia Bertolli[3], Alex Bridgland[3], Alexey Cherepanov[4], Miles C... ...owen-Rivers[3], Andrew Cowie[3], Michael Figurnov[3], Fabi... ...dman[3], Rishub Jain[3], Yousuf A. Khan[3,5], Caroline M. R. Low[4], K... ...nko[3], Pascal Savy[4], Sukhdeep Singh[3], Adrian Stecula[4], ... Catherine Tong[4], Sergei Yakneen[4], Ellen D. Zhong[3,6], ...gustin Žídek[3], Victor Bapst[1,8], Pushmeet Kohli[1,8], Max Jaderberg[2,8], ... & John M. Jumper[1,8]

...oduction of AlphaFold 2[1] has spurred a revolution in modelling the structure ...oteins and their interactions, enabling a huge range of applications in protein ...odelling and design[2–6]. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy

The Importance of Being Scalable: Improving the Speed and Accuracy of Neural Network Interatomic Potentials Across Chemical Domains

Eric Qu
UC Berkeley
ericqu@berkeley.edu

Aditi S. Krishnapriyan
UC Berkeley, LBNL
aditik1@berkeley.edu

Abstract

Scaling has been a critical factor in improving model performance and generalization across various fields of machine learning. It involves how a model's performance changes with increases in model size or input data, as well as how efficiently computational resources are utilized to support this growth. Despite successes in scaling other types of machine learning models, the study of scaling in Neural Network Interatomic Potentials (NNIPs) remains limited. NNIPs act as

Swallowing the Bitter Pill: Simplified Scalable Conformer Generation

Yuyang Wang[1] Ahmed A. Elhag[1,2] Navdeep Jaitly[1] Joshua M. Susskind[1] Miguel Ángel Bautista[1]

Abstract

We present a novel way to predict molecular conformers through a simple formulation that sidesteps many of the heuristics of prior works and achieves state of the art results by using the advantages of scale. By training a diffusion generative model directly on 3D atomic positions without making assumptions about the explicit structure of molecules (*e.g.* modeling torsional angles) we are able to radically simplify structure learning, and make it trivial to scale up the

is the vast complexity of the 3D structure space, encompassing factors such as bond lengths and torsional angles. Despite the molecular graph dictating potential 3D conformers through specific constraints, such as bond types and spatial arrangements determined by chiral centers, the conformational space experiences exponential growth with the expansion of the graph size and the number of rotatable bonds (Axelrod & Gomez-Bombarelli, 2022). This complicates brute force and exhaustive approaches, making them virtually unfeasible for even moderately small molecules.

Systematic methods, like OMEGA (Hawkins et al., 2010),

Probing the effects of broken symmetries in machine learning

Marcel F Langer, Sergey N Pozdnyakov and Michele Ceriotti*
Laboratory of Computational Science and Modeling and National Centre for Computational Design and Discovery of Novel Materials MARVEL, Institute of Materials, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
* Author to whom any correspondence should be addressed.
E-mail: michele.ceriotti@epfl.ch

Keywords: machine learning, symmetry-constrained models, atomistic modeling, molecular simulations
Supplementary material for this article is available online

Abstract

Symmetry is one of the most central concepts in physics, and it is no surprise that it has also been widely adopted as an inductive bias for machine-learning models applied to the physical sciences. This is especially true for models targeting the properties of matter at the atomic scale. Both established and state-of-the-art approaches, with almost no exceptions, are built to be exactly equivariant to translations, permutations, and rotations of the atoms. Incorporating symmetries—rotations in particular—constrains the model design space and implies more complicated architectures that are often also computationally demanding. There are indications

Data augmentation

Article

Accurate structure prediction of biomolecular interactions ... phaFold 3

No Equivariance!

https://doi.org/10.1038/s41586-024-07487-w
Received: 19 December 2023
Accepted: 29 April 2024
Published online: 8 May 2024
Open access
Check for updates

Josh Abramson[1,7], Jon... Richard Evans[1,7], Tim Green[1,7], Alexander Pritzel[1,7], ...dsay Willmore[1,7], Andrew J. Ballard[1], Joshua Bambric... ...ein[1], David A. Evans[1], Chia-Chun Hung[2], Michael O'N... ...yn Tunyasuvunakool[1], Zachary Wu[1], Akvilė Žemgulytė[1], Eirini Arva... ...avia Bertolli[3], Alex Bridgland[3], Alexey Cherepanov[4], Miles C... ...owen-Rivers[3], Andrew Cowie[3], Michael Figurnov[3], Fabi... ...dman[3], Rishub Jain[3], Yousuf A. Khan[3,5], Caroline M. R. Low[4], K... ...nko[3], Pascal Savy[4], Sukhdeep Singh[3], Adrian Stecula[4], ... Catherine Tong[4], Sergei Yakneen[4], Ellen D. Zhong[3,6], ...gustin Žídek[3], Victor Bapst[1,8], Pushmeet Kohli[1,8], Max Jaderberg[2,8], ... & John M. Jumper[1,8]

...oduction of AlphaFold 2[1] has spurred a revolution in modelling the structure ...oteins and their interactions, enabling a huge range of applications in protein ...odelling and design[2–6]. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy

The Importance of Being Scalable: Improving the Speed and Accuracy of Neural Network Interatomic Potentials Across Chemical Domains

Eric Qu
UC Berkeley
ericqu@berkeley.edu

Aditi S. Krishnapriyan
UC Berkeley, LBNL
aditik1@berkeley.edu

Abstract

Scaling has been a critical factor in improving model performance and generalization across various fields of machine learning. It involves how a model's performance changes with increases in model size or input data, as well as how efficiently computational resources are utilized to support this growth. Despite successes in scaling other types of machine learning models, the study of scaling in Neural Network Interatomic Potentials (NNIPs) remains limited. NNIPs act as

Swallowing the Bitter Pill: Simplified Scalable Conformer Generation

Yuyang Wang[1] Ahmed A. Elhag[1,2] Navdeep Jaitly[1] Joshua M. Susskind[1] Miguel Ángel Bautista[1]

Abstract

We present a novel way to predict molecular conformers through a simple formulation that sidesteps many of the heuristics of prior works and achieves state of the art results by using the advantages of scale. By training a diffusion generative model directly on 3D atomic positions without making assumptions about the explicit structure of molecules (*e.g.* modeling torsional angles) we are able to radically simplify struc-

is the vast complexity of the 3D structure space, encompassing factors such as bond lengths and torsional angles. Despite the molecular graph dictating potential 3D conformers through specific constraints, such as bond types and spatial arrangements determined by chiral centers, the conformational space experiences exponential growth with the expansion of the graph size and the number of rotatable bonds (Axelrod & Gomez-Bombarelli, 2022). This complicates brute force and exhaustive approaches, making them virtually unfeasible for even moderately small molecules.

Systematic methods, like OMEGA (Hawkins et al., 2010),

Probing the effects of broken symmetries in machine learning

Marcel F Langer, Sergey N Pozdnyakov and Michele Ceriotti*

Laboratory of Computational Science and Modeling and National Centre for Computational Design and Discovery of Novel Materials MARVEL, Institute of Materials, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
* Author to whom any correspondence should be addressed.

E-mail: michele.ceriotti@epfl.ch

Keywords: machine learning, symmetry-constrained models, atomistic modeling, molecular simulations

Supplementary material for this article is available online

Abstract

Symmetry is one of the most central concepts in physics, and it is no surprise that it has also been widely adopted as an inductive bias for machine-learning models applied to the physical sciences. This is especially true for models targeting the properties of matter at the atomic scale. Both established and state-of-the-art approaches, with almost no exceptions, are built to be exactly equivariant to translations, permutations, and rotations of the atoms. Incorporating symmetries—rotations in particular—constrains the model design space and implies more complicated architectures that are often also computationally demanding. There are indications

Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics

Marloes Arts,*,†,‡,@ Victor Garcia Satorras,*,¶,@ Chin-Wei Huang,¶ Daniel Zügner,§ Marco Federici,†,|| Cecilia Clementi,§,⊥ Frank Noé,§ Robert Pinsler,# and Rianne van den Berg¶

†*Work done during an internship at Microsoft Research (Amsterdam).*
‡*University of Copenhagen, Department of Computer Science, Universitetsparken 1, Copenhagen, 2100, Denmark.*
¶*AI4Science, Microsoft Research, Evert van de Beekstraat 354, Amsterdam, 1118 CZ, The Netherlands.*
§*AI4Science, Microsoft Research, Karl-Liebknecht-Straße 32, Berlin, 10178, Germany.*
||*University of Amsterdam, Informatics Institute, Science Park 904, Amsterdam, 1098 XH, The Netherlands.*
⊥*Freie Universität Berlin, Department of Physics, Arnimalle 12, Berlin, 14195, Germany.*
#*AI4Science, Microsoft Research, 21 Station Road, Cambridge, CB1 2FB, United Kingdom.*
@*Equal contribution.*

* E-mail: ma@di.ku.dk; victorgar@microsoft.com

Abstract

Coarse-grained (CG) molecular dynamics enables the study of biological processes at temporal and spatial scales that would be intractable at an atomistic resolution. However, accurately learning a CG force field remains a challenge. In this work, we leverage connections between score-based generative models, force fields and molecular

Data augmentation

Article

Accurate structure predicti… of biomolecular interactions … phaFold 3

No Equivariance!

https://doi.org/10.1038/s41586-024-07487-w
Received: 19 December 2023
Accepted: 29 April 2024
Published online: 8 May 2024
Open access
Check for updates

Josh Abramson[1,7], Jo… Richard Evans[1,7], Tim Green[1,7],
Alexander Pritzel[1,7] … say Willmore[1,7], Andrew J. Ballard[1],
Joshua Bambri… ein[1], David A. Evans[1], Chia-Chun Hung[2],
Michael O'N… yn Tunyasuvunakool[1], Zachary Wu[1], Akvilė Žemgulytė[1],
Eirini Arv… avia Bertolli[3], Alex Bridgland[3], Alexey Cherepanov[4],
Miles … wen-Rivers[3], Andrew Cowie[3], Michael Figurnov[3],
Fab… dman[3], Rishub Jain[3], Yousuf A. Khan[3,5], Caroline M. R. Low[4],
K… nko[3], Pascal Savy[4], Sukhdeep Singh[3], Adrian Stecula[4],
…, Catherine Tong[4], Sergei Yakneen[4], Ellen D. Zhong[3,6],
… gustin Žídek[3], Victor Bapst[1,8], Pushmeet Kohli[1,8], Max Jaderberg[2,8],
… & John M. Jumper[1,8]

…oduction of AlphaFold 2[1] has spurred a revolution in modelling the structure …oteins and their interactions, enabling a huge range of applications in protein …odelling and design[2–6]. Here we describe our AlphaFold 3 model with a substantially updated diffusion-based architecture that is capable of predicting the joint structure of complexes including proteins, nucleic acids, small molecules, ions and modified residues. The new AlphaFold model demonstrates substantially improved accuracy

The Importance of Being Scalable: Improving the Speed and Accuracy of Neural Network Interatomic Potentials Across Chemical Domains

Eric Qu
UC Berkeley
ericqu@berkeley.edu

Aditi S. Krishnapriyan
UC Berkeley, LBNL
aditik1@berkeley.edu

Abstract

Scaling has been a critical factor in improving model performance and generalization across various fields of machine learning. It involves how a model's performance changes with increases in model size or input data, as well as how efficiently computational resources are utilized to support this growth. Despite successes in scaling other types of machine learning models, the study of scaling in Neural Network Interatomic Potentials (NNIPs) remains limited. NNIPs act as

Swallowing the Bitter Pill: Simplified Scalable Conformer Generation

Yuyang Wang[1] Ahmed A. Elhag[1,2] Navdeep Jaitly[1] Joshua M. Susskind[1] Miguel Ángel Bautista[1]

Abstract

We present a novel way to predict molecular conformers through a simple formulation that sidesteps many of the heuristics of prior works and achieves state of the art results by using the advantages of scale. By training a diffusion generative model directly on 3D atomic positions without making assumptions about the explicit structure of molecules (*e.g.* modeling torsional angles) we are able to radically simplify struc-

is the vast complexity of the 3D structure space, encompassing factors such as bond lengths and torsional angles. Despite the molecular graph dictating potential 3D conformers through specific constraints, such as bond types and spatial arrangements determined by chiral centers, the conformational space experiences exponential growth with the expansion of the graph size and the number of rotatable bonds (Axelrod & Gomez-Bombarelli, 2022). This complicates brute force and exhaustive approaches, making them virtually unfeasible for even moderately small molecules.

Probing the effects of broken symmetries in machine learning

Marcel F Langer, Sergey N Pozdnyakov and Michele Ceriotti*
Laboratory of Computational Science and Modeling and National Centre for Computational Design and Discovery of Novel Materials MARVEL, Institute of Materials, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
* Author to whom any correspondence should be addressed.
E-mail: michele.ceriotti@epfl.ch

Keywords: machine learning, symmetry-constrained models, atomistic modeling, molecular simulations
Supplementary material for this article is available online

Abstract

Symmetry is one of the most central concepts in physics, and it is no surprise that it has also been widely adopted as an inductive bias for machine-learning models applied to the physical sciences. This is especially true for models targeting the properties of matter at the atomic scale. Both established and state-of-the-art approaches, with almost no exceptions, are built to be exactly equivariant to translations, permutations, and rotations of the atoms. Incorporating symmetries—rotations in particular—constrains the model design space and implies more complicated architectures that are often also computationally demanding. There are indications

Two for One: Diffusion Models and Force Fields for Coarse-Grained Molecular Dynamics

Marloes Arts,[*,†,‡,@] Victor Garcia Satorras,[*,¶,@] Chin-Wei Huang,[¶] Daniel Zügner,[§] Marco Federici,[†,‖] Cecilia Clementi,[§,⊥] Frank Noé,[§] Robert Pinsler,[#] and Rianne van den Berg[¶]

†*Work done during an internship at Microsoft Research (Amsterdam).*
‡*University of Copenhagen, Department of Computer Science, Universitetsparken 1, Copenhagen, 2100, Denmark.*
¶*AI4Science, Microsoft Research, Evert van de Beekstraat 354, Amsterdam, 1118 CZ, The Netherlands.*
§*AI4Science, Microsoft Research, Karl-Liebknecht-Straße 32, Berlin, 10178, Germany.*
‖*University of Amsterdam, Informatics Institute, Science Park 904, Amsterdam, 1098 XH, The Netherlands.*
⊥*Freie Universität Berlin, Department of Physics, Arnimalle 12, Berlin, 14195, Germany.*
#*AI4Science, Microsoft Research, 21 Station Road, Cambridge, CB1 2FB, United Kingdom.*
@*Equal contribution.*

* E-mail: ma@di.ku.dk; victorgar@microsoft.com

Abstract

Coarse-grained (CG) molecular dynamics enables the study of biological processes at temporal and spatial scales that would be intractable at an atomistic resolution. However, accurately learning a CG force field remains a challenge. In this work, we leverage connections between score-based generative models, force fields and molecular

Does Equivariance Matter at Scale?

Johann Brehmer* Sönke Behrends Pim de Haan* Taco Cohen*
Qualcomm AI Research[†]
mail@johannbrehmer.de

Abstract

Given large data sets and sufficient compute, is it beneficial to design neural architectures for the structure and symmetries of each problem? Or is it more efficient to learn them from data? We study empirically how equivariant and non-equivariant networks scale with compute and training samples. Focusing on a benchmark problem of rigid-body interactions and on general-purpose transformer architectures, we perform a series of experiments, varying the model size, training steps, and dataset size. We find evidence for three conclusions. First, equivariance improves data efficiency, but training non-equivariant models with data augmentation can close this gap given sufficient epochs. Second, scaling with compute follows a power law, with equivariant models outperforming non-equivariant ones at each tested compute budget. Finally, the optimal allocation of a compute budget onto model size and training duration differs between equivariant and non-equivariant models.

Data augmentation

- 👍 Easy to implement
- 👍 No specialized architecture necessary

Data augmentation

- 👍 Easy to implement
- 👍 No specialized architecture necessary
- 👎 No exact equivariance

Data augmentation

- 👍 Easy to implement
- 👍 No specialized architecture necessary
- 👎 No exact equivariance

Can we understand data augmentation theoretically?

Empirical NTK

Training dynamics under continuous gradient descent:

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

learning rate → η

loss → L

training sample → x_i

Empirical NTK

Training dynamics under continuous gradient descent:

learning rate

loss

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

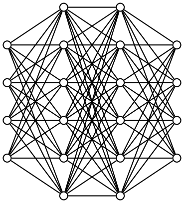
training sample

with the empirical neural tangent kernel (NTK)

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

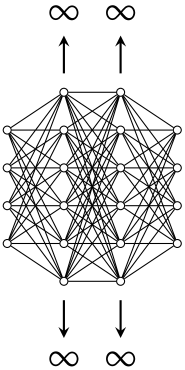
Infinite width limit

[Jacot et al. 2018]



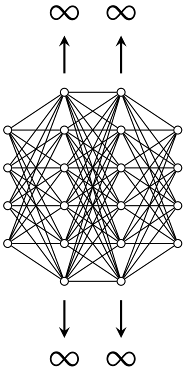
Infinite width limit

[Jacot et al. 2018]



Infinite width limit

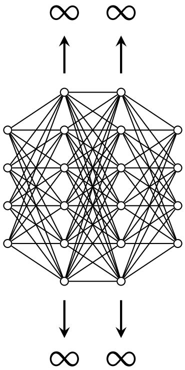
[Jacot et al. 2018]



👍 NTK becomes independent of initialization

Infinite width limit

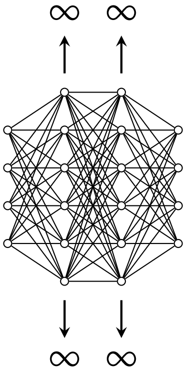
[Jacot et al. 2018]



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

Infinite width limit

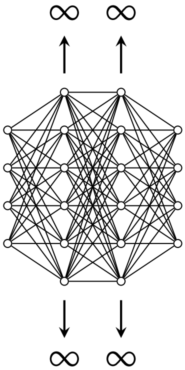
[Jacot et al. 2018]



- 👍 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]

ⓘ 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

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

learning rate

train data

Data augmentation

Data augmentation at infinite width

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

Data augmentation at infinite width

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

augmented data

augmented labels

Data augmentation at infinite width

group transformation

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

augmented data

augmented labels

Kernel transformation

The neural tangent kernel Θ as well as the NNGP kernel K transform according to

$$\Theta(\rho(g)x, \rho(g)x') = \rho_K(g)\Theta(x, x')\rho_K^\top(g)\,,$$
$$K(\rho(g)x, \rho(g)x') = \rho_K(g)K(x, x')\rho_K^\top(g)\,,$$

for all $g \in G$ and $x, x' \in X$.

Kernel transformation

The neural tangent kernel Θ as well as the NNGP kernel K transform according to

$$\Theta(\rho(g)x, \rho(g)x') = \rho_K(g)\Theta(x, x')\rho_K^\top(g)\,,$$
$$K(\rho(g)x, \rho(g)x') = \rho_K(g)K(x, x')\rho_K^\top(g)\,,$$

for all $g \in G$ and $x, x' \in X$.

Hence, for MLPs,

$$\Theta(\rho(g)x, \rho(g)x') = \Theta(x, x') \quad \Rightarrow \quad \Theta(\rho(g)x, x') = \Theta(x, \rho^{-1}(g)x')$$

Permutation shift

- On the training data, group transformations permute the samples

$$\rho(g)x_i = x_{\pi_g(i)}, \qquad \pi_g \in S_N$$

Permutation shift

- On the training data, group transformations permute the samples

$$\rho(g)x_i = x_{\pi_g(i)}, \qquad \pi_g \in S_N$$

- Therefore, for a permutation of training samples associate to g

$$\begin{aligned}\Pi(g)\Theta(X,X) &= \Theta(\rho(g)X,X) \\ &= \Theta(X,\rho^{-1}(g)X) \\ &= \Theta(X,X)(\Pi^{-1}(g))^\top \\ &= \Theta(X,X)\Pi(g)\end{aligned}$$

Data augmentation at infinite width

group transformation

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

augmented data

augmented labels

Data augmentation at infinite width

group transformation

for augmented data

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

augmented data

augmented labels

Data augmentation at infinite width

group transformation

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

augmented data

augmented labels

Data augmentation at infinite width

group transformation

augmented labels

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

for invariance

Data augmentation at infinite width

group transformation

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

for invariance

Mean prediction

$$\mu_t(x)$$

Mean prediction

$$\mu_t(x) = \mathbb{E}_{\theta_0 \sim \text{initializations}}[\mathcal{N}_{\theta_t}(x)]$$

Mean prediction

$$\mu_t(x) = \mathbb{E}_{\theta_0 \sim \text{initializations}}[\mathcal{N}_{\theta_t}(x)] = \lim_{n \to \infty} \frac{1}{n} \sum_{\theta_0 = \text{init}_1}^{\text{init}_n} \mathcal{N}_{\theta_t}(x)$$

Mean prediction

$$\mu_t(x) = \mathbb{E}_{\theta_0 \sim \text{initializations}}[\mathcal{N}_{\theta_t}(x)] = \lim_{n \to \infty} \underbrace{\frac{1}{n} \sum_{\theta_0 = \text{init}_1}^{\text{init}_n} \mathcal{N}_{\theta_t}(x)}_{\text{mean prediction of deep ensemble}}$$

Main conclusion

Deep ensembles trained with data augmentation are equivariant.

Main conclusion

Deep ensembles trained with data augmentation are equivariant.

✓ Proof of exact equivariance for
  - full data augmentation
  - infinite ensembles

Main conclusion

Deep ensembles trained with data augmentation are equivariant.

- ✓ Proof of exact equivariance for
  - full data augmentation
  - infinite ensembles
- ✓ Equivariance holds for all training times

Main conclusion

Deep ensembles trained with data augmentation are equivariant.

- ✓ Proof of exact equivariance for
  - full data augmentation
  - infinite ensembles
- ✓ Equivariance holds for all training times
- ✓ Equivariance holds away from the training data

Main conclusion

Deep ensembles trained with data augmentation are equivariant.

✓ Proof of exact equivariance for
  - full data augmentation
  - infinite ensembles

✓ Equivariance holds for all training times

✓ Equivariance holds away from the training data

✓ Holds also for finite-width networks

[Nordenfors, Flinth 2024]

Intuitive explanation

- ✓ Equivariance holds for all training times
- ✓ Equivariance holds away from the training data

Intuitive explanation

- ✓ Equivariance holds for all training times
- ✓ Equivariance holds away from the training data

ⓘ At infinite width, the mean output at initialization is zero everywhere.

Intuitive explanation

- ✓ Equivariance holds for all training times
- ✓ Equivariance holds away from the training data

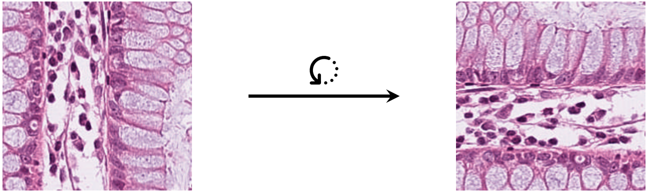
ⓘ At infinite width, the mean output at initialization is zero everywhere.

⇨ Training with full data augmentation leads to an equivariant function.

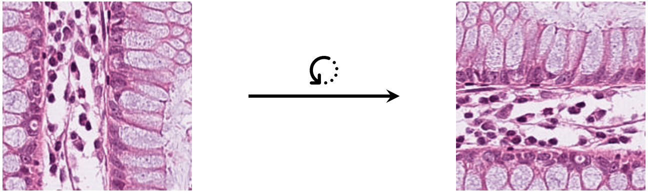
What Does An Augmented Ensemble Converge To?

Rotating images

Rotating images



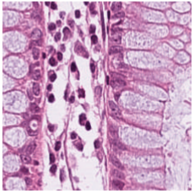
Rotating images

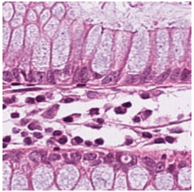


$f(x)$

f : pixels $\rightarrow$ colors

Rotating images





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

f : pixels $\to$ colors

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

Data augmentation and NTKs

Data augmentation and NTKs

Consider two ensembles:

trained without data augmentation

trained with data augmentation

Data augmentation and NTKs

Consider two ensembles:

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

Consider two ensembles:

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

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

at infinite width.

Data augmentation and NTKs

Consider two ensembles:

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.

Data augmentation and NTKs

Consider two ensembles:

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.

Data augmentation and NTKs

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

Data augmentation and NTKs

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

(!) Given an architecture with NTK Θ^{aug},
find an architecture with NTK $\Theta^{\text{non-aug}}$

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

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

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

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

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

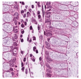
$$f' = \frac{1}{\mathrm{vol}(G)} \int_G \mathrm{d}g\, f(g) \qquad \text{group pooling}$$

GCNNs

Stack GConv-layers to obtain an invariant network

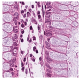
GCNNs

Stack GConv-layers to obtain an invariant network



GCNNs

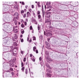
Stack GConv-layers to obtain an invariant network



→ lifting

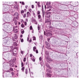
GCNNs

Stack GConv-layers to obtain an invariant network

 ⟶ lifting ⟶ σ

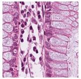
GCNNs

Stack GConv-layers to obtain an invariant network

 ⟶ lifting ⟶ σ ⟶ GConv

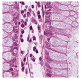
GCNNs

Stack GConv-layers to obtain an invariant network

 ⟶ lifting ⟶ σ ⟶ GConv ⟶ σ

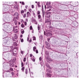
GCNNs

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

 ⟶ lifting ⟶ σ ⟶ GConv ⟶ σ ⟶ GConv ⟶ GPool

NTKs for GCNNs

For GCNN-layers, define the NNGP and NTK via

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

NTKs for GCNNs

For GCNN-layers, define the NNGP and NTK via

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

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

NTKs for GCNNs

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

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

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

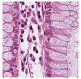
The layer-recursion for a GCNN-layer is given by

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

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

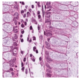
GCNNs

Stack GConv-layers to obtain an invariant network

 ⟶ lifting ⟶ σ ⟶ GConv ⟶ σ ⟶ GConv ⟶ GPool

NTKs for GCNNs

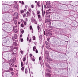
Stack GConv-layers to obtain an invariant network

 ⟶ lifting ⟶ σ ⟶ GConv ⟶ σ ⟶ GConv ⟶ GPool

Compute NTK with layer-wise recursion

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

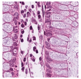
 $\longrightarrow$ lifting $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow \sigma \longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

0

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network



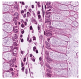
lifting → σ → GConv → σ → GConv → GPool

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

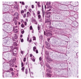
 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

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

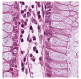
 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network



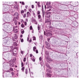
$\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

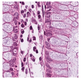
 $\longrightarrow$ lifting $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ σ $\longrightarrow$ GConv $\longrightarrow$ GPool

Compute NTK with layer-wise recursion

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

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

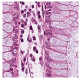
NTKs of MLPs and GCNNs

- Consider two neural networks

NTKs of MLPs and GCNNs

- Consider two neural networks

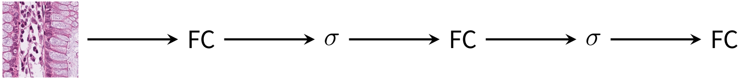
An MLP

 ⟶ FC ⟶ σ ⟶ FC ⟶ σ ⟶ FC

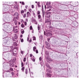
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



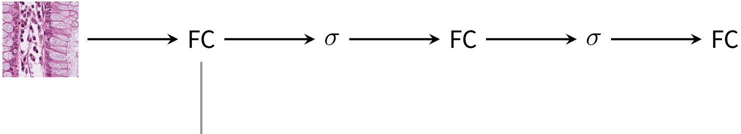
A GCNN



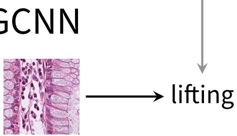
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN

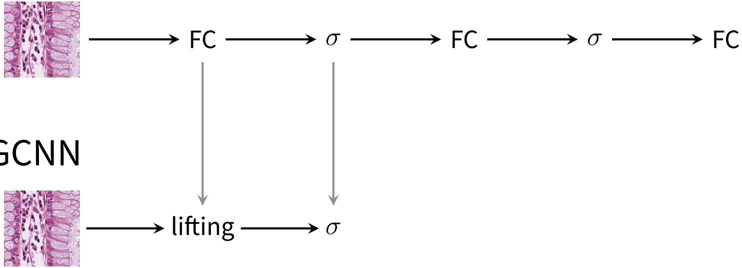


NTKs of MLPs and GCNNs

- Consider two neural networks

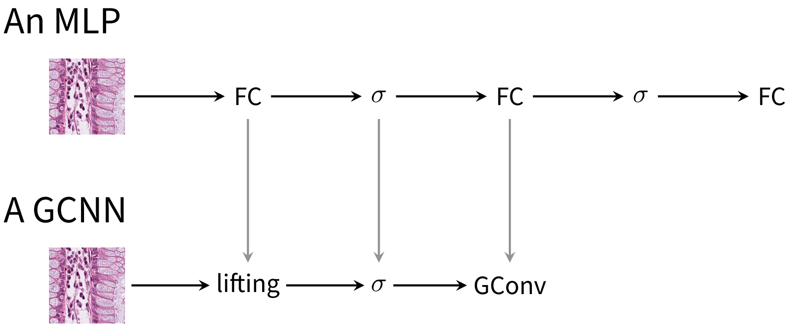
An MLP

A GCNN



NTKs of MLPs and GCNNs

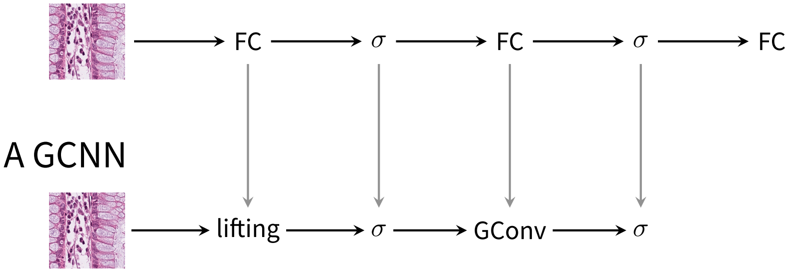
- Consider two neural networks



NTKs of MLPs and GCNNs

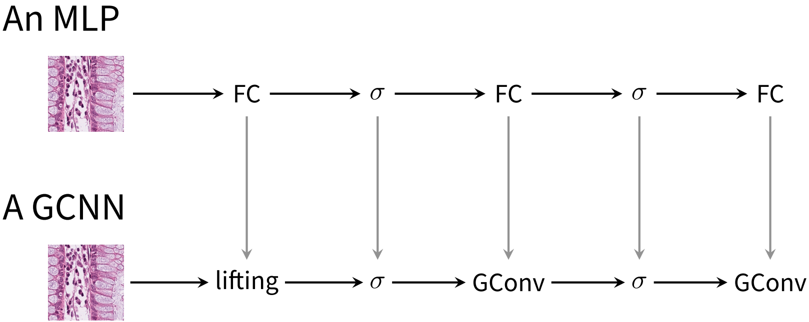
- Consider two neural networks

An MLP



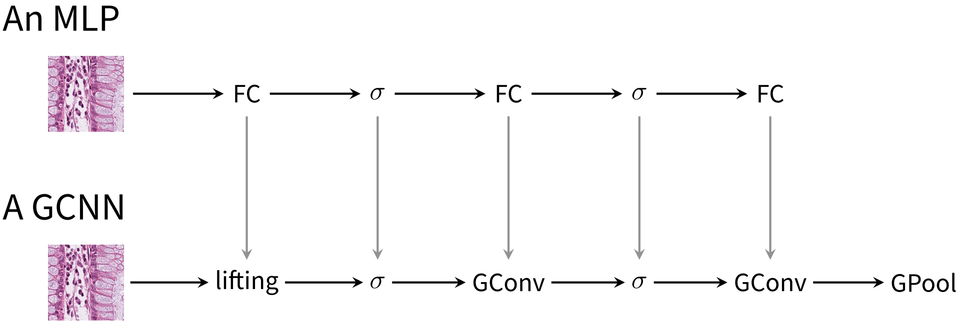
NTKs of MLPs and GCNNs

- Consider two neural networks



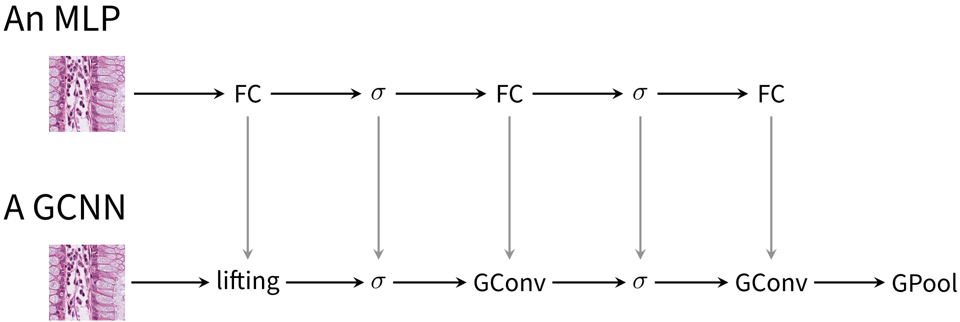
NTKs of MLPs and GCNNs

- Consider two neural networks



NTKs of MLPs and GCNNs

- Consider two neural networks



- Then

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

Data augmentation of MLPs

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

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 → Θ^{GCNN}; before: aug → Θ^{MLP}

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

⇨ 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 → Θ^{GCNN}; before: aug → Θ^{MLP}

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

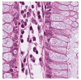
⇨ training the MLP on G-augmented data = training the GCNN on unaugmented data

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

Data augmentation of CNNs

Data augmentation of CNNs

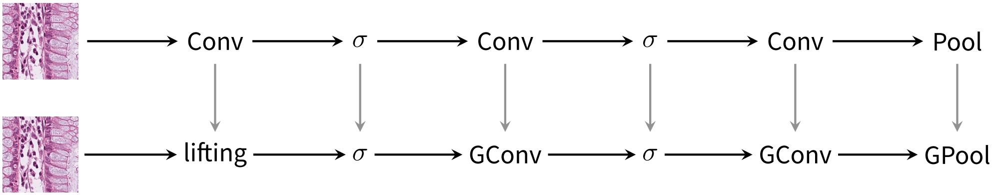
- Consider a CNN



→ Conv → σ → Conv → σ → Conv → Pool

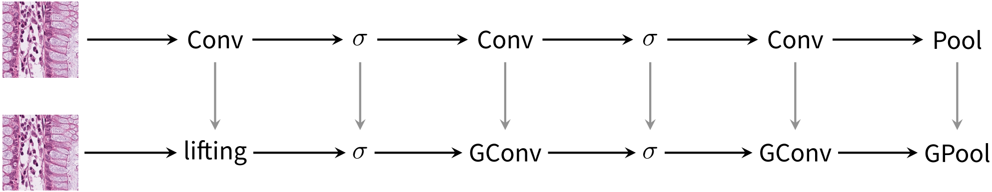
Data augmentation of CNNs

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



Data augmentation of CNNs

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

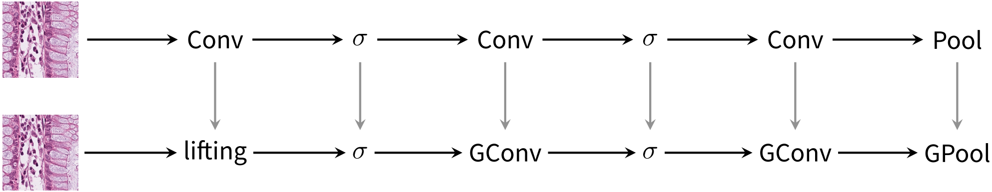


- Then

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

Data augmentation of CNNs

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



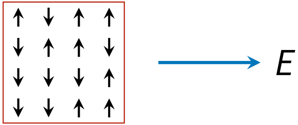
- Then

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

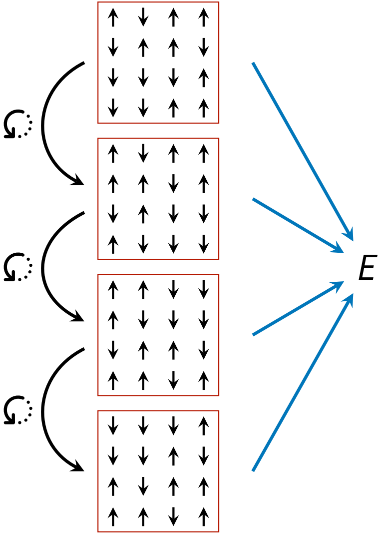
⇨ By training the CNN on rotated images, one obtains a roto-translation invariant GCNN

Experiments

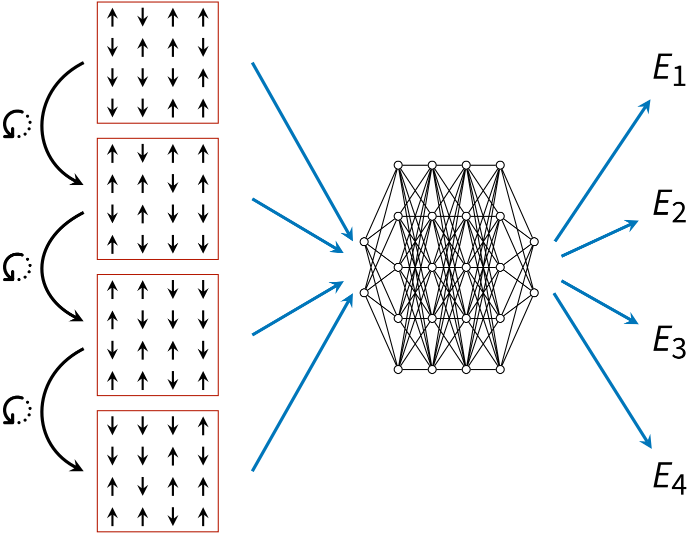
Ising model



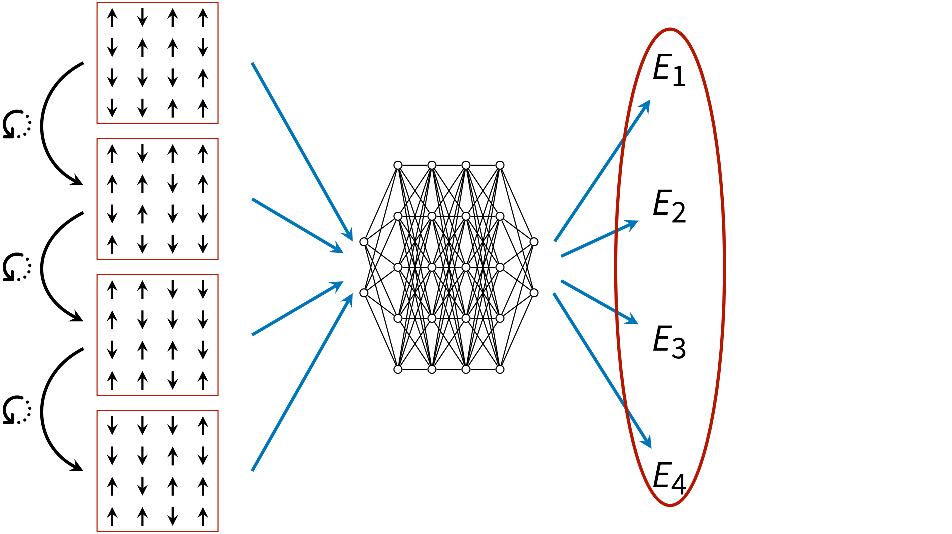
Ising model

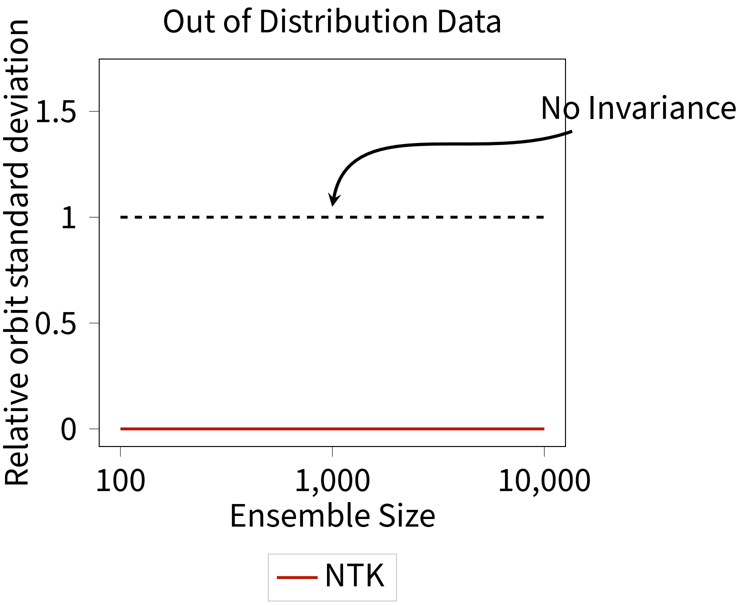


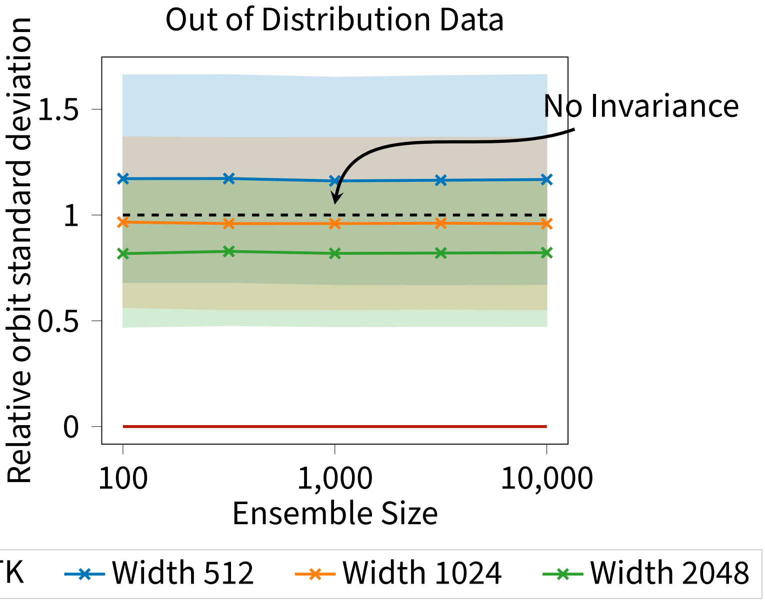
Ising model

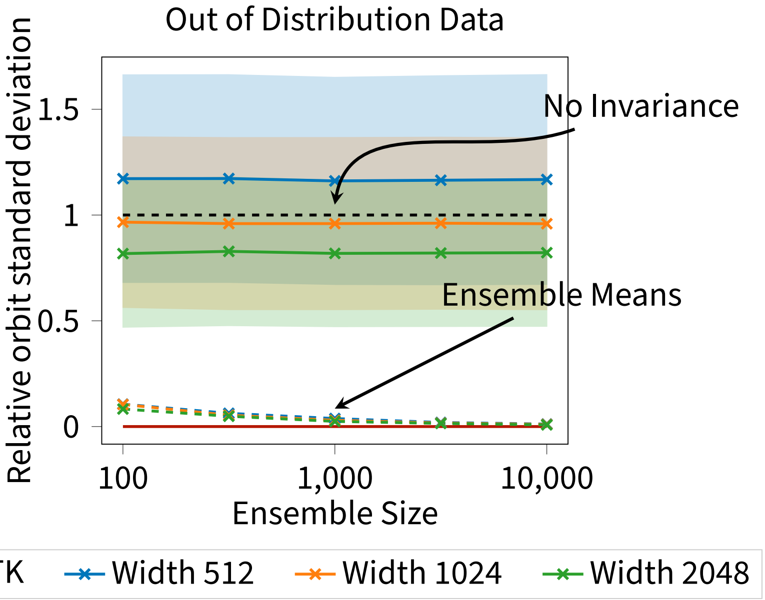


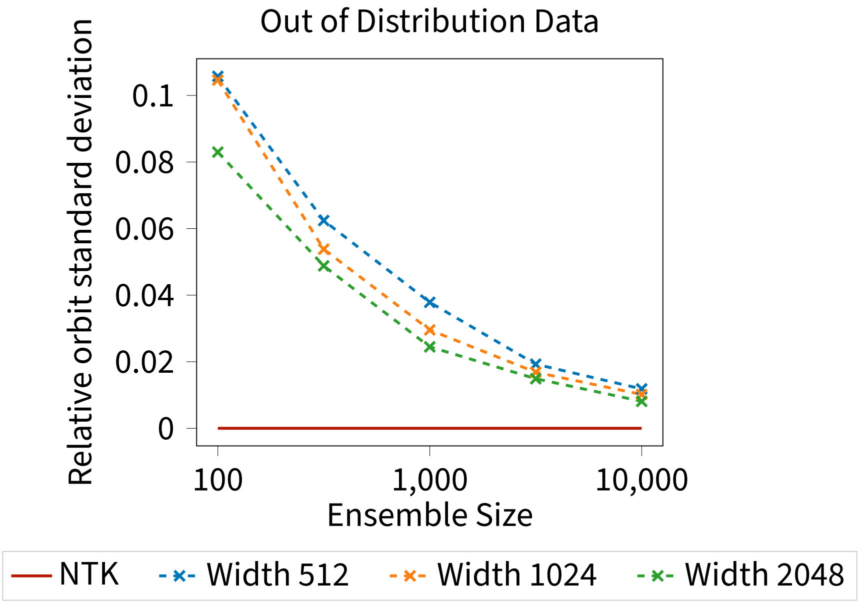
Ising model




Out of Distribution Data
No Invariance
Relative orbit standard deviation
1.5
1
0.5
0
100
1,000
10,000
Ensemble Size
NTK

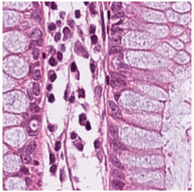

Out of Distribution Data
Relative orbit standard deviation
1.5
1
0.5
0
100
1,000
10,000
Ensemble Size
No Invariance
NTK
Width 512
Width 1024
Width 2048


Out of Distribution Data
Relative orbit standard deviation
1.5
1
0.5
0
No Invariance
Ensemble Means
100
1,000
10,000
Ensemble Size
NTK
Width 512
Width 1024
Width 2048


Out of Distribution Data
Relative orbit standard deviation
0.1
0.08
0.06
0.04
0.02
0
100
1,000
10,000
Ensemble Size
NTK
Width 512
Width 1024
Width 2048

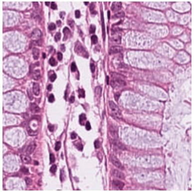
Histological slices

[Kather et al. 2018]



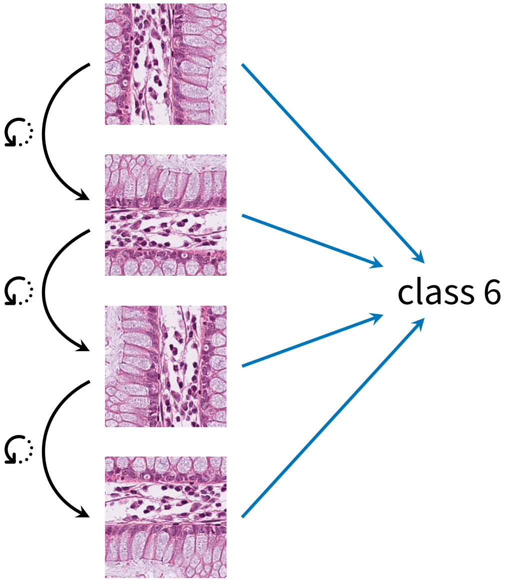
Histological slices

[Kather et al. 2018]

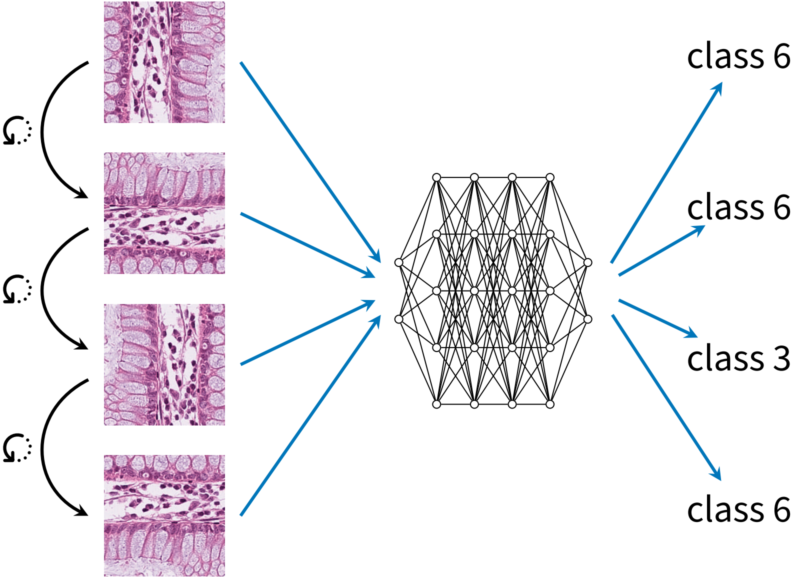


class 6

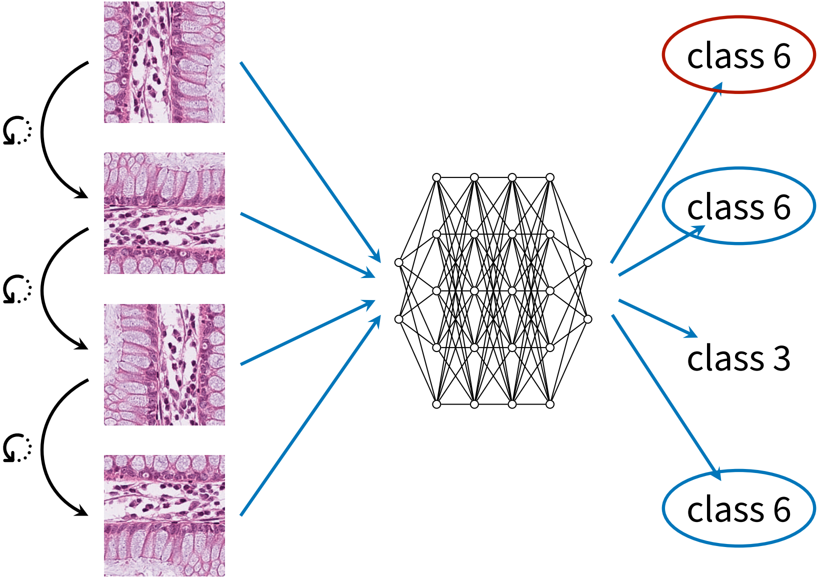
Histological slices



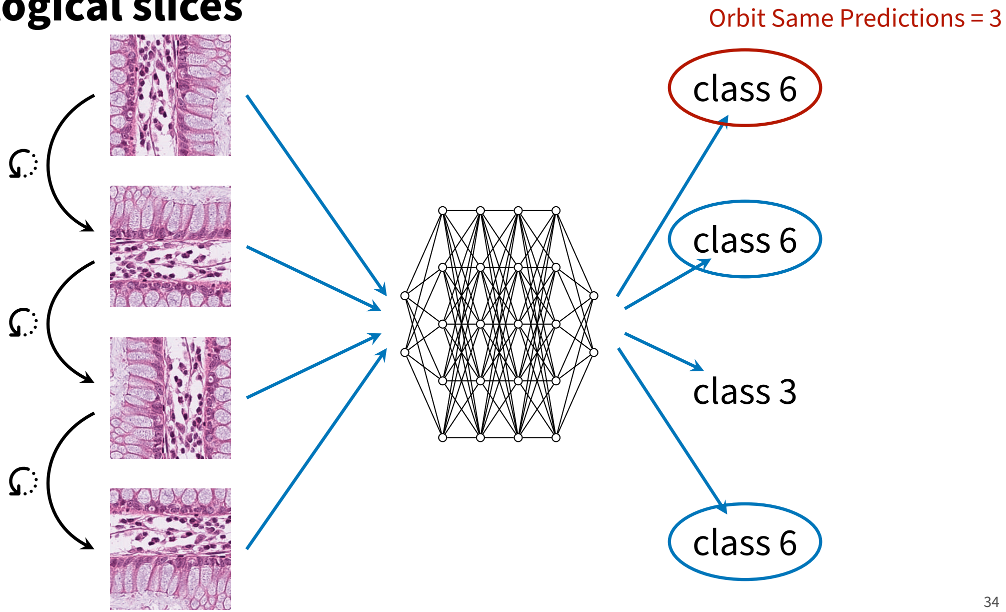
Histological slices



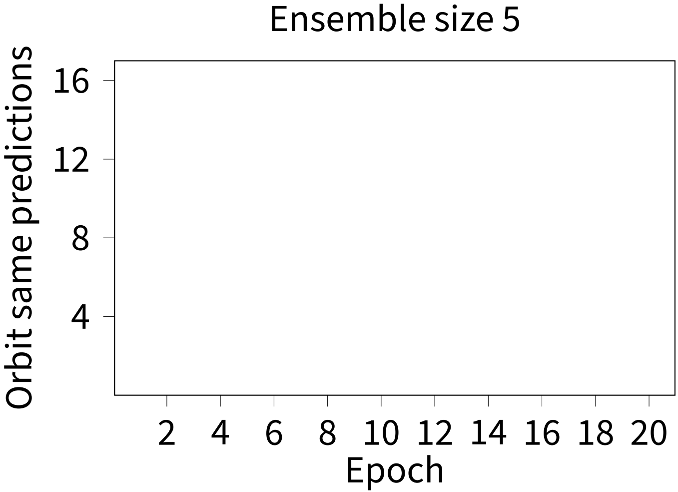
Histological slices



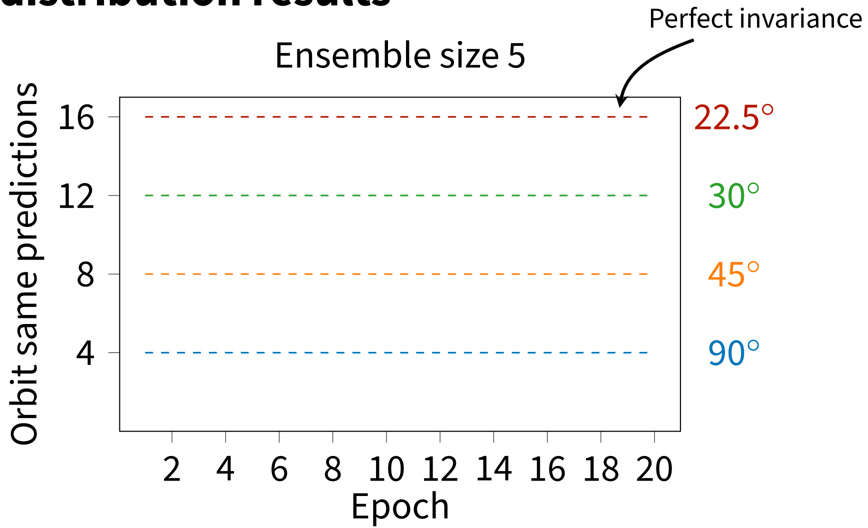
Histological slices



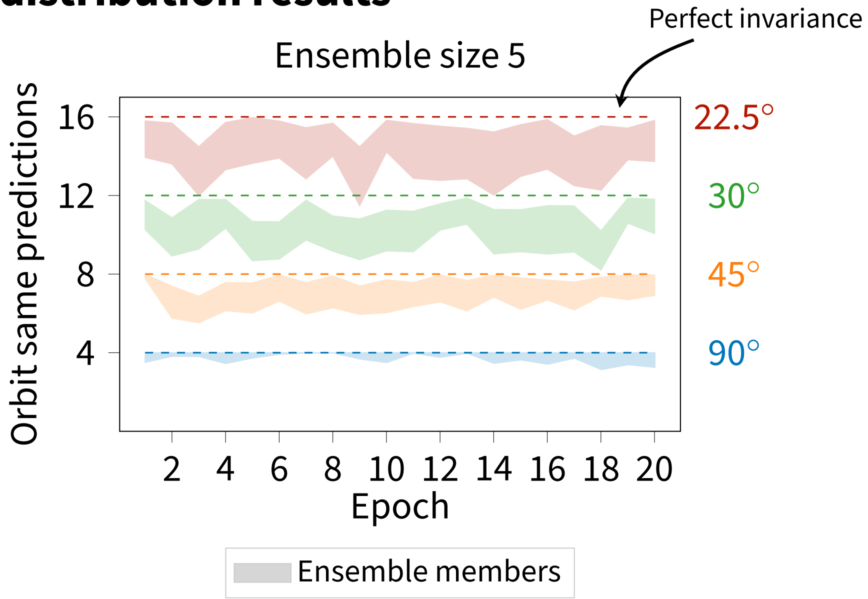
Out of distribution results



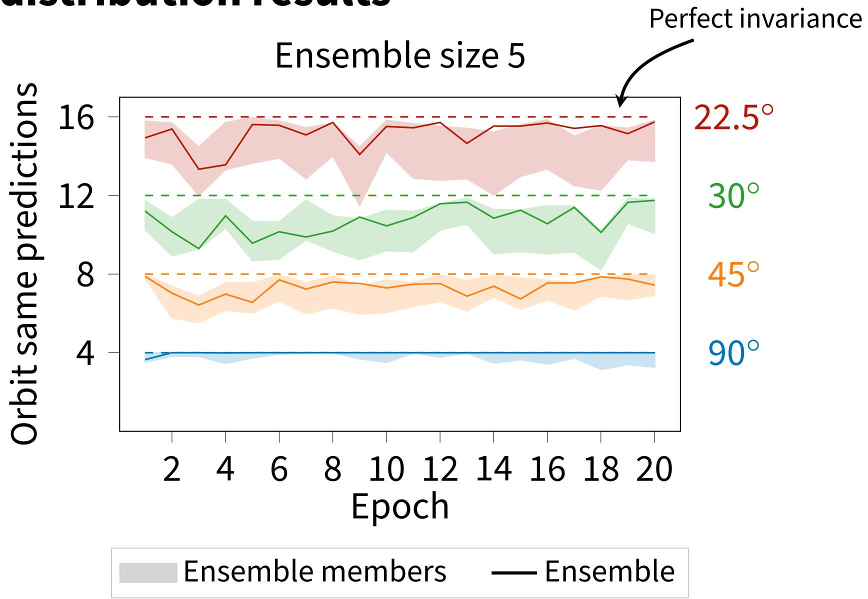
Out of distribution results



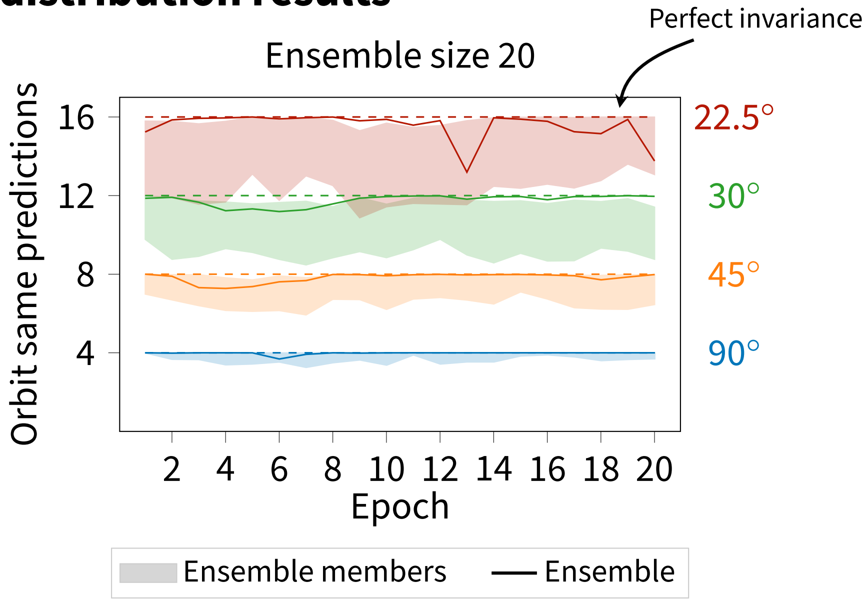
Out of distribution results



Out of distribution results



Out of distribution results



Further experimental results

Further experimental results

- ✓ Emergent invariance for rotated FashionMNIST

Further experimental results

- ✓ Emergent invariance for rotated FashionMNIST
- ✓ Partial augmentation for continuous symmetries

Further experimental results

- ✓ Emergent invariance for rotated FashionMNIST
- ✓ Partial augmentation for continuous symmetries
- ✓ Emergent equivariance (as opposed to invariance)

Comparison to other methods

Comparison to other methods

➪ Models trained on rotated FashionMNIST

Comparison to other methods

➪ Models trained on rotated FashionMNIST

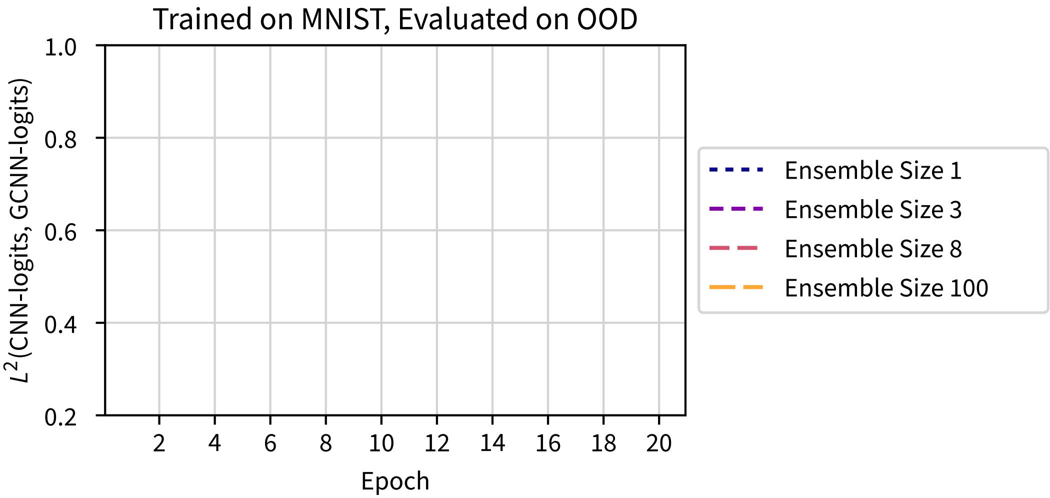
Orbit same predictions out of distribution:

	C_4	C_8	C_{16}
DeepEns+DA	3.85±0.12	**7.72±0.34**	**15.24±0.69**
only DA	3.41±0.18	6.73±0.24	12.77±0.71
E2CNN[1]	**4±0.0**	**7.71±0.21**	**15.08±0.34**
Canon[2]	**4±0.0**	**7.45±0.14**	12.41±0.85

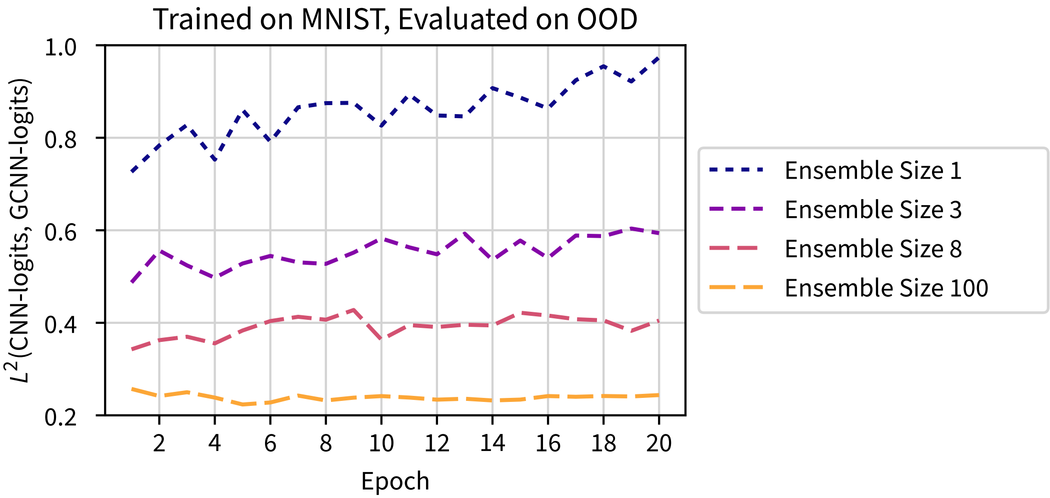
[1] [Weiler et al. 2019], [2] [Kaba et al. 2022]

Convergence of augmented CNNs to GCNNs

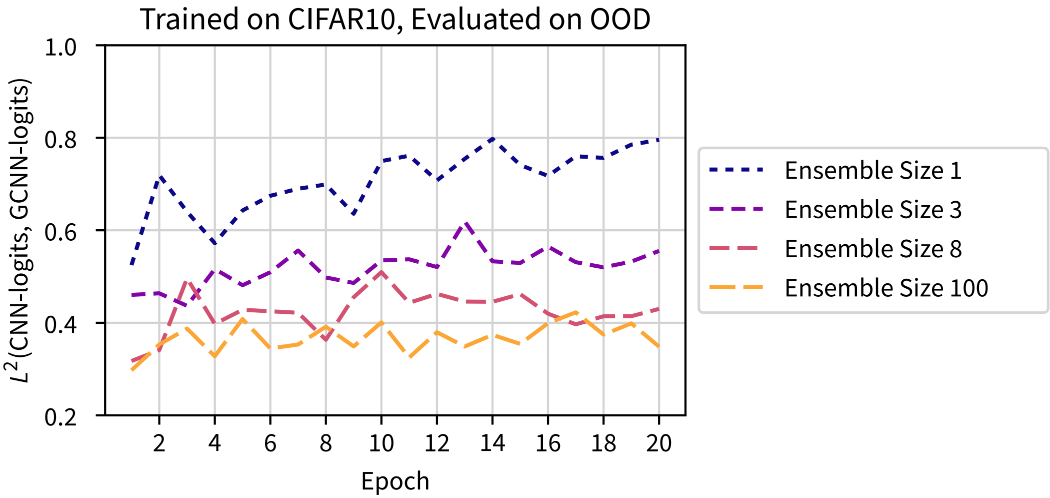
Convergence of augmented CNNs to GCNNs



Convergence of augmented CNNs to GCNNs



Convergence of augmented CNNs to GCNNs



Finite-width from Feynman diagrams

Restrictions of the infinite-width limit

The infinite-width limit...

- 👍 Yields compact expressions

- 👍 Is under complete analytic control

Restrictions of the infinite-width limit

The infinite-width limit...

- 👍 Yields compact expressions
- 👍 Is under complete analytic control
- 👎 Only Gaussian distributions
- 👎 Linearization of the model in the parameters
- 👎 No feature learning

Restrictions of the infinite-width limit

The infinite-width limit...

- 👍 Yields compact expressions
- 👍 Is under complete analytic control
- 👎 Only Gaussian distributions
- 👎 Linearization of the model in the parameters
- 👎 No feature learning

> ⓘ Use methods from physics to compute finite-width effects

Field theory

- In field theory, consider probability distribution over fields (functions)
- Typically: Gaussian probability distributions with small corrections

Field theory

- In field theory, consider probability distribution over fields (functions)
- Typically: Gaussian probability distributions with small corrections
- The Gaussian limit corresponds to free fields
- Corrections correspond to interactions between fields

Field theory

- In field theory, consider probability distribution over fields (functions)
- Typically: Gaussian probability distributions with small corrections
- The Gaussian limit corresponds to free fields
- Corrections correspond to interactions between fields
- Compute statistics of these fields

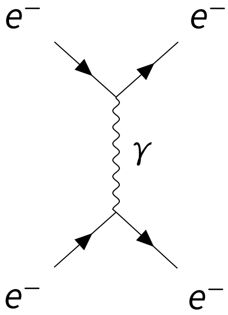
Feynman diagrams

Feynman diagrams are used to compute statistics

Feynman diagrams

Feynman diagrams are used to compute statistics

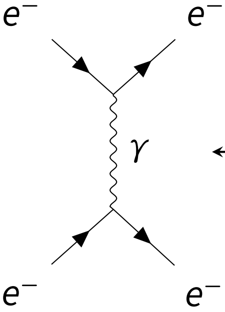
E.g. Electron-electron scattering



Feynman diagrams

Feynman diagrams are used to compute statistics

E.g. Electron-electron scattering



$$\longleftrightarrow \quad -ie^2 \frac{[\bar{u}(p_3)\gamma^\mu u(p_1)][\bar{u}(p_4)\gamma_\mu u(p_2)]}{(p_1 - p_3)^2}$$

Non-Gaussian Corrections from Physics

- Taylor-expand network statistics in 1/width
- Use Feynman diagrams to compute non-Gaussian corrections

Non-Gaussian Corrections from Physics

- Taylor-expand network statistics in 1/width
- Use Feynman diagrams to compute non-Gaussian corrections

Neural Networks	Field Theory
infinite width	no interactions
Gaussian distribution	free fields
finite-width	interactions

Tensors for neural network statistics

[Roberts, Yaida 2022]

Goal: Compute corrections for all neural network statistics at initialization, e.g.

Tensors for neural network statistics

[Roberts, Yaida 2022]

Goal: Compute corrections for all neural network statistics at initialization, e.g.

$$\mathbb{E}_\theta^c[z_{i_1}^{(\ell)}(x_1), z_{i_2}^{(\ell)}(x_2), \widehat{\Delta\Theta}_{i_3 i_4}^{(\ell)}(x_3, x_4)]$$

Tensors for neural network statistics

[Roberts, Yaida 2022]

Goal: Compute corrections for all neural network statistics at initialization, e.g.

cumulant → $\mathbb{E}^c_\theta$; preactivation → z; NTK fluctuation, $\widehat{\Delta\Theta}^{(\ell)} = \widehat{\Theta}^{(\ell)} - \mathbb{E}_\theta[\widehat{\Theta}^{(\ell)}]$

$$\mathbb{E}^c_\theta[z^{(\ell)}_{i_1}(x_1), z^{(\ell)}_{i_2}(x_2), \widehat{\Delta\Theta}^{(\ell)}_{i_3 i_4}(x_3, x_4)]$$

Tensors for neural network statistics

[Roberts, Yaida 2022]

Goal: Compute corrections for all neural network statistics at initialization, e.g.

cumulant → $\mathbb{E}^c_\theta$; preactivation → z; NTK fluctuation, $\widehat{\Delta\Theta}^{(\ell)} = \widehat{\Theta}^{(\ell)} - \mathbb{E}_\theta[\widehat{\Theta}^{(\ell)}]$

$$\mathbb{E}^c_\theta[z^{(\ell)}_{i_1}(x_1), z^{(\ell)}_{i_2}(x_2), \widehat{\Delta\Theta}^{(\ell)}_{i_3 i_4}(x_3, x_4)]$$

Decompose these into tensors

$$= \frac{1}{n}\left(D^{(\ell)}_{1234}\delta_{i_1 i_2}\delta_{i_3 i_4} + F^{(\ell)}_{1324}\delta_{i_1 i_3}\delta_{i_2 i_4} + F^{(\ell)}_{1423}\delta_{i_1 i_4}\delta_{i_2 i_3}\right)$$

Tensors for neural network statistics

[Roberts, Yaida 2022]

Goal: Compute corrections for all neural network statistics at initialization, e.g.

cumulant; preactivation; NTK fluctuation, $\widehat{\Delta\Theta}^{(\ell)} = \widehat{\Theta}^{(\ell)} - \mathbb{E}_\theta[\widehat{\Theta}^{(\ell)}]$

$$\mathbb{E}^c_\theta[z^{(\ell)}_{i_1}(x_1), z^{(\ell)}_{i_2}(x_2), \widehat{\Delta\Theta}^{(\ell)}_{i_3 i_4}(x_3, x_4)]$$

Decompose these into tensors

$$= \frac{1}{n}\left(D^{(\ell)}_{1234}\delta_{i_1 i_2}\delta_{i_3 i_4} + F^{(\ell)}_{1324}\delta_{i_1 i_3}\delta_{i_2 i_4} + F^{(\ell)}_{1423}\delta_{i_1 i_4}\delta_{i_2 i_3}\right)$$

Gram tensor, $F^{(\ell)}_{1324} = F^{(\ell)}(x_1, x_3, x_2, x_4)$

Tensors for neural network statistics

[Roberts, Yaida 2022]

⇨ Keeping track of these tensors is sufficient.

Tensors for neural network statistics

[Roberts, Yaida 2022]

⇨ Keeping track of these tensors is sufficient.

At order $1/n$, we have

Tensors for neural network statistics

[Roberts, Yaida 2022]

⇨ Keeping track of these tensors is sufficient.

At order $1/n$, we have

- $K, K^{\{1\}}$ and V_4 for preactivations

Tensors for neural network statistics

[Roberts, Yaida 2022]

⇨ Keeping track of these tensors is sufficient.

At order $1/n$, we have

- $K, K^{\{1\}}$ and V_4 for preactivations
- $\Theta, \Theta^{\{1\}}, A$ and B for derivatives

Tensors for neural network statistics

[Roberts, Yaida 2022]

⇨ Keeping track of these tensors is sufficient.

At order $1/n$, we have

- $K, K^{\{1\}}$ and V_4 for preactivations
- $\Theta, \Theta^{\{1\}}, A$ and B for derivatives
- D and F for mixed statistics

Tensors for neural network statistics

[Roberts, Yaida 2022]

⇨ Keeping track of these tensors is sufficient.

At order $1/n$, we have

- $K, K^{\{1\}}$ and V_4 for preactivations
- $\Theta, \Theta^{\{1\}}, A$ and B for derivatives
- D and F for mixed statistics
- 6 more for higher derivatives, needed for training

Tensor recursions

For each tensor, there is a layer-wise recursion relation, e.g.

Tensor recursions

For each tensor, there is a layer-wise recursion relation, e.g.

$$F^{(\ell+1)}_{1324} = \langle \sigma^{(\ell)}_1 \sigma^{(\ell)}_2 \sigma'^{(\ell)}_3 \sigma'^{(\ell)}_4 \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34}$$

$$+ \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma^{(\ell)}_1 \sigma'^{(\ell)}_3 z^{(\ell)}_\alpha \rangle_{K^{(\ell)}} \langle \sigma^{(\ell)}_2 \sigma'^{(\ell)}_4 z^{(\ell)}_\beta \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4}$$

Tensor recursions

For each tensor, there is a layer-wise recursion relation, e.g.

$\sigma(z^{(\ell)}(x_1))$ — Gaussian expectation with cov. $K^{(\ell)}$ — Components of $(K^{(\ell)})^{-1}$

$$F^{(\ell+1)}_{1324} = \langle \sigma^{(\ell)}_1 \sigma^{(\ell)}_2 \sigma'^{(\ell)}_3 \sigma'^{(\ell)}_4 \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34}$$

$$+ \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma^{(\ell)}_1 \sigma'^{(\ell)}_3 z^{(\ell)}_\alpha \rangle_{K^{(\ell)}} \langle \sigma^{(\ell)}_2 \sigma'^{(\ell)}_4 z^{(\ell)}_\beta \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4}$$

Tensor recursions

For each tensor, there is a layer-wise recursion relation, e.g.

$\sigma(z^{(\ell)}(x_1))$

Gaussian expectation with cov. $K^{(\ell)}$

Components of $(K^{(\ell)})^{-1}$

$$F^{(\ell+1)}_{1324} = \langle \sigma^{(\ell)}_1 \sigma^{(\ell)}_2 \sigma'^{(\ell)}_3 \sigma'^{(\ell)}_4 \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34} + \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma^{(\ell)}_1 \sigma'^{(\ell)}_3 z^{(\ell)}_\alpha \rangle_{K^{(\ell)}} \langle \sigma^{(\ell)}_2 \sigma'^{(\ell)}_4 z^{(\ell)}_\beta \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4}$$

Solving this system of recursions yields the complete network statistics at order $1/n$.

Tensor recursions

For each tensor, there is a layer-wise recursion relation, e.g.

$\sigma(z^{(\ell)}(x_1))$

Gaussian expectation with cov. $K^{(\ell)}$

Components of $(K^{(\ell)})^{-1}$

$$F^{(\ell+1)}_{1324} = \langle \sigma^{(\ell)}_1 \sigma^{(\ell)}_2 \sigma'^{(\ell)}_3 \sigma'^{(\ell)}_4 \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34} + \sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma^{(\ell)}_1 \sigma'^{(\ell)}_3 z^{(\ell)}_\alpha \rangle_{K^{(\ell)}} \langle \sigma^{(\ell)}_2 \sigma'^{(\ell)}_4 z^{(\ell)}_\beta \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4}$$

Solving this system of recursions yields the complete network statistics at order $1/n$.

> ⓘ Computing these recursions analytically is very laborious

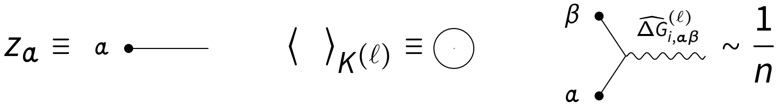
Feynman diagrams

Use Feynman diagrams to compute these recursions.

Feynman diagrams

Use Feynman diagrams to compute these recursions.

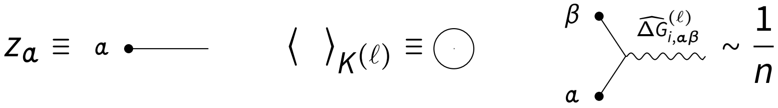
- For preactivations, can read off Feynman rules from NN probability distribution, e.g. [Banta et al. 2024]



Feynman diagrams

Use Feynman diagrams to compute these recursions.

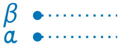
- For preactivations, can read off Feynman rules from NN probability distribution, e.g. [Banta et al. 2024]

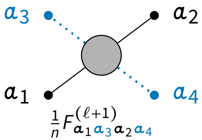


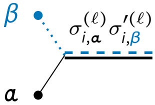
- For derivatives, need to find Feynman rules by inspecting analytic expressions

Feynman rules relevant for the F-tensor

$\widehat{\Delta\Theta}_{\alpha\beta} \equiv$



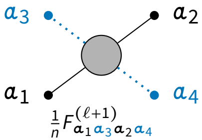


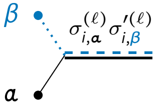


$\sim \dfrac{1}{n}$

Feynman rules relevant for the F-tensor

$\widehat{\Delta\Theta}_{\alpha\beta} \equiv$ 

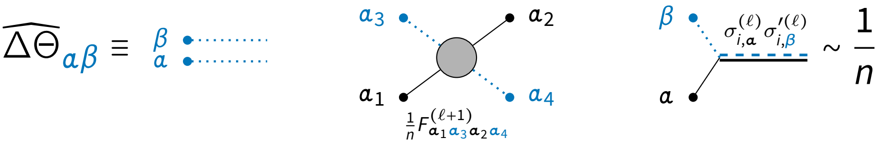




$\sim \frac{1}{n}$

The propagator  satisfies selection rules, e.g.

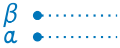
Feynman rules relevant for the F-tensor

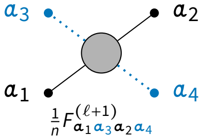


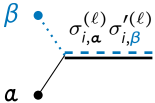
The propagator ○ satisfies selection rules, e.g.

- It cannot be directly connected to other propagators

Feynman rules relevant for the F-tensor

$\widehat{\Delta\Theta}_{\alpha\beta} \equiv$ 



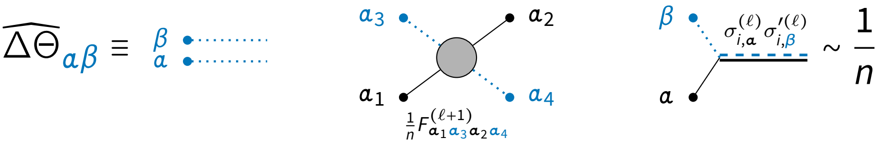


$\sim \dfrac{1}{n}$

The propagator  satisfies selection rules, e.g.

- It cannot be directly connected to other propagators
- Dotted lines attached to a propagator do not appear in the Gaussian expectation value

Feynman rules relevant for the F-tensor

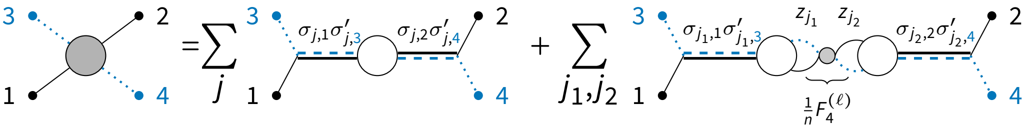


The propagator ◯ satisfies selection rules, e.g.

- It cannot be directly connected to other propagators
- Dotted lines attached to a propagator do not appear in the Gaussian expectation value
- Pairs of dashed lines of the same color connected to the propagator add a factor of Θ

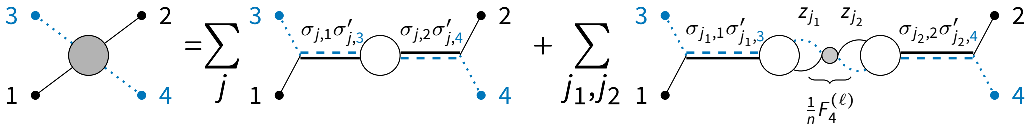
The F-recursion

Draw all diagrams possible for the F-tensor at order $1/n$



The F-recursion

Draw all diagrams possible for the F-tensor at order $1/n$

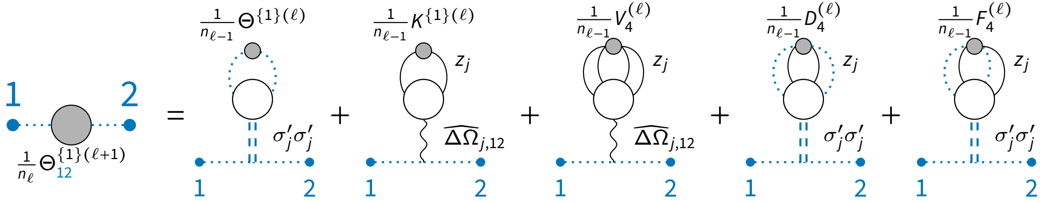


Compare to the analytical expression

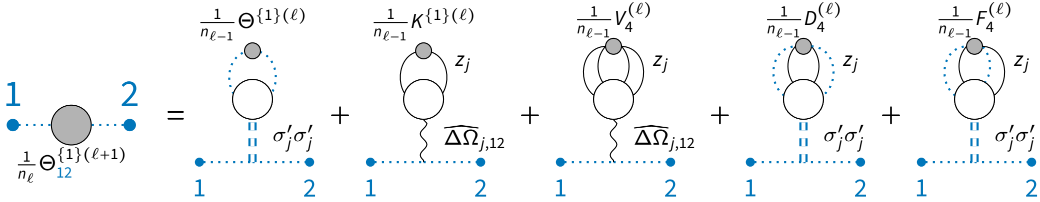
$$\frac{1}{n}F^{(\ell+1)}_{1324} = \frac{1}{n}\langle \sigma^{(\ell)}_1 \sigma^{(\ell)}_2 \sigma'^{(\ell)}_3 \sigma'^{(\ell)}_4 \rangle_{K^{(\ell)}} \Theta^{(\ell)}_{34}$$

$$+ \frac{1}{n}\sum_{\alpha,\beta,\gamma,\delta=1}^{4} \langle \sigma^{(\ell)}_1 \sigma'^{(\ell)}_3 z^{(\ell)}_\alpha \rangle_{K^{(\ell)}} \langle \sigma^{(\ell)}_2 \sigma'^{(\ell)}_4 z^{(\ell)}_\beta \rangle_{K^{(\ell)}} K^{\alpha\gamma}_{(\ell)} K^{\beta\delta}_{(\ell)} F^{(\ell)}_{\gamma 3 \delta 4}$$

The NTK recursion at finite width

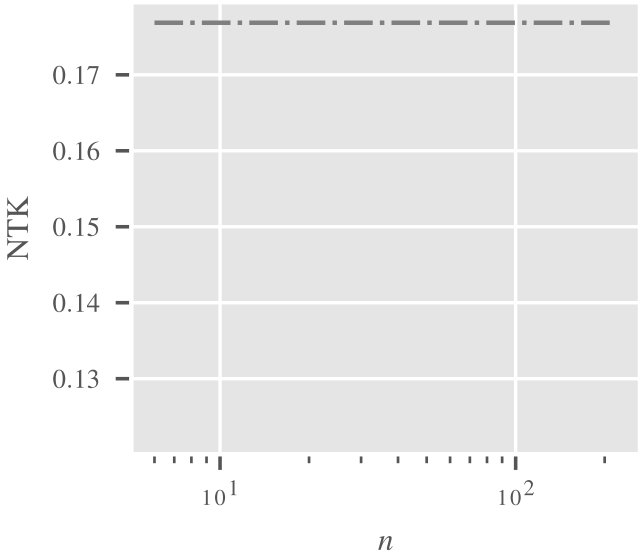


The NTK recursion at finite width

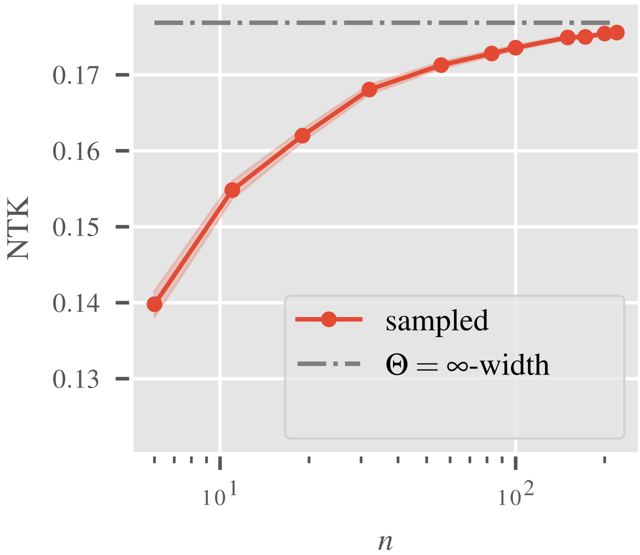


ⓘ Feynman diagrams allow for much simpler derivation of recursion relations

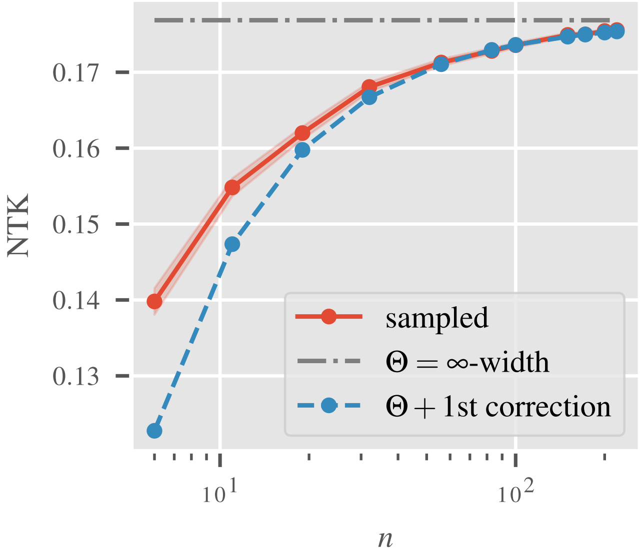
Numerical results



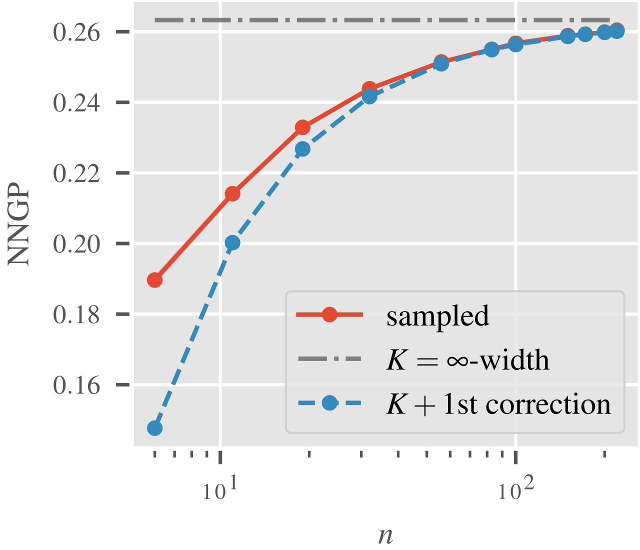
Numerical results



Numerical results



Numerical results



Key takeaways

Key takeaways

- Neural tangent kernels at infinite width can be used to understand data augmentation of deep ensembles:

Key takeaways

- Neural tangent kernels at infinite width can be used to understand data augmentation of deep ensembles:
    - Deep ensembles become exactly equivariant

Key takeaways

- Neural tangent kernels at infinite width can be used to understand data augmentation of deep ensembles:
  - Deep ensembles become exactly equivariant
  - Deep ensembles trained with data augmentation are group convolutional networks

Key takeaways

- Neural tangent kernels at infinite width can be used to understand data augmentation of deep ensembles:
    - Deep ensembles become exactly equivariant
    - Deep ensembles trained with data augmentation are group convolutional networks

- To consider non-Gaussian corrections away from infinite width, consider 1/width expansion

Key takeaways

- Neural tangent kernels at infinite width can be used to understand data augmentation of deep ensembles:
  - Deep ensembles become exactly equivariant
  - Deep ensembles trained with data augmentation are group convolutional networks

- To consider non-Gaussian corrections away from infinite width, consider 1/width expansion

- The corrections can be computed conveniently using Feynman diagrams

Papers

- Emergent Equivariance in Deep Ensembles
  Jan E. Gerken*, Pan Kessel*
  ICML 2024 (Oral)
- Equivariant Neural Tangent Kernels
  Philipp Misof, Pan Kessel, Jan E. Gerken
  ICML 2025
- Finite-Width Neural Tangent Kernels from Feynman Diagrams
  Max Guillen*, Philipp Misof*, Jan E. Gerken
  arXiv: 2508.11522

* Equal contribution

Thank you!