

Neural Tangent Kernels:

Data augmentation and Feynman diagrams

Jan E. Gerken



UNIVERSITY OF
GOTHENBURG

WASP | WALLENBERG AI
AUTONOMOUS SYSTEMS
AND SOFTWARE PROGRAM

in collaboration with



Pan Kessel



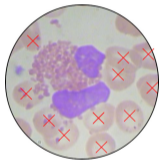
Philipp Misof



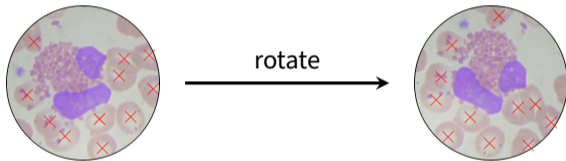
Max Guillen

Symmetries in deep learning

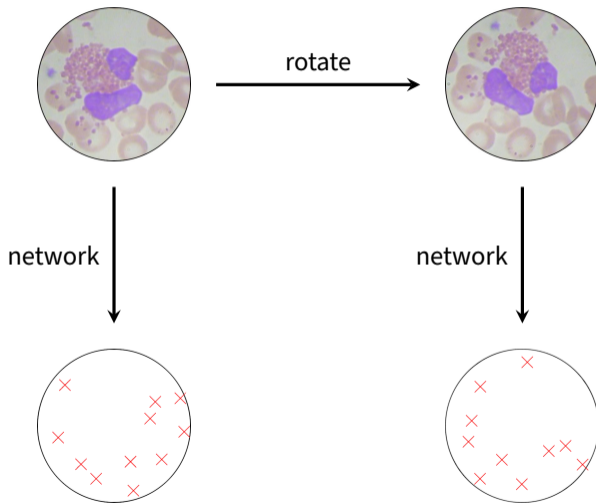
Symmetries in deep learning



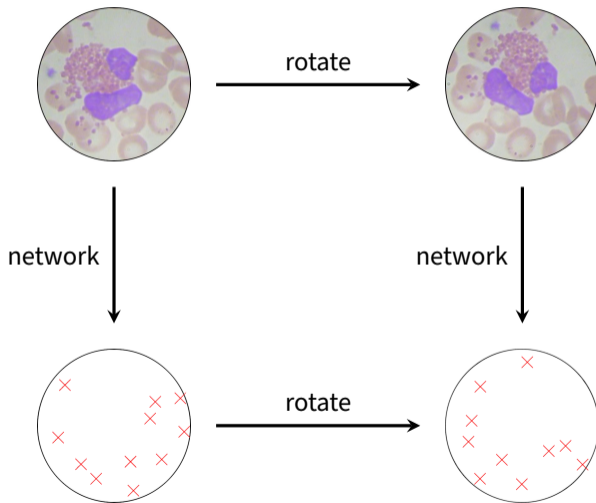
Symmetries in deep learning



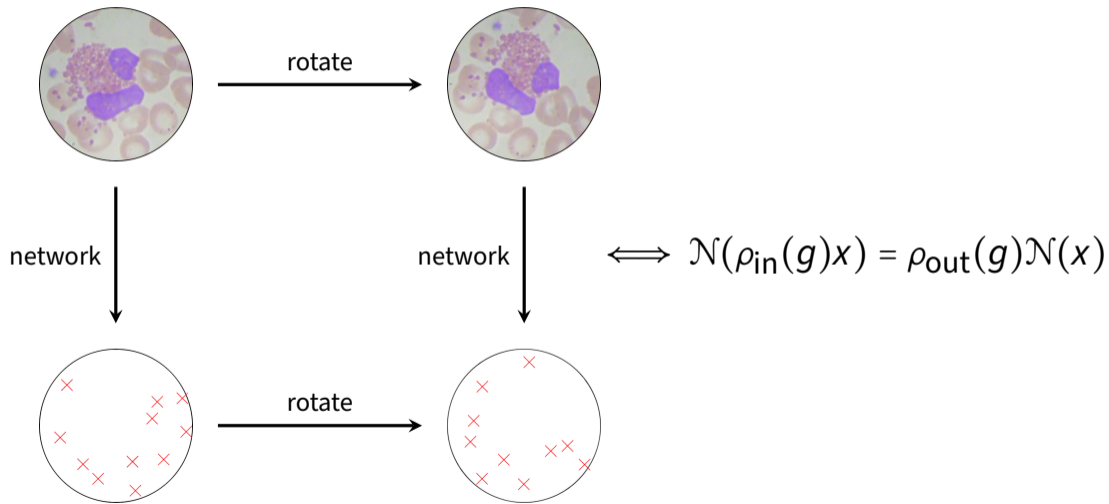
Symmetries in deep learning



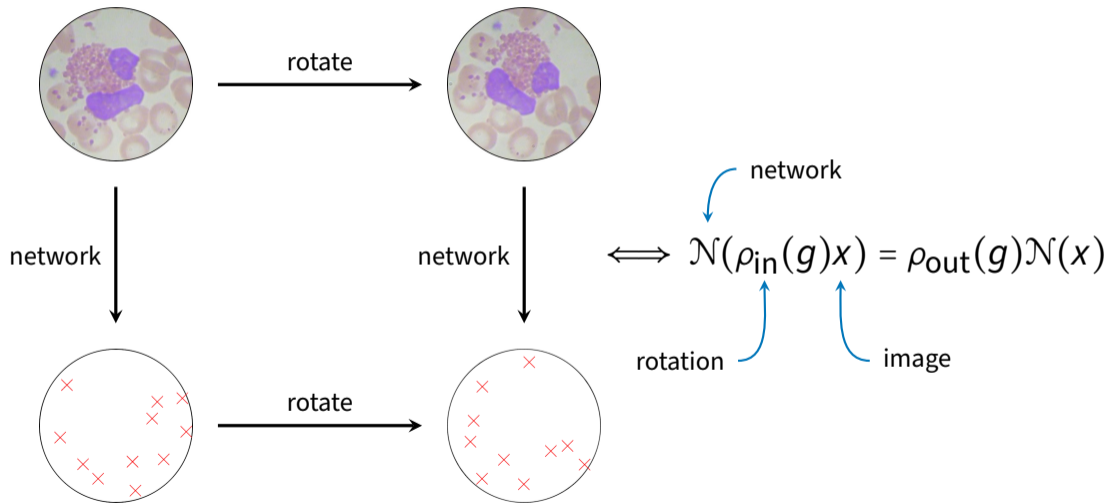
Symmetries in deep learning



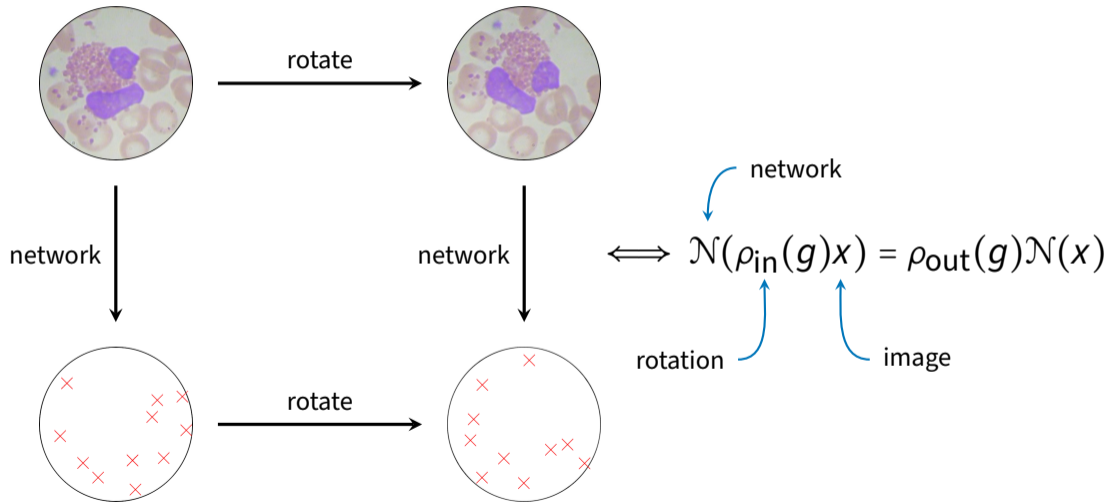
Symmetries in deep learning



Symmetries in deep learning



Equivariance



Equivariant neural networks

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen

University of Amsterdam

Max Welling

University of Amsterdam

University of California Irvine

Canadian Institute for Advanced Research

T.S.COHEN@UVA.NL

M.WELLING@UVA.NL

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
University of Amsterdam
Max Welling
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

T.S.COHEN@UVA.NL

M.WELLING@UVA.NL

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 Tai¹ Peter Bailis¹ Gregory Valiant¹

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 & Veličković, 2017; Cohen et al., 2018) has explored new CNN architectures that are *invariant to certain modifiable to particular transforms*.

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen
University of Amsterdam
Max Welling
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

T.S.COEN@UVA.NL

M.WELLING@UVA.NL

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 Tai¹ Peter Bailis¹ Gregory Valiant¹

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 & Veličković, 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 Bucci,^{1,5} Michael Ragone,^{1,6} Patrick J. Cules,¹ Frédéric Sauvage,¹ Martin Laroche,^{1,7} and M. Cerezo¹

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

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

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen
University of Amsterdam

T.S.COHEM@UVA.NL

Max Welling
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

M.WELLING@UVA.NL

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 Tai¹ Peter Bailis¹ Gregory Valiant¹

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 & Veličković, 2017; Cohen et al., 2018) has explored new CNN architectures that are invariant to various non-linearly to particular transforms.

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. Cules,¹ Frédéric Sauvage,¹ Martin Laroche,^{1,7} and M. Cerezo¹

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

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

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong^{a,b,c,1} Qi Meng^b Jue Zhang^b Hulin Qu^c Congqiao Li² Sitian Qian^d Weitao Du^a Zhi-Ming Ma^a Tie-Yan Liu^b

^aAcademy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road, Beijing 100190, China

^bMicrosoft Research Asia, Danling Street, Beijing 100080, China

^cCERN, EP Department, CH-1211 Geneva 23, Switzerland

^dSchool of Physics, Peking University, Chenkefu Road, Beizhou 100871, China

Equivariant neural networks

Group Equivariant Convolutional Networks

Taco S. Cohen
University of Amsterdam

T.S.COHEN@UVA.NL

Max Welling
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

M.WELLING@UVA.NL

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

Kal Sheng Tai¹ Peter Bailis¹ Gregory Valiant¹

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 & Veličković, 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. Cules,¹ Frédéric Sauvage,¹ Martin Larooca,^{1,7} and M. Cerezo¹

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

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

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong^{a,b,1} Qi Meng^b Jue Zhang^b Hulin Qu^c Congqiao Li² Sitian Qian^d Weitao Du^a Zhi-Ming Ma^a Tie-Yan Liu^b

^aAcademy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road, Beijing 100190, China

^bMicrosoft Research Asia, Danling Street, Beijing 100080, China

^cCERN, EP Department, CH-1211 Geneva 23, Switzerland

^dSchool of Physics, Peking University, Cheneba Road, Beizina 100871, China

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

Simon Batzner^{a,1} Albert Muscelian,¹ Litxin Sun,¹ Mario Geiger,² Jonathan P. Mailon,³ Mordechai Korbath,² Nicola Molinari,¹ Tess E. Smidt,^{4,5} and Boris Kozinsky^{a,1,2}

¹John A. Pashon School of Engineering and Applied Sciences,

Harvard University, Cambridge, MA 02138, USA

²École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

³Robert Bosch Research and Technology Center, Cambridge, MA 02139, USA

⁴Computational Research Division and Center for Advanced Mathematics for Energy Research Applications, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

⁵Massachusetts 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 informative-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
University of Amsterdam

T.S.COHEN@UVA.NL

Max Welling
University of Amsterdam
University of California Irvine
Canadian Institute for Advanced Research

M.WELLING@UVA.NL

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

Kal Sheng Tai¹ Peter Bailis¹ Gregory Valiant¹

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 & Veličković, 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. Cules,¹ Frédéric Sauvage,¹ Martin Lucocca,^{1,7} and M. Cerezo¹

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

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

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

Shiqi Gong^{a,b,1} Qi Meng^b Jue Zhang^b Hulin Qu^b Gonggao Li² Sitian Qian¹ Weitao Du^b Zhi-Ming Ma^a Tie-Yan Liu^b

^aAcademy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancun East Road, Beijing 100190, China

^bMicrosoft Research Asia, Danling Street, Beijing 100080, China

^cCERN, EP Department, CH-1211 Geneva 23, Switzerland

^dSchool of Physics, Peking University, Cheneiya Road, Beizhen 100871, China

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

Simon Batzner^{a,1} Albert Muscelian,¹ Lixin Sun,¹ Mario Geiger,² Jonathan P. Mailon,³ Mordechai Kohnbluth,² Nicola Molinari,¹ Tess E. Smidt,^{4,5} and Boris Kozinsky^{a,1,2}

¹John A. Pashen School of Engineering and Applied Sciences,

Harvard University, Cambridge, MA 02138, USA

²École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

³Robert Bosch Research and Technology Center, Cambridge, MA 02139, USA

⁴Computational Research Division and Center for Advanced Mathematics for Energy Research Applications,

Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

⁵Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA 02139, 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^a
Department of Applied Physics
Stanford University
seismann@stanford.edu

Raphael J.L. Townsend^b
Department of Computer Science
Stanford University
raphael@cs.stanford.edu

Nathaniel Thomas^c
Department of Physics
Stanford University
nthomas103@gmail.com

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

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

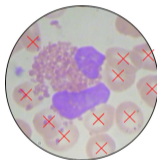
Ron O. Dror
Department of Computer Science
Stanford University
rondror@cs.stanford.edu

ABSTRACT

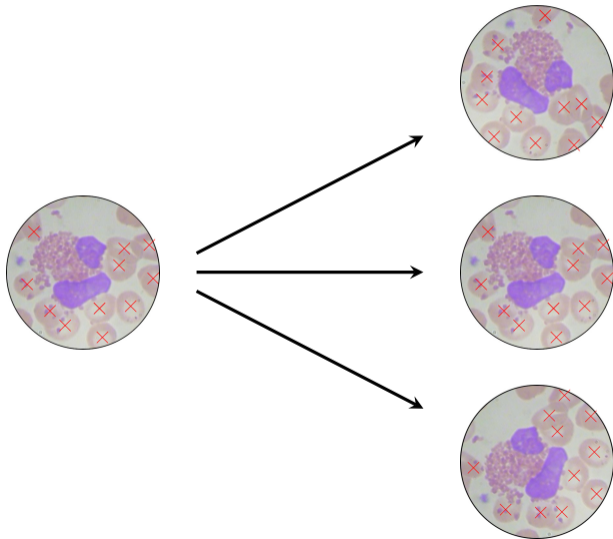
Predicting the structure of multi-protein complexes is a grand challenge in biochemistry, with major implications for basic science and drug discovery. Computational structure prediction methods generally leverage pre-defined structural features to distinguish accurate structural models from less accurate 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/s41586-024-03487-y>

Received: 18 December 2022

Accepted: 29 April 2024

Published online: 8 May 2024

Open access

 Check for updates

Josh Abramson^{1,2}, Anna Adler^{1,2}, Jack Dongus^{1,2}, Richard Evans^{1,2}, Tim Green^{1,2}, Alexander Pritzel^{1,2}, Olaf Ronneberger^{1,2}, Lindsay Wilkerson^{1,2}, Andrew J. Ballard¹, Joshua Bonbrink¹, Sebastian W. Bowers^{1,2}, David A. Evans¹, Chao-Chun Hung¹, Michael O'Neill¹, David Reiss¹, Kathryn Tarnopolska^{1,2}, Zachary Wu¹, Abhishek Baghel^{1,2}, Evin Arvaniti¹, Charles Beattie¹, Ottavia Bertolli¹, Alan Bridgland¹, Alessio Chiorboli¹, Miles Congreve¹, Alexander I. Cowen-Rivers¹, Andrew Cowie¹, Michael Figurnov¹, Fabian D. Fuchs¹, Harshad Guadagni¹, Rohan Jain¹, Yusuf A. Khan^{1,2}, Caroline M. K. Lee¹, Kuba Peltin¹, Anna Potapenko¹, Pascal Savay¹, Sukhdeep Singh¹, Adrian Stecula¹, Ashutk Thilakavandana¹, Catherine Tong¹, Sergei Yelensky¹, Ellen D. Zhang^{1,2}, Michal Zaidman¹, Augustin Eden¹, Victor Bapst^{1,2}, Pauliusius Kubilius^{1,2}, Max Jaderberg^{1,2}, Demis Hassabis^{1,2,3} & John M. Jumper^{1,2}

The introduction of AlphaFold 2¹ has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein modelling and design^{2–5}. 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

Article

Accurate structure prediction of biomolecular interactions with AlphaFold 3

<https://doi.org/10.1038/s41586-024-03847-4>

Received: 18 December 2022

Accepted: 28 April 2024

Published online: 8 May 2024

Open access

 Check for updates

Josh Abramson^{1,2}, John Jumper^{1,2}, Richard Evans^{1,2}, Tim Green^{1,2}, Alexander Pritzel^{1,2}, Amy Williams^{1,2}, Andrew J. Ballard¹, Joshua Bernstein¹, David A. Evans¹, Chao-Chun Huang¹, Michael O'Neil¹, Tommie Harrison¹, Zachary Wu¹, Abella Serrano-Pedraza¹, Emma Anyanwu¹, David Bertolli¹, Alan Bridgland¹, Alessio Chignone¹, Adam Evans¹, Andrew Goss¹, Michael Figurnov¹, Fabian F. Hoyer¹, Rohan Joshi¹, Yusuf A. Khan^{1,2}, Caroline M. H. Lee¹, Ali H. Li¹, Pascal Savary¹, Sukhdeep Singh¹, Adrian Stecula¹, Catherine Tong¹, Sergei Valasek¹, Ellen D. Zhang¹, Martin Zidek¹, Victor Zepher¹, Paulius Velička^{1,2}, Max Jaderberg^{1,2}, & John M. Jumper^{1,2}

The introduction of AlphaFold 2¹ has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein design and drug discovery^{2–5}. 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!

Data augmentation

Article

Accurate structure prediction of biomolecular interactions with AlphaFold3

<https://doi.org/10.1038/s41586-024-03847-4>

Received: 18 December 2022

Accepted: 28 April 2024

Published online: 8 May 2024

Open access

 Check for updates

Josh Abramson^{1,2}, Jonas Alnæs^{1,2}, Richard Evans^{1,2}, Tim Green^{1,2}, Alexander Pritzel^{1,2}, Amy Williams^{1,2}, Andrew J. Ballard¹, Joshua Bonhoeffer¹, David A. Evans¹, Chao-Chun Huang¹, Michael O'Byrne¹, John J. H. Taylor¹, Zachary Wu¹, Abhishek Bera^{1,3}, David Bertolli¹, Alan Bridgland¹, Alessio Chignone^{1,4}, Owen Elser¹, Andrew Goss¹, Michael Figurnov¹, Fabian F. Friedrich¹, Robert Gao¹, Yousaf A. Khan^{1,5}, Caroline M. R. Lee¹, Jakub Marczak¹, Pascal Savy¹, Sukhdeep Singh¹, Adrian Stecula¹, Catherine Tong¹, Sergei Valasek¹, Ellen D. Zhang^{1,6}, Martin Zidek¹, Victor Zepher^{1,6}, Paulius Velička^{1,6}, Max Jaderberg^{1,6}, & John M. Jumper^{1,2}

The prediction of AlphaFold 2 has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein design and drug discovery¹. 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

Article

<https://doi.org/10.1008/978989-034-03487-9>

Received: 19 December 2023

Accepted: 29 April 2014

© 2004 Blackwell Publishing Ltd *Journal of Internal Medicine* 255: 103–110

© 2004 Blackwell Publishing Ltd

 [cc BY-SA](https://creativecommons.org/licenses/by-sa/4.0/)

 Check for updates

Equivalents

Introduction of AlphaFold 2¹ has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein modelling and design²⁻⁴. 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 Wang¹ Ahmed A. Elhag^{1,2} Navdeep Jaitly¹ Joshua M. Susskind¹ Miguel Ángel Bautista¹

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-generation, and enable it to scale up to

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 utterly infeasible. In recent years, machine learning methods

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

Data augmentation

Article

Accurate structure prediction of biomolecular interactions with AlphaFold 3

<https://doi.org/10.1038/s41586-024-0387-9>

Received: 18 December 2023

Accepted: 28 April 2024

Published online: 8 May 2024

Open access

Check for updates

Josh Abramson^{1,2}, Alexander Pritikin^{1,2}, Andrew Evans^{1,2}, Tim Ovchinnikov^{1,2}, David A. Evans^{1,2}, Chao-Chun Huang^{1,2}, Joshua Bernick^{1,2}, Michael J. Bryant^{1,2}, Tatyana Potapova^{1,2}, Zachary Wu^{1,2}, Abhishek Jaiswal^{1,2}, Eric Aker^{1,2}, David Bertolli^{1,2}, Alex Bridgland^{1,2}, Alessio Caporaso^{1,2}, Adam Evans^{1,2}, Andrew Cowie^{1,2}, Michael Figurnov^{1,2}, Miles Huggins^{1,2}, Patrick Jain^{1,2}, Yousang Kim^{1,2}, Caroline M. Lee^{1,2}, John Jumper^{1,2}, Pascal Savary^{1,2}, Subhojit Singh^{1,2}, Adrian Stecula^{1,2}, Catherine Yang^{1,2}, Sergei Yelensky^{1,2}, Ellen D. Zhang^{1,2}, Martin Zidek^{1,2}, Victor Zepeda^{1,2}, Pauliusius Zukas^{1,2}, Max Zuckerman^{1,2}, & John M. Jumper^{1,2}

Structure prediction of AlphaFold 2 has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein design and drug discovery¹. 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 substantial 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
aditiks@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

Yiyang Wang¹, Ahmed A. Elhag^{1,2}, Navdeep Jaitly¹, Joshua M. Susskind¹, Miguel Ángel Bautista¹

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 prediction and enable us to scale our model

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 (Aschard & 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. Langer¹, Sergey N. Poddanyakov¹, and Michele Corietti¹

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.corietti@epfl.ch

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

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 with AlphaFold3

<https://doi.org/10.1038/s41586-024-03877-w>

Received: 18 December 2022

Accepted: 28 April 2024

Published online: 8 May 2024

Open access

Check for updates

No Equivariance!

Josh Abramson^{1,2}, Alexander Pritzel^{1,2}, Andrew J. Ballard¹, Joshua Bernstein¹, Michael Bryden¹, Eric C. Chen¹, David A. Evans², Chao-Chun Hung¹, Tamas Korbai¹, Zachary Wu¹, Abhishek S. Jaiswal¹, David Berntz¹, Alex Bridgland¹, Alessio Caporaso¹, Adam Evans¹, Andrew Gower¹, Michael Hargrave¹, James H. Hill¹, Yousang H. Kim¹, Caroline M. Lee¹, Pascal Savary¹, Subhojit Singh¹, Adrian Stecula¹, Catherine Tong¹, Sergei Yalovnev¹, Ellen D. Zhang¹, Justin Zlot¹, Victor Bapst¹, Paulius Radek¹, Max Jaderberg^{1,2}, & John M. Jumper^{1,2}

Structure prediction of AlphaFold 2 has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein design and drug discovery. 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 3 model demonstrates substantial 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. Krishnamoorti
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

Yiyang Wang¹, Ahmed A. Elhag^{1,2}, Navdeep Jaitly¹, Joshua M. Susskind¹, Miguel Ángel Bautista¹

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 prediction and enable us to predict conformers that

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 (Ashford & Gomez-Bombarelli, 2022). This complicates brute force and exhaustive approaches, making them virtually unfeasible for even moderately small molecules.

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

Probing the effects of broken symmetries in machine learning

Marcel F. Langer¹, Sergey N. Poudyakov¹, & Michele Corietti¹

Laboratory of Computational Science and Modeling and National Centre for Computational Design and Discovery of Novel Materials MARVEL, Institute of Materials, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

* Author to whom any correspondence should be addressed.

E-mail: michele.corietti@epfl.ch

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

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,^{1,2,3,4} Victor Garcia Satorras,^{1,4,5} Chin-Wei Huang,¹ Daniel Zügner,¹ Marco Federici,^{1,6} Cecilia Clementi,^{3,4} Frank Noé,¹ Robert Pinsler,⁸ and Rianne van den Berg¹

¹Work done during an internship at Microsoft Research (Amsterdam).

²University of Copenhagen, Department of Computer Science, Universitetsparken 1, Copenhagen, 2100, Denmark.

³AI4Science, Microsoft Research, Evert van de Beekstraat 354, Amsterdam, 1118 CZ, The Netherlands.

⁴AI4Science, Microsoft Research, Karl-Liebknecht-Strasse 32, Berlin, 10178, Germany.

⁵University of Amsterdam, Informatics Institute, Science Park 904, Amsterdam, 1098 XH, The Netherlands.

⁶Freie Universität Berlin, Department of Physics, Arnimallee 12, Berlin, 14195, Germany.

⁷AI4Science, Microsoft Research, 21 Station Road, Cambridge, CB1 3FB, United Kingdom.

⁸Equal contribution.

* E-mail: ma@cs.ku.dk; victorgarcia@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 prediction of biomolecular interactions with AlphaFold3

<https://doi.org/10.1038/s41586-024-03877-w>

Received: 18 December 2023

Accepted: 28 April 2024

Published online: 8 May 2024

Open access

Check for updates

Josh Abramson^{1,2}, Alexander Pritzel¹, Andrew J. Ballard¹, Joshua Bernstein¹, Michael Bryden¹, Eric C. Chen¹, David A. Evans¹, Chao-Chun Hung¹, Tamas Kuzsok¹, Zachary Wu¹, Alexei Arvanitidis¹, David Berntz¹, Alex Bridgland¹, Alessio Caporaso¹, Adam Evans¹, Andrew Ganev¹, Michael Figurelli¹, James G. Hahn¹, Yousuik A. Kim¹, Caroline M. Lee¹, Adam Papp¹, Pascal Parv¹, Subhojit Sanyal¹, Adrian Stecula¹, Catherine Tong¹, Sergei Yalovoy¹, Ellen D. Zhang¹, Martin Zidek¹, Victor Zepher¹, Paulius Zukas¹, Max Jaderberg^{1,2}, & John M. Jumper^{1,2}

Structure prediction of AlphaFold 2 has spurred a revolution in modelling the structure of proteins and their interactions, enabling a huge range of applications in protein design and drug discovery. 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 3 model demonstrates substantial accuracy

No Equivalence!

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. Krishnamurthy
UC Berkeley, LBNL
aditik@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

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

Marloes Arts,^{1,2,3} Victor Garcia Satorras,^{1,4,5} Chin-Wei Huang,¹ Daniel Zügner,¹ Marco Federici,^{1,6} Cecilia Clementi,^{1,6} Frank Noé,¹ Robert Pinsler,⁶ and Rianne van den Berg¹

¹Work done during an internship at Microsoft Research (Amsterdam).
²University of Copenhagen, Department of Computer Science, Universitetsparken 1, Copenhagen, 2100, Denmark.
³AI4Science, Microsoft Research, Evert van de Beekstraat 354, Amsterdam, 1118 CZ, The Netherlands.
⁴AI4Science, Microsoft Research, Karl-Liebknecht-Straße 32, Berlin, 10178, Germany.
⁵University of Amsterdam, Informatics Institute, Science Park 904, Amsterdam, 1098 XH, The Netherlands.
⁶Freie Universität Berlin, Department of Physics, Arnimallee 13, Berlin, 14195, Germany.
⁷AI4Science, Microsoft Research, 21 Station Road, Cambridge, CB1 3FB, United Kingdom.
⁸Equal contribution.

*E-mail: ma@ku.dk; victorga@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

Swallowing the Bitter Pill: Simplified Scalable Conformer Generation

Yiyang Wang¹, Ahmed A. Elhag^{1,2}, Navdeep Jaitly¹, Joshua M. Susskind¹, Miguel Ángel Bautista¹

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 prediction and enable the prediction of

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 (Aschard & Gomez-Bombarelli, 2022). This complicates brute force and exhaustive approaches, making them virtually unfeasible for even moderately small molecules. Scalability methods like OMIGA (Hawkins et al., 2010)

Probing the effects of broken symmetries in machine learning

Marcel F. Langer¹, Sergey N. Rudnitskiy¹, and Michele Ceriotti¹

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.cerotti@epfl.ch

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

DOES EQUIVARIANCE MATTER AT SCALE?

Johann Brechmer¹, Sönke Behrendts¹, Pin de Haan¹, Theo Cohen¹
Qualcomm AI Research
naill@johannbrechmer.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

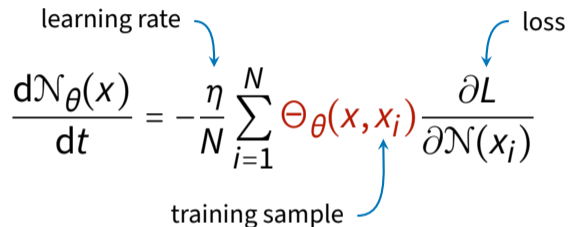
Training dynamics under continuous gradient descent:

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

learning rate

loss

training sample



Empirical NTK

Training dynamics under continuous gradient descent:

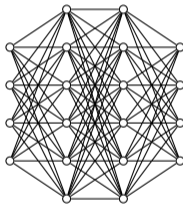
$$\frac{d\mathcal{N}_\theta(x)}{dt} = -\frac{\eta}{N} \sum_{i=1}^N \Theta_\theta(x, x_i) \frac{\partial L}{\partial \mathcal{N}(x_i)}$$

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

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

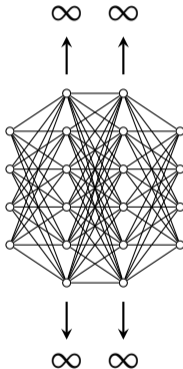
Infinite width limit

[Jacot et al. 2018]



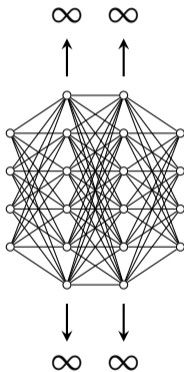
Infinite width limit

[Jacot et al. 2018]



Infinite width limit

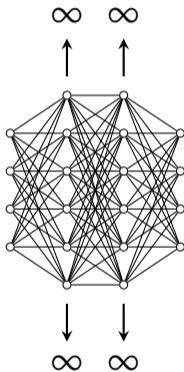
[Jacot et al. 2018]



👍 NTK becomes independent of initialization

Infinite width limit

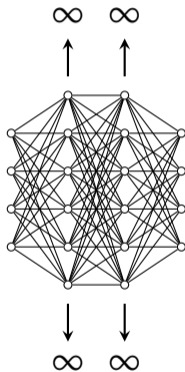
[Jacot et al. 2018]



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

Infinite width limit

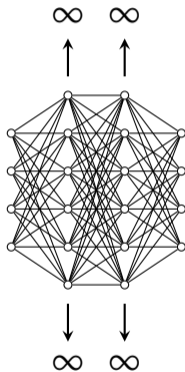
[Jacot et al. 2018]



- 👍 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]


① At infinite width, the mean prediction is given by

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

Mean prediction from NTK

[Jacot et al. 2018]

① At infinite width, the mean prediction is given by



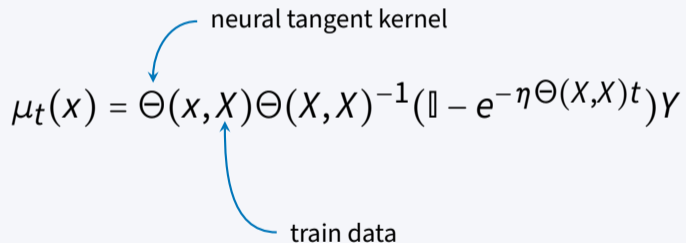
neural tangent kernel

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

Mean prediction from NTK

[Jacot et al. 2018]

① At infinite width, the mean prediction is given by



The diagram shows the equation $\mu_t(x) = \Theta(x, X) \Theta(X, X)^{-1} (\mathbb{I} - e^{-\eta \Theta(X, X) t}) \gamma$. A blue curved arrow points from the text "neural tangent kernel" to the $\Theta(x, X)$ term. Another blue curved arrow points from the text "train data" to the X in the $\Theta(X, X)$ term.

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

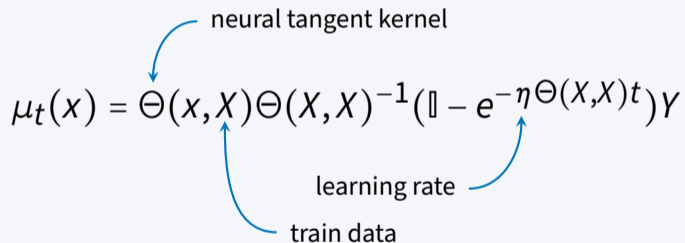
neural tangent kernel

train data

Mean prediction from NTK

[Jacot et al. 2018]

① At infinite width, the mean prediction is given by



The diagram shows the formula $\mu_t(x) = \Theta(x, X) \Theta(X, X)^{-1} (\mathbb{I} - e^{-\eta \Theta(X, X) t}) \gamma$ with three blue arrows pointing to specific parts: one from 'neural tangent kernel' to $\Theta(x, X)$, one from 'train data' to γ , and one from 'learning rate' to η .

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

neural tangent kernel

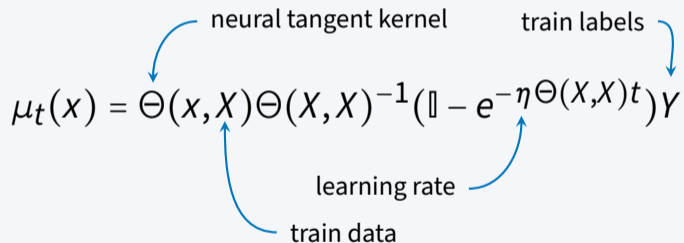
learning rate

train data

Mean prediction from NTK

[Jacot et al. 2018]

① At infinite width, the mean prediction is given by



The diagram shows the formula $\mu_t(x) = \Theta(x, X) \Theta(X, X)^{-1} (\mathbb{I} - e^{-\eta \Theta(X, X) t}) Y$ with four blue arrows pointing to its components: 'neural tangent kernel' points to $\Theta(x, X)$, 'train labels' points to Y , 'learning rate' points to η , and 'train data' points to X .

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

neural tangent kernel

train labels

learning rate

train data

Data augmentation

Data augmentation at infinite width

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

Data augmentation at infinite width

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

The diagram illustrates the components of the equation $\mu_t(x) = \Theta(x, X) \Theta(X, X)^{-1} (\mathbb{I} - e^{-\eta \Theta(X, X) t}) \gamma$. Blue arrows point from the text labels to the corresponding terms in the equation:

- augmented data** points to $\Theta(x, X)$.
- augmented data** points to $\Theta(X, X)$.
- augmented data** points to $\Theta(X, X)^{-1}$.
- augmented labels** points to γ .

Data augmentation at infinite width

group transformation

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

augmented data

augmented labels

The diagram illustrates the equation for data augmentation at infinite width. A blue arrow points from the text 'group transformation' to the term $\rho(g)$ in the expression $\rho(g)x$. Below the equation, the text 'augmented data' has three blue arrows pointing to the terms $\rho(g)x$, X , and $\Theta(X, X)^{-1}$. The text 'augmented labels' has two blue arrows pointing to the terms $\Theta(X, X)$ and Y .

Kernel transformation

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

$$\begin{aligned}\Theta(\rho(g)x, \rho(g)x') &= \rho_K(g) \Theta(x, x') \rho_K^\top(g), \\ K(\rho(g)x, \rho(g)x') &= \rho_K(g) K(x, x') \rho_K^\top(g),\end{aligned}$$

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

$$\begin{aligned}\Theta(\rho(g)x, \rho(g)x') &= \rho_K(g) \Theta(x, x') \rho_K^\top(g), \\ K(\rho(g)x, \rho(g)x') &= \rho_K(g) K(x, x') \rho_K^\top(g),\end{aligned}$$

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

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

Permutation shift

- On the training data, group transformations permute the samples

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

- 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

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

augmented data

augmented labels

The diagram illustrates the equation for data augmentation at infinite width. A blue arrow points from the text 'group transformation' to the term $\rho(g)$ in the expression $\rho(g)x$. Below the equation, the text 'augmented data' has four blue arrows pointing to the terms $\rho(g)x$, X , X , and X in the expression $\Theta(\rho(g)x, X) \Theta(X, X)^{-1}$. The text 'augmented labels' has two blue arrows pointing to the terms X and X in the expression $\Theta(X, X)^{-1}$. A final blue arrow points from 'augmented labels' to the Y term in the expression $(\mathbb{I} - e^{-\eta \Theta(X, X)t}) Y$.

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

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

augmented data

augmented labels

The diagram illustrates the equation for data augmentation at infinite width. A blue arrow points from the text 'group transformation' to the $\rho(g)$ term in the equation. Another blue arrow points from the text 'augmented data' to the x term. A third blue arrow points from the text 'augmented labels' to the Y term. Additionally, there are three blue arrows pointing from the 'augmented data' label to the $\Theta(x, X)$ and $\Theta(X, X)^{-1}$ terms, and two blue arrows pointing from the 'augmented labels' label to the $\Theta(X, X)^{-1}$ and $\Theta(X, X)^{-1}(\mathbb{I} - e^{-\eta\Theta(X, X)t})$ terms.

Data augmentation at infinite width

group transformation

augmented labels

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

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

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

Mean prediction

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

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

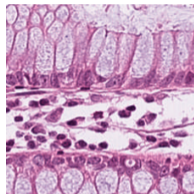
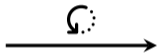
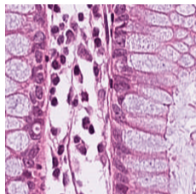
ⓘ At infinite width, the mean output at initialization is zero everywhere.

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

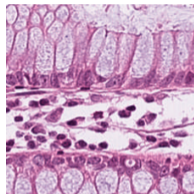
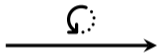
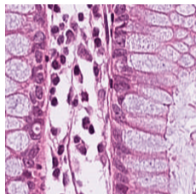
What Does An Augmented Ensemble Converge To?

Rotating images

Rotating images



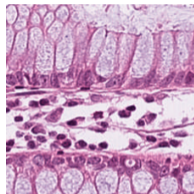
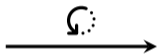
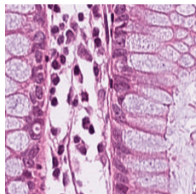
Rotating images



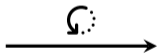
$f(x)$

$f : \text{pixels} \rightarrow \text{colors}$

Rotating images



$f(x)$
 $f : \text{pixels} \rightarrow \text{colors}$



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

Data augmentation and NTKs

Data augmentation and NTKs

Consider two ensembles:

trained without data augmentation

trained with data augmentation

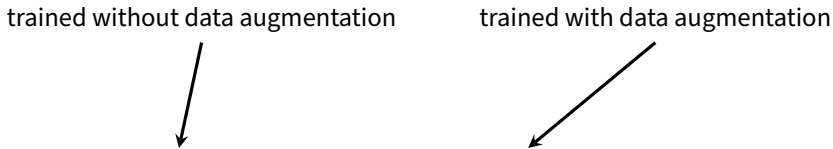
Data augmentation and NTKs

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

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

- ① 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

$$f'(y) = \int_X dx \, \kappa(x - y) f(x)$$

Group convolutions

[Cohen, Welling 2016]

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

- Ordinary convolutions

$$f'(y) = \int_X dx \, \kappa(x - y) f(x)$$

- Group convolutions

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

lifting

Group convolutions

[Cohen, Welling 2016]

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

- Ordinary convolutions

$$f'(y) = \int_X dx \, \kappa(x - y) f(x)$$

- Group convolutions

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

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

Group convolutions

[Cohen, Welling 2016]

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

- Ordinary convolutions

$$f'(y) = \int_X dx \, \kappa(x - y) f(x)$$

- Group convolutions

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

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

$$f' = \frac{1}{\text{vol}(G)} \int_G dg \, f(g) \quad \text{group pooling}$$

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

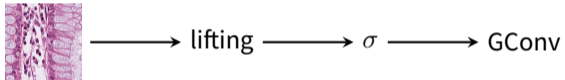
Stack GConv-layers to obtain an invariant network



→ lifting → σ

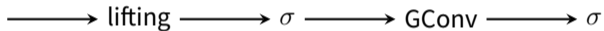
GCNNs

Stack GConv-layers to obtain an invariant network



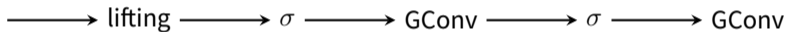
GCNNs

Stack GConv-layers to obtain an invariant network



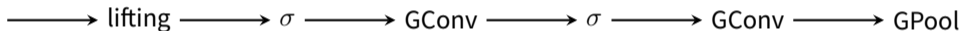
GCNNs

Stack GConv-layers to obtain an invariant network



GCNNs

Stack GConv-layers to obtain an invariant network



NTKs for GCNNs

For GCNN-layers, define the NNGP and NTK via

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

NTKs for GCNNs

For GCNN-layers, define the NNGP and NTK via

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

NTKs for GCNNs

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

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

$$K_{g,g'}^{(\ell+1)}(f,f') = \frac{1}{|S_K|} \int_{S_K} dh K_{gh,g'h}^{(\ell)}(f,f')$$

NTKs for GCNNs

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

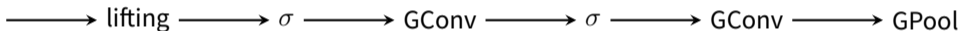
The layer-recursion for a GCNN-layer is given by

$$K_{g,g'}^{(\ell+1)}(f,f') = \frac{1}{|S_K|} \int_{S_K} dh \, K_{gh,g'h}^{(\ell)}(f,f')$$

$$\Theta_{g,g'}^{(\ell+1)}(f,f') = K_{g,g'}^{(\ell+1)}(f,f') + \frac{1}{|S_K|} \int_{S_K} dh \, \Theta_{gh,g'h}^{(\ell)}(f,f')$$

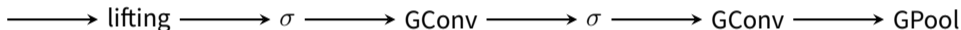
GCNNs

Stack GConv-layers to obtain an invariant network



NTKs for GCNNs

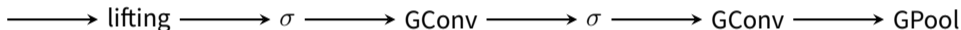
Stack GConv-layers to obtain an invariant network



Compute NTK with layer-wise recursion

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

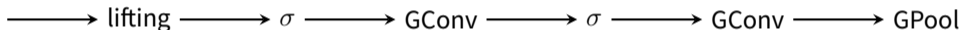


Compute NTK with layer-wise recursion

0

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

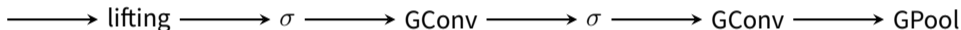


Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

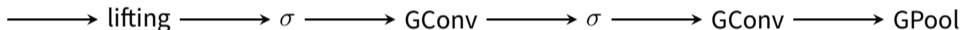


Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

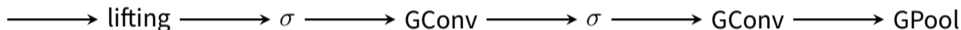


Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

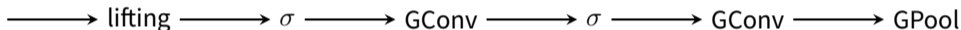


Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network

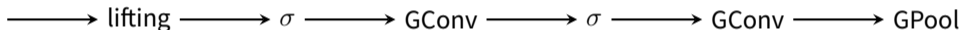


Compute NTK with layer-wise recursion

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

NTKs for GCNNs

Stack GConv-layers to obtain an invariant network



Compute NTK with layer-wise recursion

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

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



NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN



NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



→ FC → σ → FC → σ → FC

A GCNN



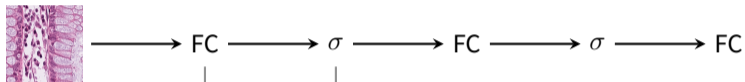
→ lifting



NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



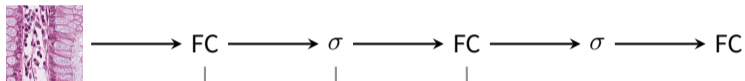
A GCNN



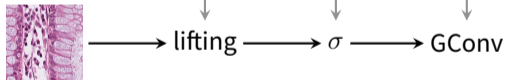
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN



NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



→ FC → σ → FC → σ → FC

A GCNN



→ lifting → σ → GConv → σ



NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



→ FC → σ → FC → σ → FC

A GCNN



→ lifting → σ → GConv → σ → GConv



NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



→ FC → σ → FC → σ → FC

A GCNN



→ lifting → σ → GConv → σ → GConv → GPool



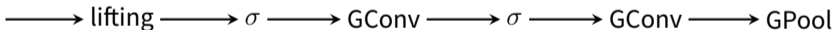
NTKs of MLPs and GCNNs

- Consider two neural networks

An MLP



A GCNN



- Then


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

Data augmentation of MLPs

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

Data augmentation of MLPs

before: non-aug


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

Data augmentation of MLPs

before: non-aug

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

before: aug

Data augmentation of MLPs

before: non-aug

before: aug

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

⇒ training the MLP on
G-augmented data

Data augmentation of MLPs

before: non-aug

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

before: aug

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

Data augmentation of MLPs

before: non-aug

before: aug

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



training the MLP on
G-augmented data

=

training the GCNN on
unaugmented data



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

Data augmentation of CNNs

Data augmentation of CNNs

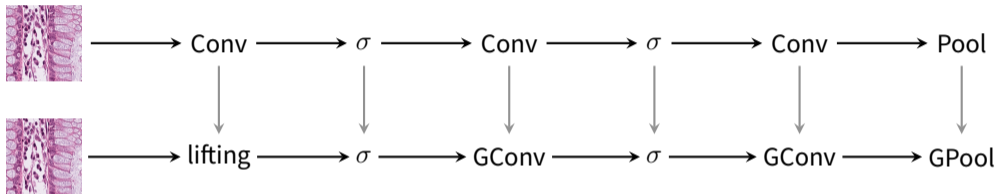
- Consider a CNN



→ Conv → σ → Conv → σ → Conv → Pool

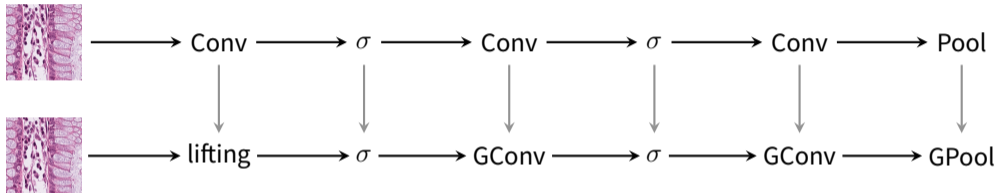
Data augmentation of CNNs

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



Data augmentation of CNNs

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

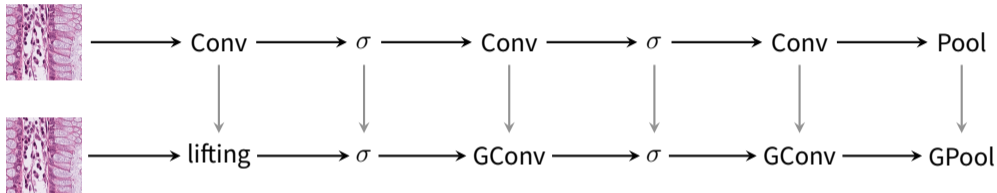


- Then

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

Data augmentation of CNNs

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



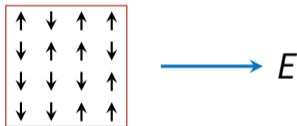
- Then

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

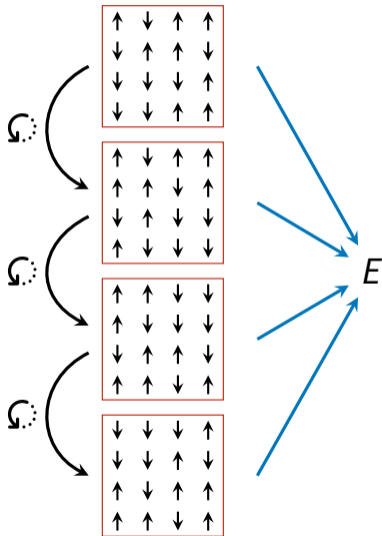
⇒ By training the CNN on rotated images, one obtains a roto-translation invariant GCNN

Experiments

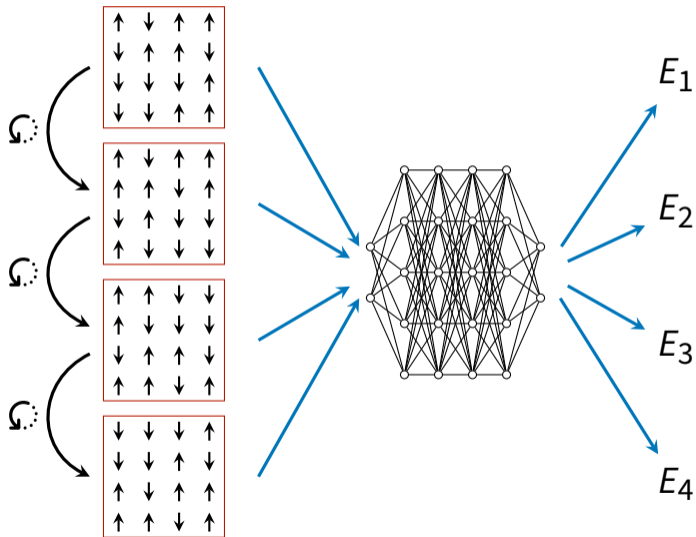
Ising model



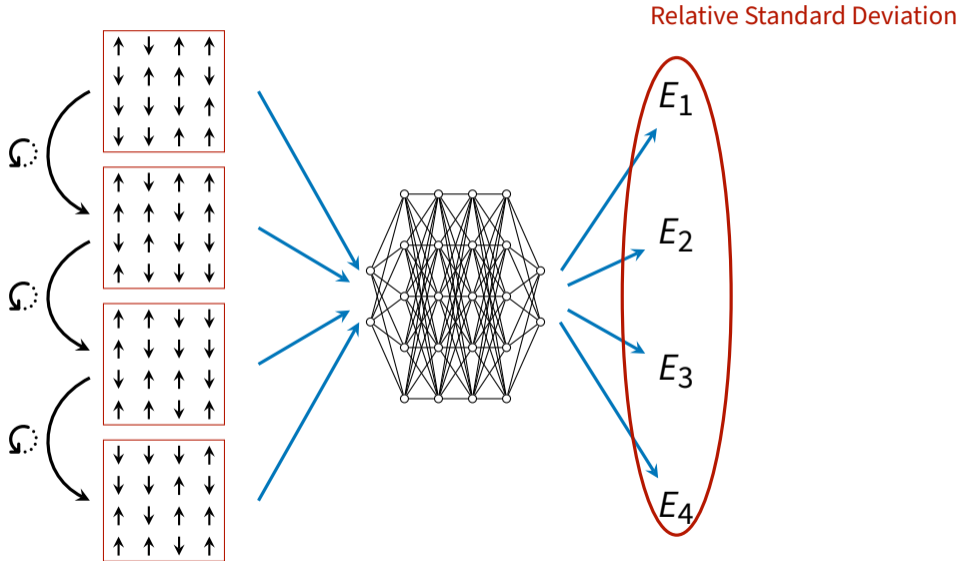
Ising model

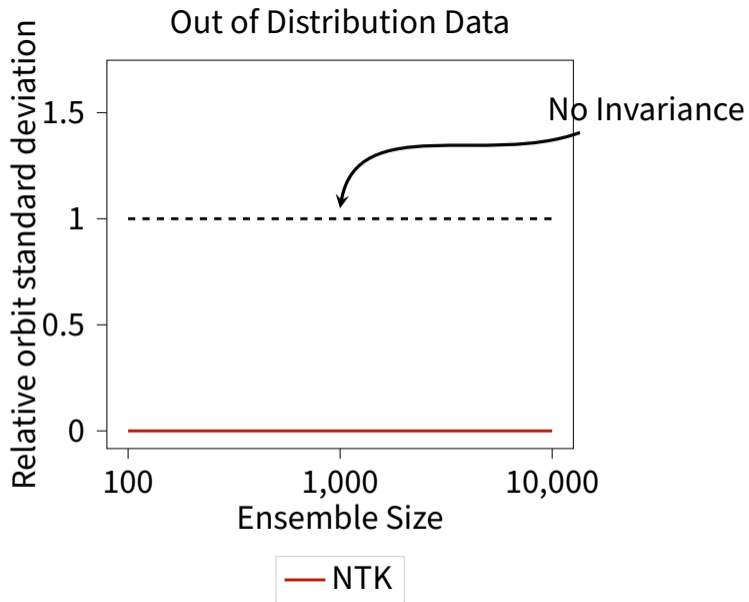


Ising model

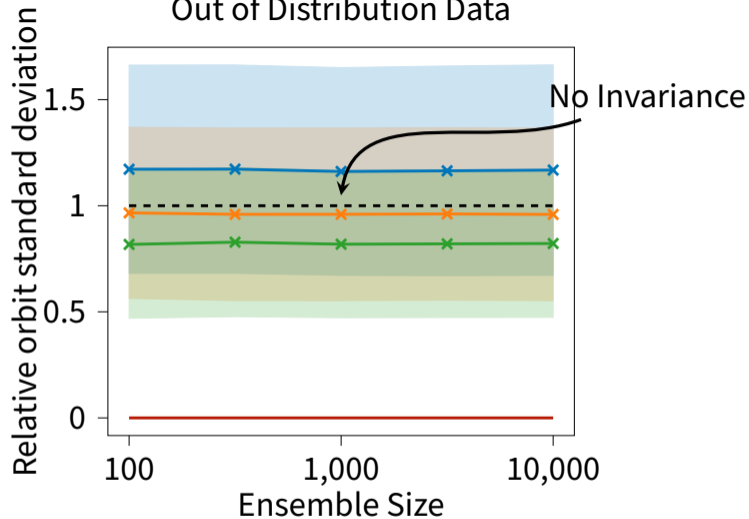


Ising model



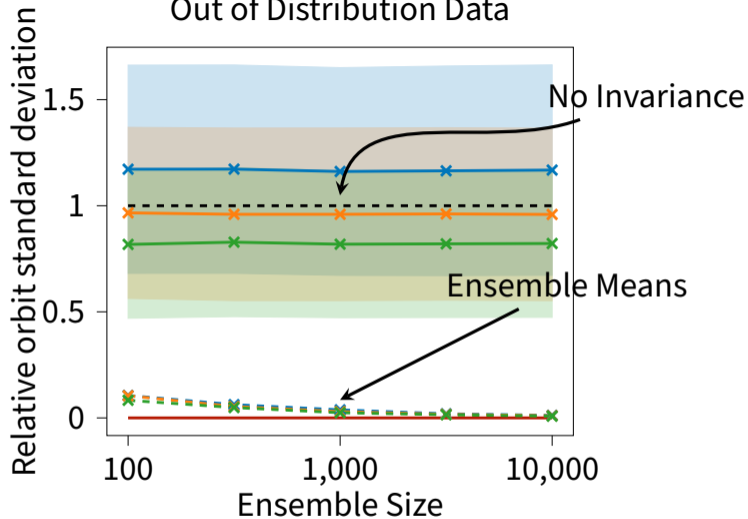


Out of Distribution Data

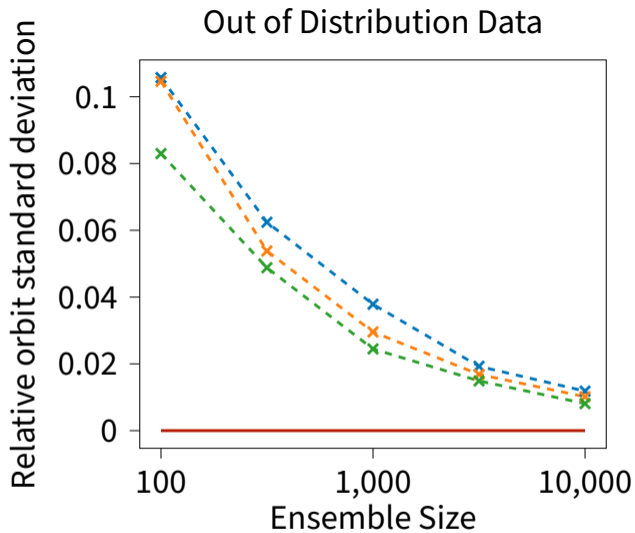


— NTK × Width 512 × Width 1024 × Width 2048

Out of Distribution Data



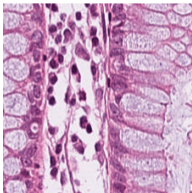
— NTK × Width 512 × Width 1024 × Width 2048



— NTK -x- Width 512 -x- Width 1024 -x- Width 2048

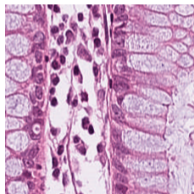
Histological slices

[Kather et al. 2018]



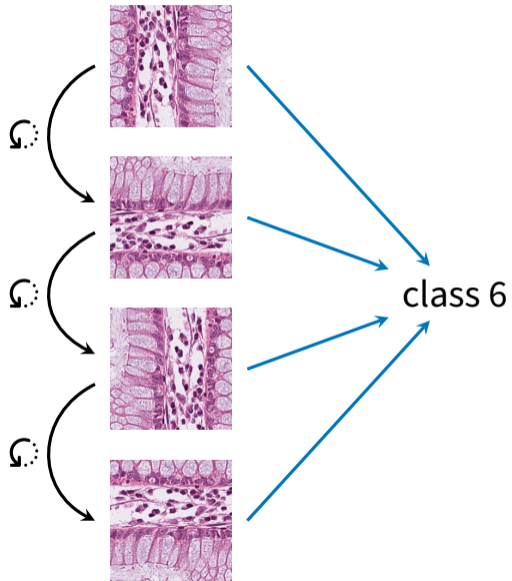
Histological slices

[Kather et al. 2018]

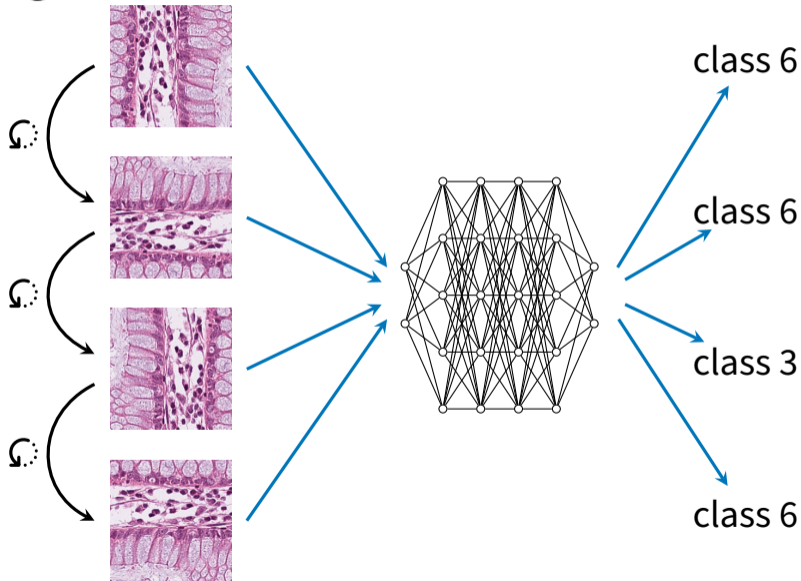


→ class 6

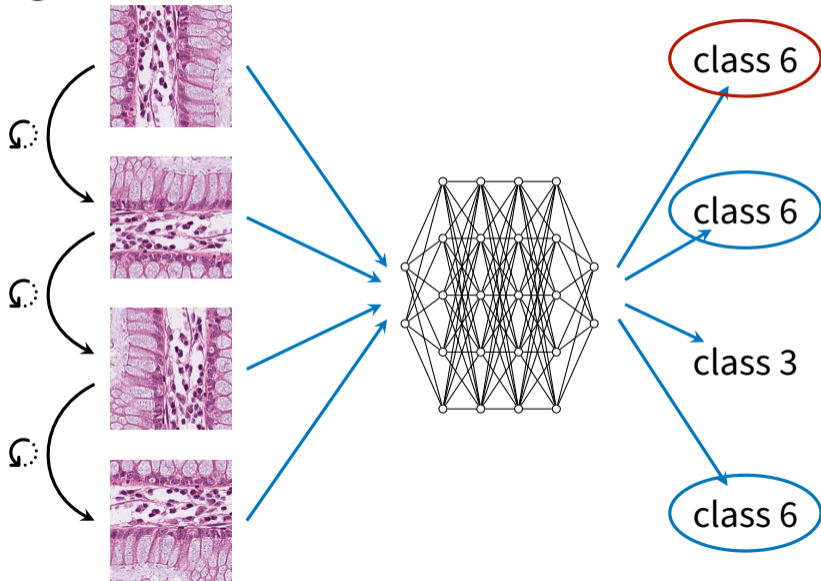
Histological slices



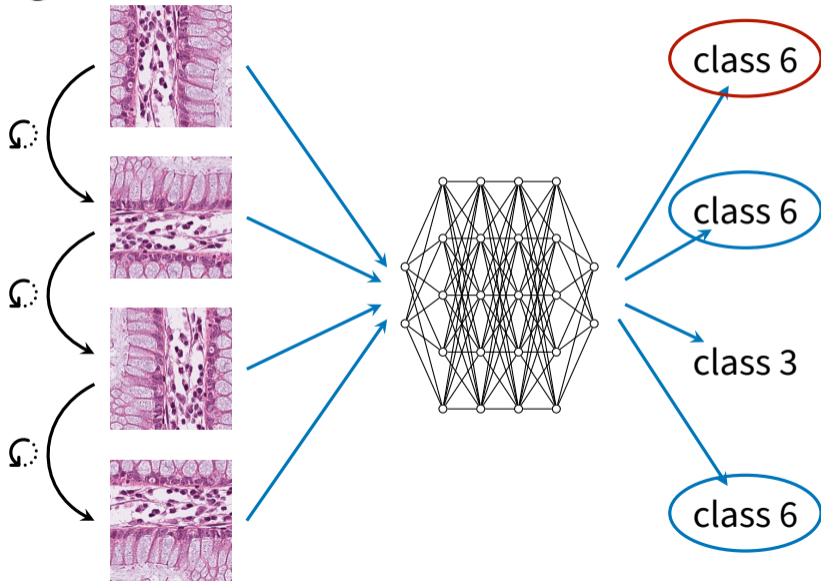
Histological slices



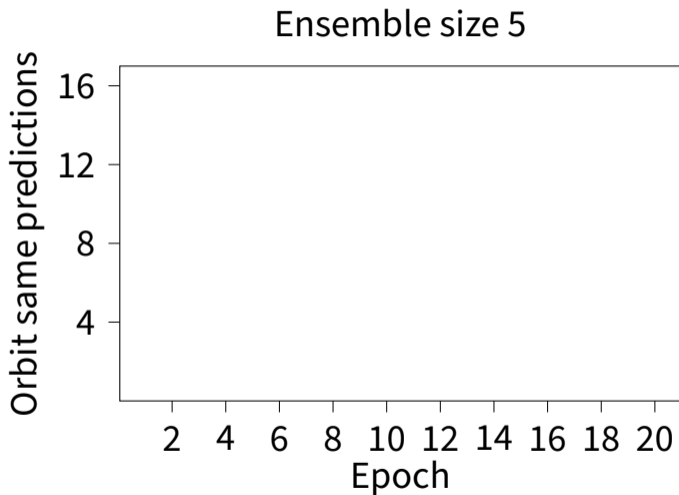
Histological slices



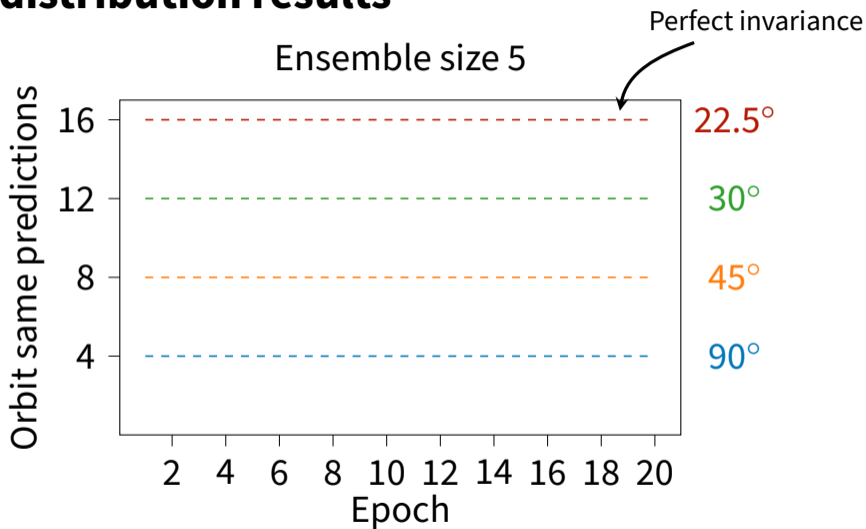
Histological slices



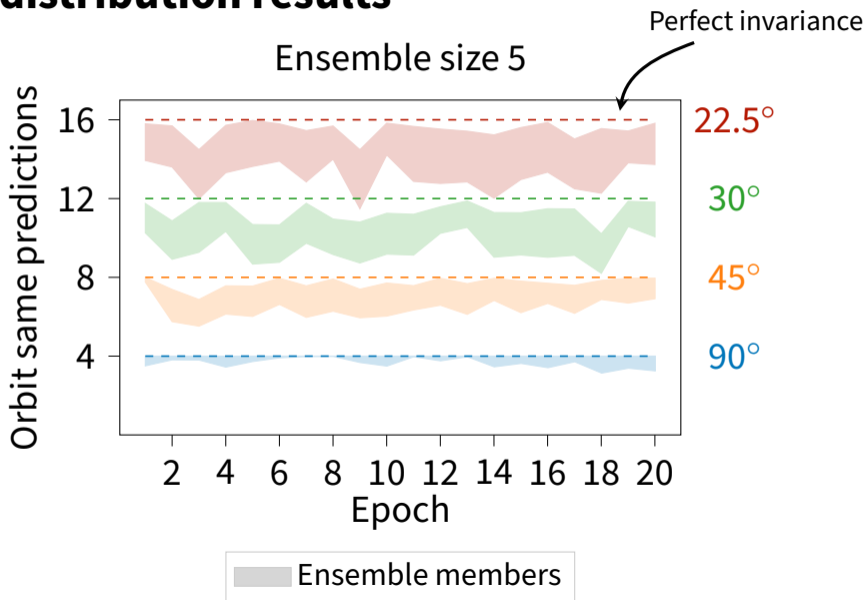
Out of distribution results



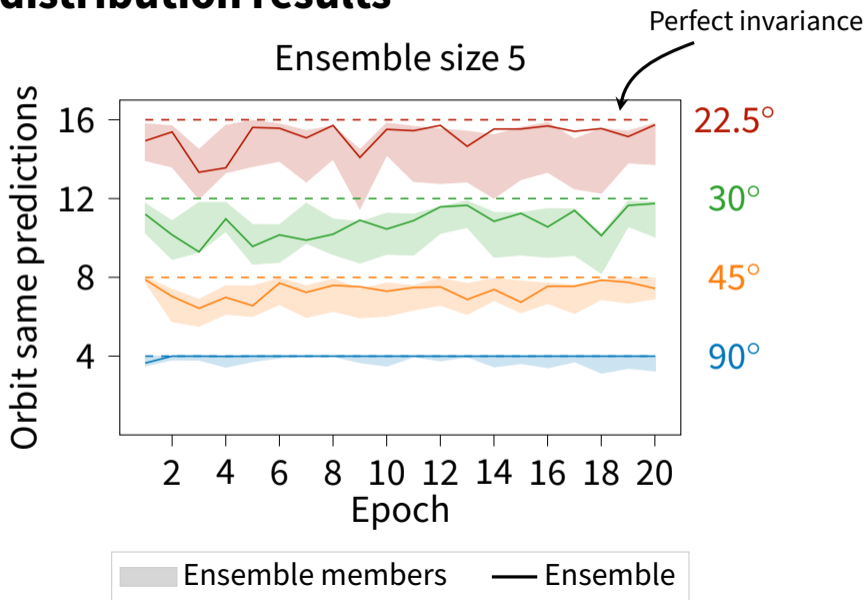
Out of distribution results



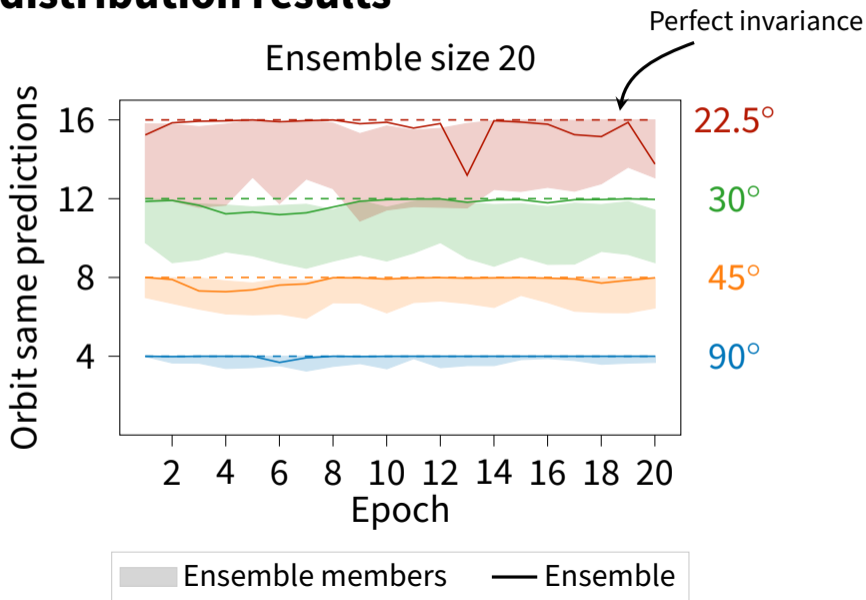
Out of distribution results



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

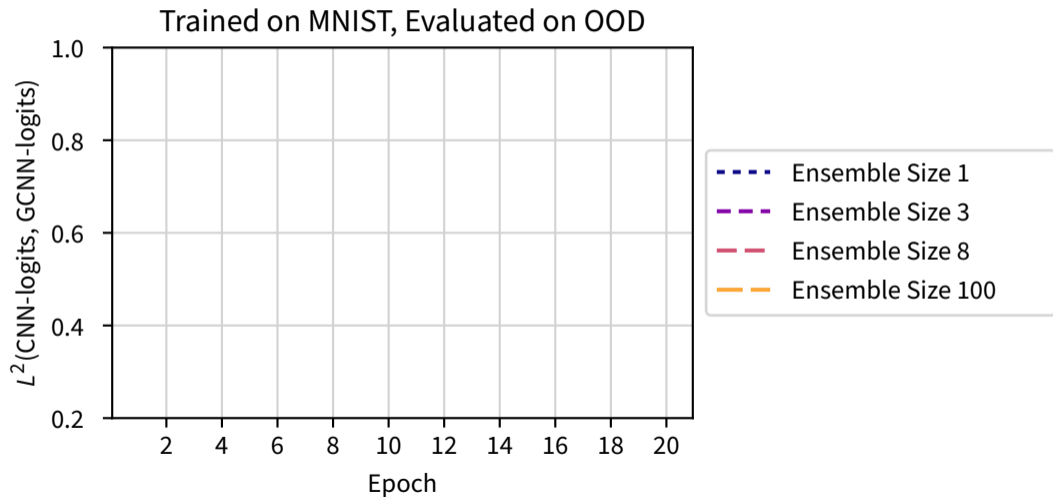
Orbit same predictions out of distribution:

	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

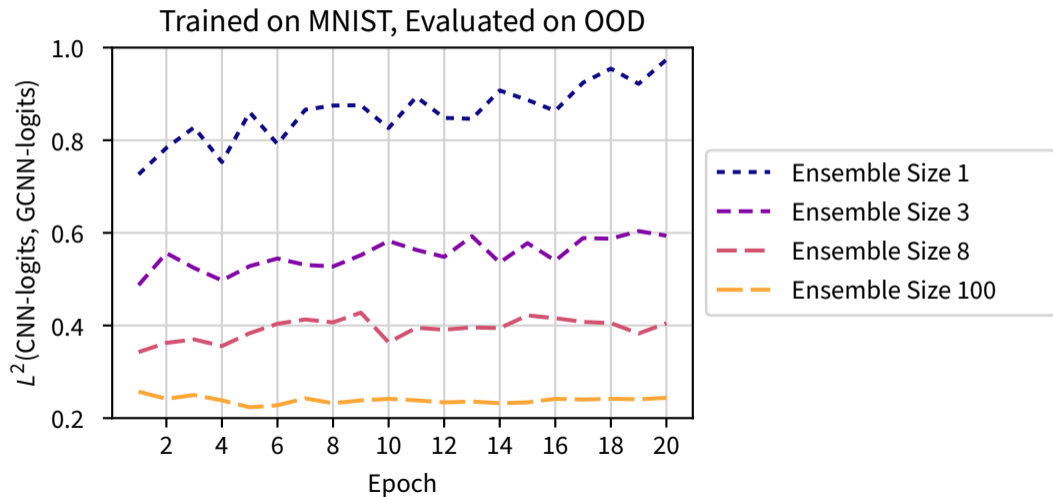
¹[Weiler et al. 2019], ²[Kaba et al. 2022]

Convergence of augmented CNNs to GCNNs

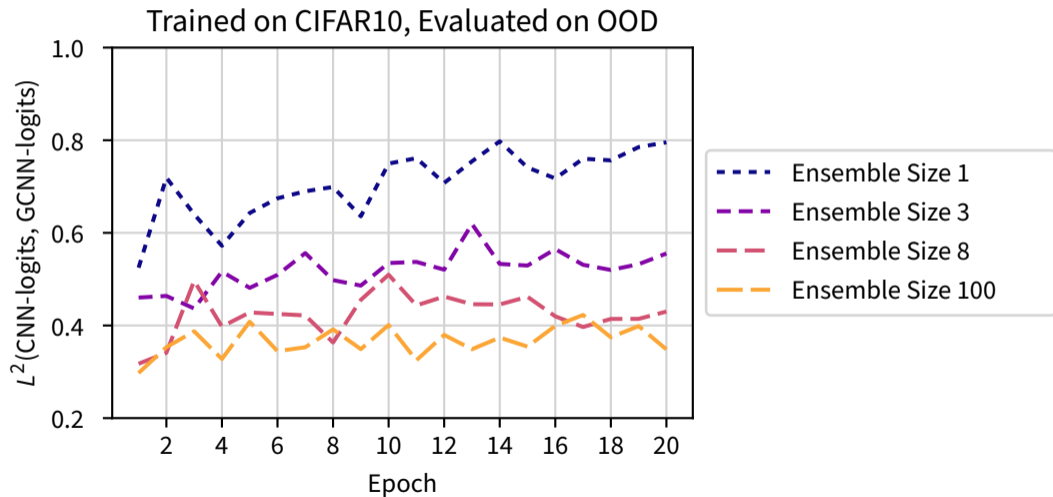
Convergence of augmented CNNs to GCNNs



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

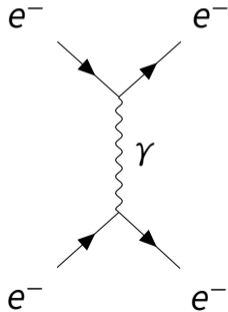
Feynman diagrams

Feynman diagrams are used to compute statistics

Feynman diagrams

Feynman diagrams are used to compute statistics

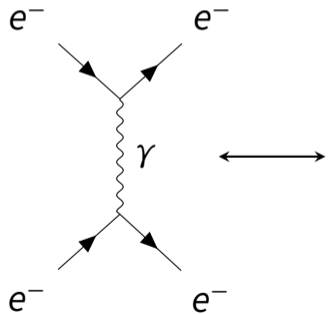
E.g. Electron-electron scattering



Feynman diagrams

Feynman diagrams are used to compute statistics

E.g. Electron-electron scattering



The diagram shows four external electron lines (solid lines with arrows pointing towards the vertices) and a vertical wavy line representing a photon exchange between two vertices. The top vertex has two incoming lines from the left and two outgoing lines to the right. The bottom vertex has two incoming lines from the left and two outgoing lines to the right. The wavy line connects the two vertices and is labeled with the Greek letter gamma (γ). A double-headed arrow points from the diagram to the corresponding mathematical expression.

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

Non-Gaussian Corrections from Physics

- Taylor-expand network statistics in $1/\text{width}$
- Use Feynman diagrams to compute non-Gaussian corrections

Non-Gaussian Corrections from Physics

- Taylor-expand network statistics in $1/\text{width}$
- 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.

The diagram illustrates the decomposition of a tensor into three components: cumulant, preactivation, and NTK fluctuation. The tensor is represented as $\mathbb{E}_{\theta}^c[z_{i_1}^{(\ell)}(x_1), z_{i_2}^{(\ell)}(x_2), \widehat{\Delta\Theta}_{i_3 i_4}^{(\ell)}(x_3, x_4)]$. Blue arrows point from the labels to the corresponding parts of the expression: 'cumulant' points to the expectation operator \mathbb{E}_{θ}^c , 'preactivation' points to the preactivation functions $z_{i_1}^{(\ell)}(x_1)$ and $z_{i_2}^{(\ell)}(x_2)$, and 'NTK fluctuation' points to the NTK fluctuation term $\widehat{\Delta\Theta}_{i_3 i_4}^{(\ell)}(x_3, x_4)$. A definition for the NTK fluctuation is provided: $\widehat{\Delta\Theta}^{(\ell)} = \widehat{\Theta}^{(\ell)} - \mathbb{E}_{\theta}[\widehat{\Theta}^{(\ell)}]$.

cumulant \mathbb{E}_{θ}^c preactivation $z_{i_1}^{(\ell)}(x_1), z_{i_2}^{(\ell)}(x_2)$ NTK fluctuation, $\widehat{\Delta\Theta}^{(\ell)} = \widehat{\Theta}^{(\ell)} - \mathbb{E}_{\theta}[\widehat{\Theta}^{(\ell)}]$

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

Tensors for neural network statistics

[Roberts, Yaida 2022]

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

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

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

Decompose these into tensors

$$= \frac{1}{n} \left(D_{12\mathbf{34}}^{(\ell)} \delta_{i_1 i_2} \delta_{i_3 i_4} + F_{\mathbf{1324}}^{(\ell)} \delta_{i_1 i_3} \delta_{i_2 i_4} + F_{1\mathbf{423}}^{(\ell)} \delta_{i_1 i_4} \delta_{i_2 i_3} \right)$$

Tensors for neural network statistics

[Roberts, Yaida 2022]

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

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

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

Decompose these into tensors

$$= \frac{1}{n} \left(D_{12\mathbf{3}4}^{(\ell)} \delta_{i_1 i_2} \delta_{i_3 i_4} + F_{\mathbf{1}324}^{(\ell)} \delta_{i_1 i_3} \delta_{i_2 i_4} + F_{1\mathbf{4}23}^{(\ell)} \delta_{i_1 i_4} \delta_{i_2 i_3} \right)$$

Gram tensor, $F_{\mathbf{1}324}^{(\ell)} = F^{(\ell)}(x_1, \mathbf{x}_3, x_2, x_4)$

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.

$$F_{1\mathbf{3}2\mathbf{4}}^{(\ell+1)} = \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_{\mathbf{3}}'^{(\ell)} \sigma_{\mathbf{4}}'^{(\ell)} \rangle_{K^{(\ell)}} \Theta_{\mathbf{34}}^{(\ell)} \\ + \sum_{a,\beta,\gamma,\delta=1}^4 \langle \sigma_1^{(\ell)} \sigma_{\mathbf{3}}'^{(\ell)} z_a^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_{\mathbf{4}}'^{(\ell)} z_{\beta}^{(\ell)} \rangle_{K^{(\ell)}} K_{(\ell)}^{a\gamma} K_{(\ell)}^{\beta\delta} F_{\gamma\mathbf{3}\delta\mathbf{4}}^{(\ell)}$$

Tensor recursions

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

$$\begin{aligned}
 F_{1324}^{(\ell+1)} &= \langle \overset{\sigma(z^{(\ell)}(x_1))}{\sigma_1^{(\ell)}} \sigma_2^{(\ell)} \overset{\text{Gaussian expectation with cov. } K^{(\ell)}}{\sigma_3'^{(\ell)}} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}} \Theta_{34}^{(\ell)} \\
 &+ \sum_{a,\beta,\gamma,\delta=1}^4 \langle \sigma_1^{(\ell)} \sigma_3'^{(\ell)} z_a^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_4'^{(\ell)} z_\beta^{(\ell)} \rangle_{K^{(\ell)}} \overset{\text{Components of } (K^{(\ell)})^{-1}}{K_{(\ell)}^{\alpha\gamma} K_{(\ell)}^{\beta\delta}} F_{\gamma 3 \delta 4}^{(\ell)}
 \end{aligned}$$

Tensor recursions

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

$$\begin{aligned}
 F_{1324}^{(\ell+1)} &= \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3^{(\ell)} \sigma_4^{(\ell)} \rangle_{K^{(\ell)}} \Theta_{34}^{(\ell)} \\
 &+ \sum_{a,\beta,\gamma,\delta=1}^4 \langle \sigma_1^{(\ell)} \sigma_3^{(\ell)} z_a^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_4^{(\ell)} z_\beta^{(\ell)} \rangle_{K^{(\ell)}} K_{(\ell)}^{\alpha\gamma} K_{(\ell)}^{\beta\delta} F_{\gamma 3 \delta 4}^{(\ell)}
 \end{aligned}$$

Annotations:

- $\sigma(z^{(\ell)}(x_1))$ points to $\sigma_1^{(\ell)}$
- Gaussian expectation with cov. $K^{(\ell)}$ points to $\langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3^{(\ell)} \sigma_4^{(\ell)} \rangle_{K^{(\ell)}}$
- Components of $(K^{(\ell)})^{-1}$ points to $K_{(\ell)}^{\alpha\gamma} K_{(\ell)}^{\beta\delta}$

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.

$$F_{1324}^{(\ell+1)} = \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3'^{(\ell)} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}} \Theta_{34}^{(\ell)} + \sum_{a,\beta,\gamma,\delta=1}^4 \langle \sigma_1^{(\ell)} \sigma_3'^{(\ell)} z_a^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_4'^{(\ell)} z_\beta^{(\ell)} \rangle_{K^{(\ell)}} K_{(\ell)}^{\alpha\gamma} K_{(\ell)}^{\beta\delta} F_{\gamma 3 \delta 4}^{(\ell)}$$

Annotations in the diagram:

- Blue arrow from $\sigma(z^{(\ell)}(x_1))$ to $\sigma_1^{(\ell)}$
- Blue arrow from "Gaussian expectation with cov. $K^{(\ell)}$ " to $\langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3'^{(\ell)} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}}$
- Blue arrow from "Components of $(K^{(\ell)})^{-1}$ " to $K_{(\ell)}^{\alpha\gamma} K_{(\ell)}^{\beta\delta}$

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]

$$Z_a \equiv a \bullet \text{---} \quad \langle \rangle_{K^{(\ell)}} \equiv \bigcirc \quad \begin{array}{c} \beta \bullet \\ \diagup \\ \text{---} \widehat{\Delta G}_{i,\alpha\beta}^{(\ell)} \text{---} \\ \diagdown \\ a \bullet \end{array} \sim \frac{1}{n}$$

Feynman diagrams

Use Feynman diagrams to compute these recursions.

- For preactivations, can read off Feynman rules from NN probability distribution, e.g.

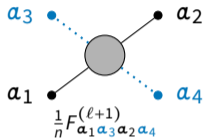
[Banta et al. 2024]

$$Z_a \equiv a \bullet \text{---} \quad \langle \rangle_{K^{(\ell)}} \equiv \bigcirc \quad \begin{array}{c} \beta \bullet \\ \diagup \\ \text{---} \widehat{\Delta G}_{i,\alpha\beta}^{(\ell)} \text{---} \\ \diagdown \\ a \bullet \end{array} \sim \frac{1}{n}$$

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

Feynman rules relevant for the F-tensor

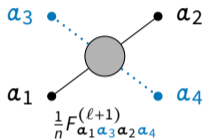
$$\widehat{\Delta\Theta}_{a\beta} \equiv \begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array}$$



$$\begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array} \begin{array}{c} \sigma_{i,a}^{(\ell)} \sigma_{i,\beta}^{(\ell)} \\ \hline \hline \end{array} \sim \frac{1}{n}$$

Feynman rules relevant for the F-tensor

$$\widehat{\Delta\Theta}_{a\beta} \equiv \begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array}$$

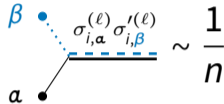
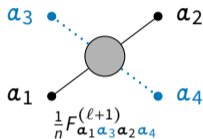


$$\begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array} \begin{array}{c} \sigma_{i,a}^{(\ell)} \sigma_{i,\beta}^{(\ell)} \\ \hline \hline \end{array} \sim \frac{1}{n}$$

The propagator \bigcirc satisfies selection rules, e.g.

Feynman rules relevant for the F-tensor

$$\widehat{\Delta\Theta}_{a\beta} \equiv \begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array}$$

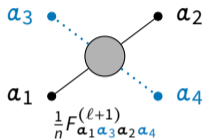


The **propagator**  satisfies selection rules, e.g.

- It cannot be directly connected to other propagators

Feynman rules relevant for the F-tensor

$$\widehat{\Delta\Theta}_{a\beta} \equiv \begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array}$$



$$\begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array} \begin{array}{c} \sigma_{i,a}^{(\ell)} \sigma_{i,\beta}^{(\ell)} \\ \hline \hline \end{array} \sim \frac{1}{n}$$

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

$$\widehat{\Delta\Theta}_{a\beta} \equiv \begin{array}{c} \beta \\ a \end{array} \begin{array}{c} \bullet \cdots \cdots \bullet \\ \bullet \cdots \cdots \bullet \end{array}$$

$$\begin{array}{c} a_3 \bullet \cdots \cdots \bullet a_2 \\ \bullet \cdots \cdots \bullet a_4 \\ a_1 \bullet \end{array} \begin{array}{c} \bullet \\ \bullet \end{array} \frac{1}{n} F_{a_1 a_3 a_2 a_4}^{(\ell+1)}$$

$$\begin{array}{c} \beta \bullet \cdots \cdots \bullet \\ a \bullet \end{array} \begin{array}{c} \sigma_{i,a}^{(\ell)} \sigma_{i,\beta}^{(\ell)} \\ \hline \hline \end{array} \sim \frac{1}{n}$$

The **propagator** \bigcirc 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

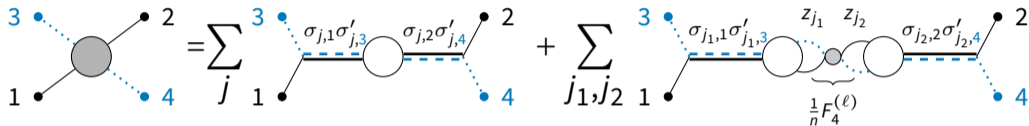
The F-recursion

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

$$\begin{array}{c} \text{Diagram 1: A grey circle with four external legs. Top-left: blue dot labeled 3, solid line labeled 1. Top-right: black dot labeled 2, solid line labeled 4. Bottom-left: solid line labeled 1, solid line labeled 3. Bottom-right: blue dot labeled 4, solid line labeled 2.} \end{array} = \sum_j \begin{array}{c} \text{Diagram 2: A white circle with two internal lines. Left: blue dot labeled 3, solid line labeled 1. Right: black dot labeled 2, solid line labeled 4. Internal lines: top-left to top-right labeled } \sigma_{j,1}\sigma'_{j,3} \text{ (dashed blue), bottom-left to bottom-right labeled } \sigma_{j,2}\sigma'_{j,4} \text{ (dashed blue).} \end{array} + \sum_{j_1, j_2} \begin{array}{c} \text{Diagram 3: A sequence of three circles. Left: blue dot labeled 3, solid line labeled 1. Middle: white circle with two internal lines labeled } \sigma_{j_1,1}\sigma'_{j_1,3} \text{ (dashed blue) and } \sigma_{j_2,2}\sigma'_{j_2,4} \text{ (dashed blue). Right: black dot labeled 2, solid line labeled 4. Between the first and second circle: two internal lines labeled } z_{j_1} \text{ and } z_{j_2} \text{ meeting at a grey circle. Below this grey circle is a bracket labeled } \frac{1}{n}F_4^{(\ell)}. \end{array}$$

The F-recursion

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



Compare to the analytical expression

$$\frac{1}{n} F_{1324}^{(\ell+1)} = \frac{1}{n} \langle \sigma_1^{(\ell)} \sigma_2^{(\ell)} \sigma_3'^{(\ell)} \sigma_4'^{(\ell)} \rangle_{K^{(\ell)}} \Theta_{34}^{(\ell)} \\ + \frac{1}{n} \sum_{a,\beta,\gamma,\delta=1}^4 \langle \sigma_1^{(\ell)} \sigma_3'^{(\ell)} z_a^{(\ell)} \rangle_{K^{(\ell)}} \langle \sigma_2^{(\ell)} \sigma_4'^{(\ell)} z_\beta^{(\ell)} \rangle_{K^{(\ell)}} K_{(\ell)}^{a\gamma} K_{(\ell)}^{\beta\delta} F_{\gamma 3 \delta 4}^{(\ell)}$$

The NTK recursion at finite width

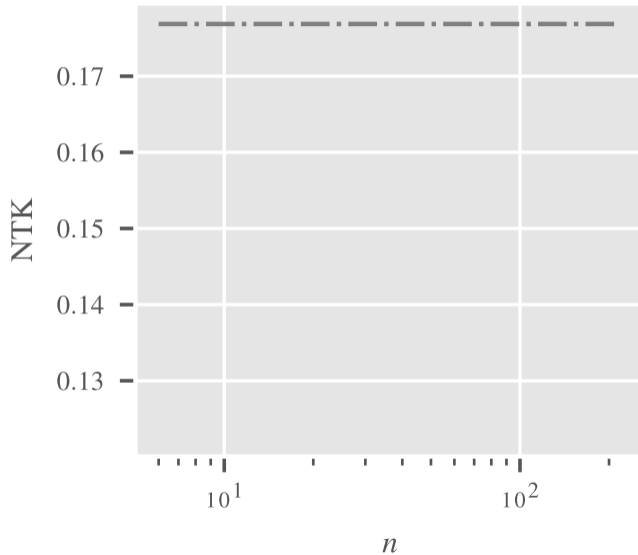
$$\begin{aligned}
 & \text{Diagram 1} = \text{Diagram 2} + \text{Diagram 3} + \text{Diagram 4} + \text{Diagram 5} + \text{Diagram 6} \\
 & \frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)} = \frac{1}{n_{\ell-1}} \Theta_{12}^{\{1\}(\ell)} + \frac{1}{n_{\ell-1}} K_{12}^{\{1\}(\ell)} + \frac{1}{n_{\ell-1}} V_4^{(\ell)} + \frac{1}{n_{\ell-1}} D_4^{(\ell)} + \frac{1}{n_{\ell-1}} F_4^{(\ell)}
 \end{aligned}$$

The NTK recursion at finite width

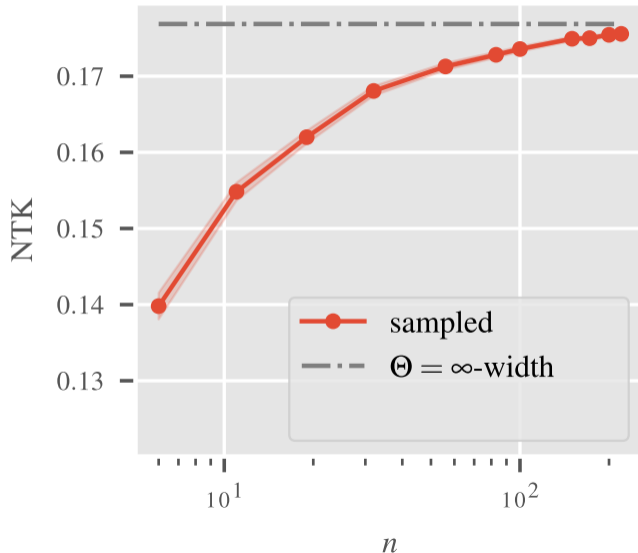
$$\begin{aligned}
 & \text{Diagram 1: } \frac{1}{n_\ell} \Theta_{12}^{\{1\}(\ell+1)} \\
 &= \text{Diagram 2: } \frac{1}{n_{\ell-1}} \Theta^{\{1\}(\ell)} + \text{Diagram 3: } \frac{1}{n_{\ell-1}} K^{\{1\}(\ell)} + \text{Diagram 4: } \frac{1}{n_{\ell-1}} V_4^{(\ell)} + \text{Diagram 5: } \frac{1}{n_{\ell-1}} D_4^{(\ell)} + \text{Diagram 6: } \frac{1}{n_{\ell-1}} F_4^{(\ell)}
 \end{aligned}$$

Ⓢ Feynman diagrams allow for much simpler derivation of recursion relations

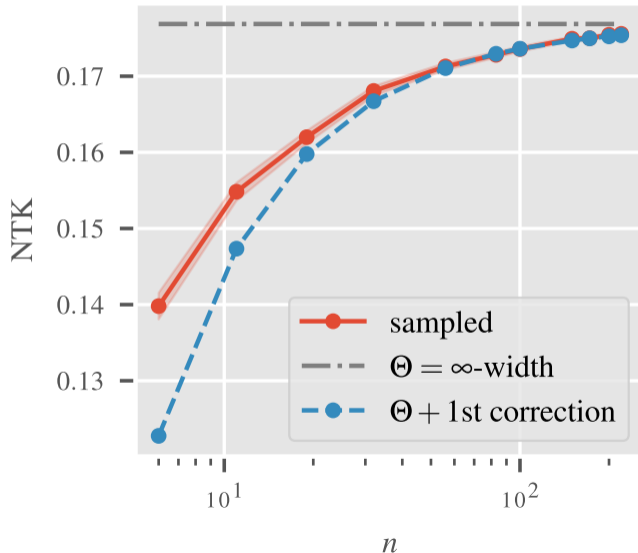
Numerical results



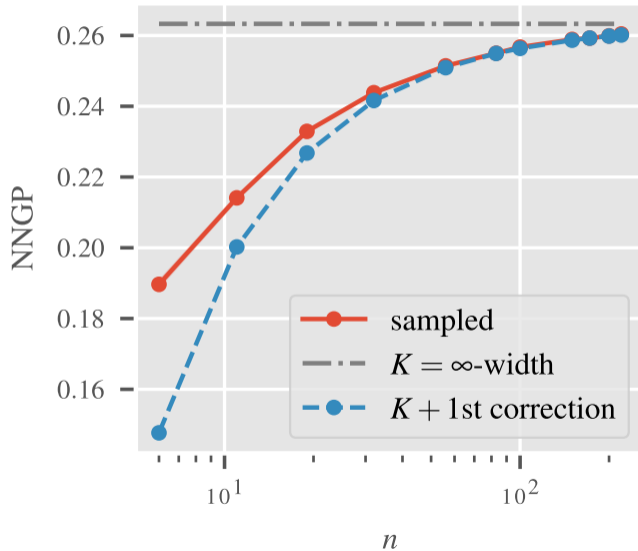
Numerical results



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 $1/\text{width}$ 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 $1/\text{width}$ 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!