Generalization capabilities of neural networks in lattice applications

Matteo Favoni

Institute for Theoretical Physics, TU Wien

July 29, 2021

Lattice 2021

Based on: S. Bulusu, M. Favoni, A. Ipp, D. I. Müller, D. Schuh, *Preprint* (2021) [2103.14686]

Code: gitlab.com/openpixi/scalar-ml

TECHNISCHE UNIVERSITÄT WIEN Vienna | Austria

Der Wissenschaftsfonds.

Doktoratskolleg Particles and Interactions

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas
- Strategy: meet the requirements of the specific problem

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas
- Strategy: meet the requirements of the specific problem
- In quantum field theories, symmetries play a key role

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas
- Strategy: meet the requirements of the specific problem
- In quantum field theories, symmetries play a key role
- A desirable approach is to design NNs so that such symmetries are respected

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas
- Strategy: meet the requirements of the specific problem
- In quantum field theories, symmetries play a key role
- A desirable approach is to design NNs so that such symmetries are respected
- Previous talk $\rightarrow$ gauge symmetry, this talk $\rightarrow$ translational symmetry

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas
- Strategy: meet the requirements of the specific problem
- In quantum field theories, symmetries play a key role
- A desirable approach is to design NNs so that such symmetries are respected
- Previous talk $\rightarrow$ gauge symmetry, this talk $\rightarrow$ translational symmetry
- Convolutional neural networks (CNNs) incorporate translational symmetry under certain circumstances

Introduction

- Neural networks (NNs) are a widely used tool in many scientific areas
- Strategy: meet the requirements of the specific problem
- In quantum field theories, symmetries play a key role
- A desirable approach is to design NNs so that such symmetries are respected
- Previous talk $\rightarrow$ gauge symmetry, this talk $\rightarrow$ translational symmetry
- Convolutional neural networks (CNNs) incorporate translational symmetry under certain circumstances
- Investigate generalization capabilities in terms of different lattice sizes and different physical parameters

CNNs

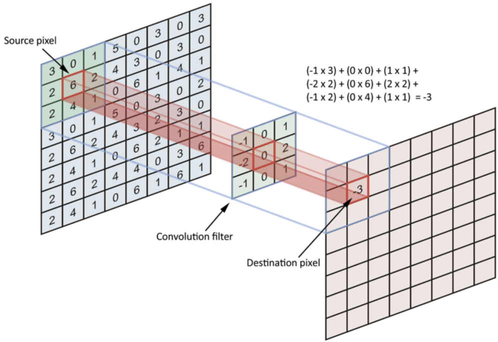


Image from here

CNNs

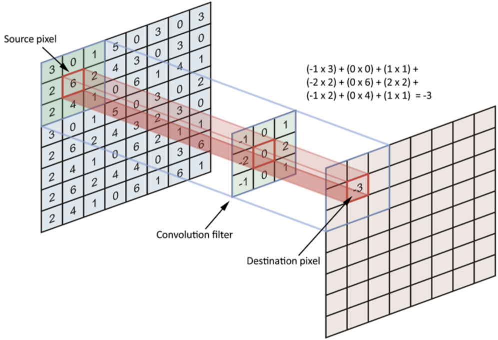


Image from here

- Equivariance vs invariance

CNNs

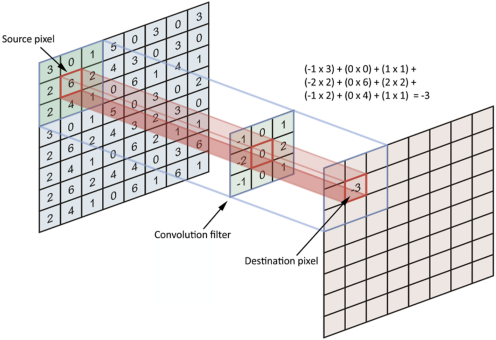


Image from here

- Equivariance vs invariance
- Equivariance before a global pooling layer is a sufficient condition for output invariance

CNNs

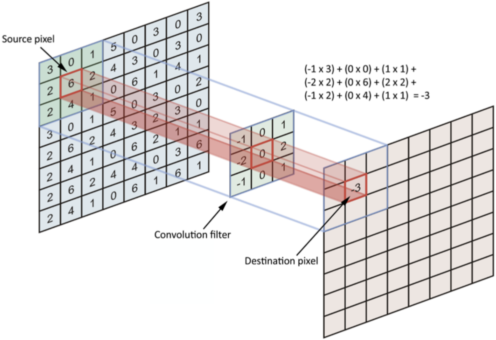
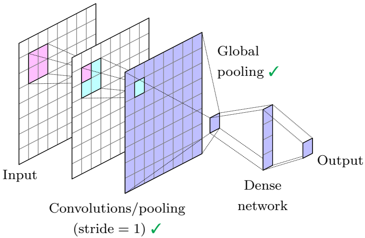


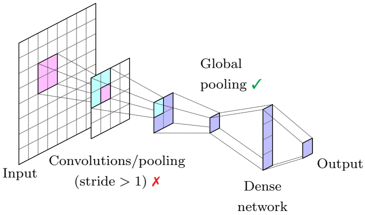
Image from here

- Equivariance vs invariance
- Equivariance before a global pooling layer is a sufficient condition for output invariance
- Does translational symmetry make a significant difference?

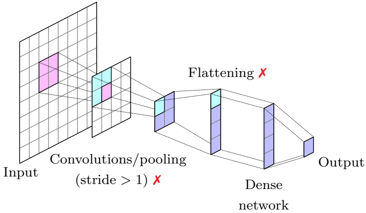
Architecture types



Equivariant architecture (EQ)



Strided architecture (ST)



Flattening architecture (FL)

Physical system

- Complex scalar field in 1+1D with nonzero chemical potential

$$S = \int \mathrm{d}x_0 \mathrm{d}x_1 \left(|D_0\phi|^2 - |\partial_1\phi|^2 - m^2|\phi|^2 - \lambda|\phi|^4 \right), \quad D_0 = \partial_0 - i\mu \tag{1}$$

Physical system

- Complex scalar field in 1+1D with nonzero chemical potential

$$S = \int \mathrm{d}x_0 \mathrm{d}x_1 \left(|D_0\phi|^2 - |\partial_1\phi|^2 - m^2|\phi|^2 - \lambda|\phi|^4 \right), \quad D_0 = \partial_0 - i\mu \tag{1}$$

- Discretized action

$$S_{lat} = \sum_x \left(\eta|\phi_x|^2 + \lambda|\phi_x|^4 - \sum_{\nu=1}^{2} \left(e^{\mu\,\delta_{\nu,2}} \phi_x^* \phi_{x+\hat{\nu}} + e^{-\mu\,\delta_{\nu,2}} \phi_x^* \phi_{x-\hat{\nu}} \right) \right), \quad \eta = 2D + m^2 \tag{2}$$

Physical system

- Complex scalar field in 1+1D with nonzero chemical potential

$$S = \int \mathrm{d}x_0 \mathrm{d}x_1 \left(|D_0\phi|^2 - |\partial_1\phi|^2 - m^2|\phi|^2 - \lambda|\phi|^4 \right), \quad D_0 = \partial_0 - i\mu \tag{1}$$

- Discretized action

$$S_{lat} = \sum_x \left(\eta|\phi_x|^2 + \lambda|\phi_x|^4 - \sum_{\nu=1}^{2} \left(e^{\mu\,\delta_{\nu,2}} \phi_x^* \phi_{x+\hat{\nu}} + e^{-\mu\,\delta_{\nu,2}} \phi_x^* \phi_{x-\hat{\nu}} \right) \right), \quad \eta = 2D + m^2 \tag{2}$$

- Sign problem solved by a dual formulation: $\phi_x \to \{k_{x,\nu}, l_{x,\nu}\}$ integer fields, Gattringer, Kloiber, arxiv:1206.2954

Physical system

- Complex scalar field in 1+1D with nonzero chemical potential

$$S = \int \mathrm{d}x_0 \mathrm{d}x_1 \left(|D_0 \phi|^2 - |\partial_1 \phi|^2 - m^2 |\phi|^2 - \lambda |\phi|^4 \right), \quad D_0 = \partial_0 - i\mu \tag{1}$$

- Discretized action

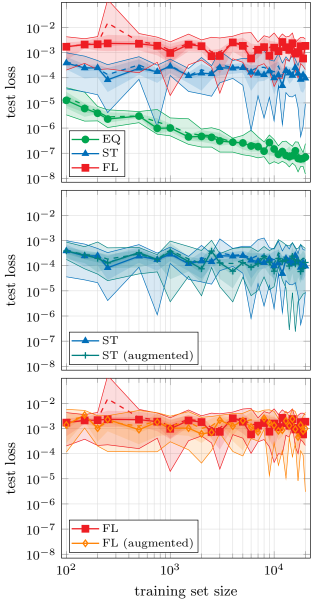
$$S_{lat} = \sum_x \left(\eta |\phi_x|^2 + \lambda |\phi_x|^4 - \sum_{\nu=1}^{2} \left(e^{\mu\, \delta_{\nu,2}} \phi_x^* \phi_{x+\hat{\nu}} + e^{-\mu\, \delta_{\nu,2}} \phi_x^* \phi_{x-\hat{\nu}} \right) \right), \quad \eta = 2D + m^2 \tag{2}$$

- Sign problem solved by a dual formulation: $\phi_x \to \{k_{x,\nu}, l_{x,\nu}\}$ integer fields, Gattringer, Kloiber, arxiv:1206.2954
- Regression task: predicting observables

$$n = \frac{1}{N} \sum_x k_{x,2}\,, \qquad |\phi|^2 = \frac{1}{N} \sum_x \frac{W(f_x + 2)}{W(f_x)} \tag{3}$$

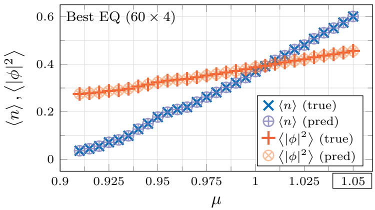
$$f_x = \sum_\nu [|k_{x,\nu}| + |k_{x-\hat{\nu},\nu}| + 2(l_{x,\nu} + l_{x-\hat{\nu},\nu})]\,, \quad W(f_x) = \int_0^\infty \mathrm{d}x\, x^{f_x+1} \mathrm{e}^{-\eta x^2 - \lambda x^4} \tag{4}$$

Architecture comparison

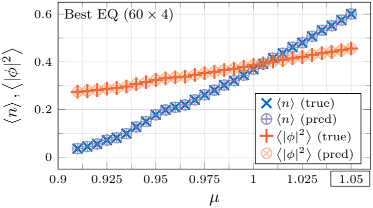


- Systematic architecture search with optuna, Akiba et al., arxiv:1907.10902
- 10 instances of the winning architectures are retrained from scratch for various training set size
- EQ beats ST and FL for any number of training samples
- EQ improves with more samples, while the other two do not
- Data augmentation does not help the two non-equivariant architectures

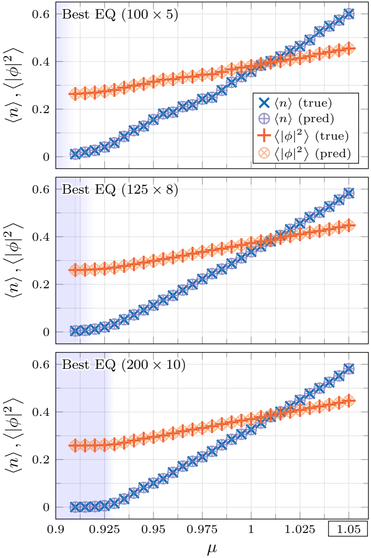
Silver blaze phase transition



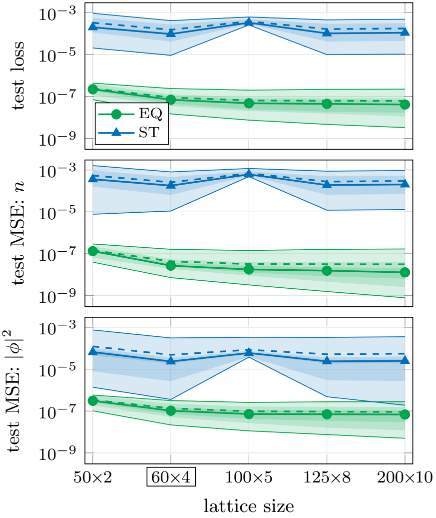
Silver blaze phase transition



Training on both phases is not necessary as long as the expression of the observables is independent of the transition

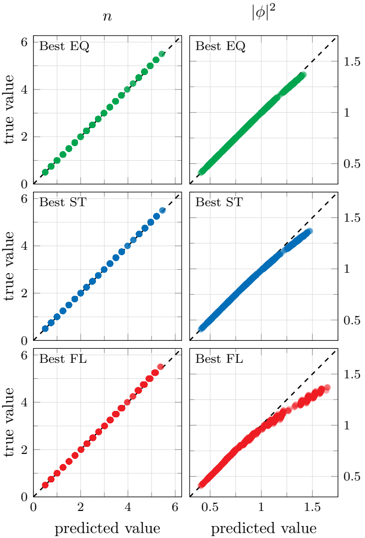


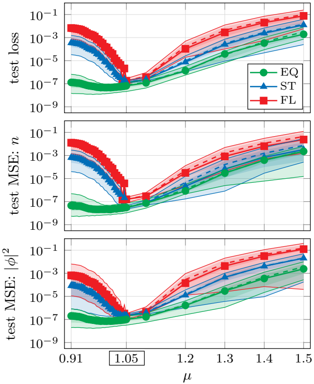
Generalization to other lattice sizes and physical parameters



- FL cannot be tested on larger lattices
- Training only on 60×4
- Kink in ST at 100×5 due to $s = 2$ in spatial pooling layer
- EQ clearly outperforms ST

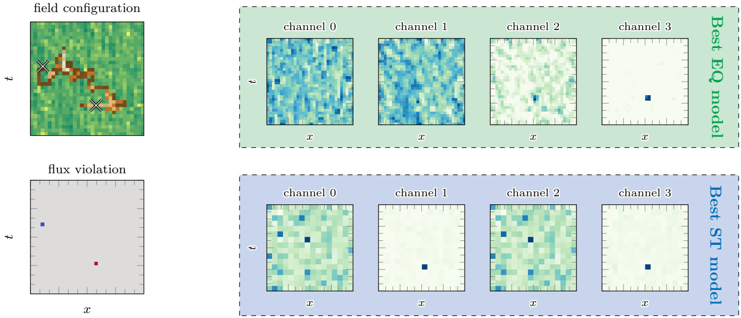
Extrapolation to larger chemical potentials





Detecting flux violations

The field k obeys the conservation law $\sum_\nu (k_{x,\nu} - k_{x-\hat{\nu},\nu}) = 0$. We artificially created flux violations to be detected by the models.



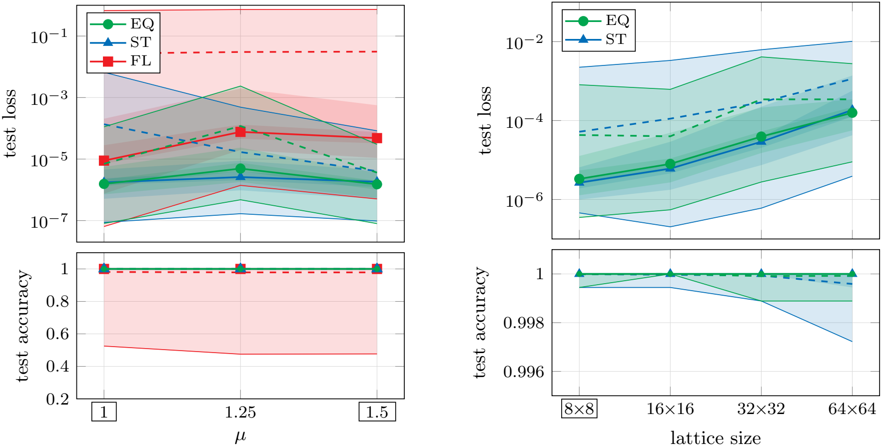
(a) Example field configuration

(b) Feature maps of convolutional network in best EQ and ST models

- 2x2 convolutions are necessary for this task
- Similar approach with optuna

Results

Training at $(\eta, \mu) = (4.25, 1)$ and $(4.01, 1.5)$ on 8x8 lattice with $N_{train} = 4000$; testing at $\eta \in \{4.01, 4.04, 4.25\}$, $\mu \in \{1, 1.25, 1.5\}$ on 4 lattice sizes



Counting flux violations

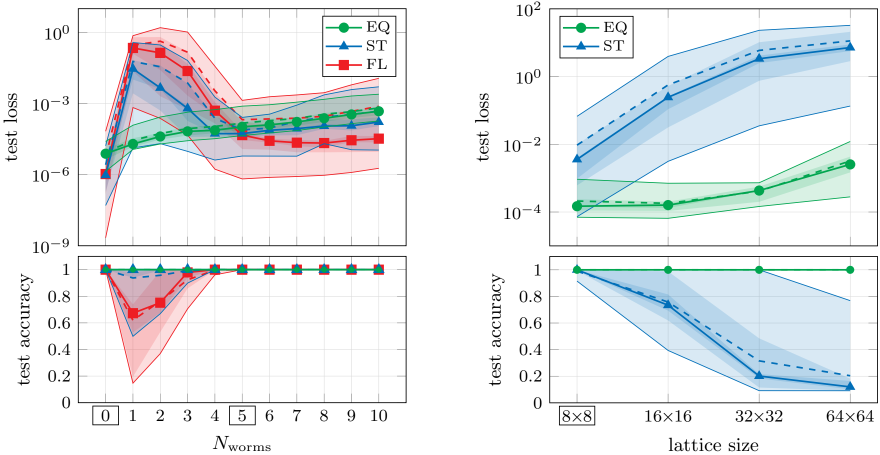
- Simplified version of counting problems (e.g.: crowd counting)

Counting flux violations

- Simplified version of counting problems (e.g.: crowd counting)
- Training only at 0 and 5 worms (we train on 4 combinations of parameters out of 396 used for testing) with $N_{train} = 20000$

Counting flux violations

- Simplified version of counting problems (e.g.: crowd counting)
- Training only at 0 and 5 worms (we train on 4 combinations of parameters out of 396 used for testing) with $N_{train} = 20000$



Summary

- Performance of three architecture types on three different tasks

Summary

- Performance of three architecture types on three different tasks
- Stride $s > 1$ and flattening layer break translational equivariance

Summary

- Performance of three architecture types on three different tasks
- Stride $s > 1$ and flattening layer break translational equivariance
- In all tasks EQ proved to be a highly reliable choice

Summary

- Performance of three architecture types on three different tasks
- Stride $s > 1$ and flattening layer break translational equivariance
- In all tasks EQ proved to be a highly reliable choice
- Optuna favoured architectures with $< 10^5$ parameters

Summary

- Performance of three architecture types on three different tasks
- Stride $s > 1$ and flattening layer break translational equivariance
- In all tasks EQ proved to be a highly reliable choice
- Optuna favoured architectures with $< 10^5$ parameters
- Remarkable generalization capabilities of EQ

Backup slides

First task optuna winners

EQ	ST	FL
Conv(1 $\times$ 1, 4, 64)	Conv(1 $\times$ 1, 4, 80)	Conv(1 $\times$ 1, 4, 64)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(1 $\times$ 1, 64, 48)	Conv(1 $\times$ 1, 80, 80)	Conv(2 $\times$ 2, 64, 80)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(1 $\times$ 1, 48, 80)	Conv(1 $\times$ 1, 80, 48)	AvgPool(2 $\times$ 2, 2)
LeakyReLU	LeakyReLU	Conv(1 $\times$ 1, 80, 48)
Conv(2 $\times$ 2, 80, 80)	AvgPool(2 $\times$ 2, 2)	LeakyReLU
LeakyReLU	Conv(2 $\times$ 2, 48, 80)	Conv(2 $\times$ 2, 48, 64)
GlobalAvgPool	LeakyReLU	LeakyReLU
Linear(80, 2)	GlobalAvgPool	AvgPool(2 $\times$ 2, 2)
	Linear(80, 2)	Conv(1 $\times$ 1, 64, 24)
		Flatten
		Linear(360, 24)
		LeakyReLU
		Linear(24, 2)
33202	26370	47394

Second task optuna winners

EQ	**ST**	**FL**
Conv(2×2, 4, 32)	Conv*(2×2, 4, 16)	Conv*(3×3, 4, 8)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(1×1, 32, 32)	MaxPool(2×2, 2)	MaxPool(2×2, 2)
LeakyReLU	Conv(1×1, 16, 16)	Conv(2×2, 8, 32)
GlobalMaxPool	LeakyReLU	LeakyReLU
Linear(32, 32)	Conv(1×1, 16, 8)	AvgPool(2×2, 2)
LeakyReLU	LeakyReLU	Conv(2×2, 32, 32)
Linear*(32, 1)	GlobalMaxPool	LeakyReLU
Sigmoid	Linear*(8, 32)	Flatten
	Linear(32, 1)	Linear*(128, 1)
	Sigmoid	Sigmoid
2657	953	5600

The star (e.g. Conv*) indicates that the bias in that layer is set to 0

Third task EQ optuna winners

1st EQ	2nd EQ	3rd EQ
Conv(1×1, 4, 32)	Conv(2×2, 4, 8)	Conv(1×1, 4, 4)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(2×2, 32, 8)	Conv(2×2, 8, 8)	Conv(2×2, 4, 8)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(2×2, 8, 16)	Conv(1×1, 8, 4)	Conv(2×2, 8, 4)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(1×1, 16, 8)	Conv(1×1, 4, 8)	Conv(3×3, 4, 1)
LeakyReLU	LeakyReLU	LeakyReLU
GlobalSumPool	GlobalSumPool	GlobalSumPool
Linear(8, 1)	Linear(8, 1)	
1800	456	308

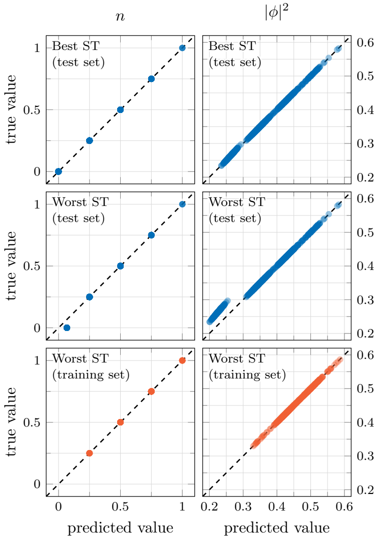
Third task ST optuna winners

1st ST	2nd ST	3rd ST
Conv(2×2, 4, 16)	Conv(2×2, 4, 4)	Conv(2×2, 4, 4)
LeakyReLU	LeakyReLU	LeakyReLU
Conv(1×1, 16, 32)	MaxPool(2×2, 2)	AvgPool(2×2, 2)
LeakyReLU	Conv(2×2, 4, 4)	Conv(3×3, 4, 16)
Conv(1×1, 32, 32)	LeakyReLU	LeakyReLU
LeakyReLU	GlobalSumPool	GlobalSumPool
AvgPool(2×2, 2)	Linear(4, 1)	Linear(16, 32)
Conv(1×1, 32, 8)		LeakyReLU
LeakyReLU		Linear(32, 1)
GlobalSumPool		
Linear(8, 32)		
LeakyReLU		
Linear(32, 1)		
2336	132	1184

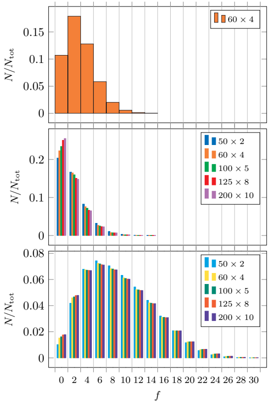
Third task FL optuna winners

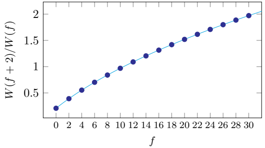
1st FL	2nd FL	3rd FL
Conv(2×2, 4, 4)	Conv(2×2, 4, 8)	Conv(2×2, 4, 32)
LeakyReLU	LeakyReLU	LeakyReLU
AvgPool(2×2, 2)	AvgPool(2×2, 2)	AvgPool(2×2, 2)
Conv(3×3, 4, 8)	Conv(3×3, 8, 4)	Conv(3×3, 32, 4)
LeakyReLU	LeakyReLU	LeakyReLU
AvgPool(2×2, 2)	AvgPool(2×2, 2)	AvgPool(2×2, 2)
Flattening	Flattening	Flattening
Linear(8, 4)	Linear(4, 4)	Linear(4, 32)
LeakyReLU	LeakyReLU	LeakyReLU
Linear(4, 32)	Linear(4, 32)	Linear(32, 16)
LeakyReLU	LeakyReLU	LeakyReLU
Linear(32, 1)	Linear(32, 1)	Linear(16, 1)
640	640	2704

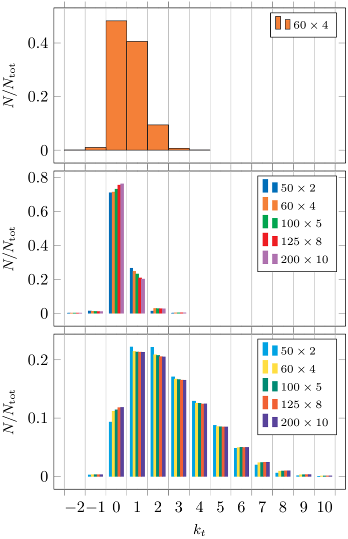
Why do ST fail?



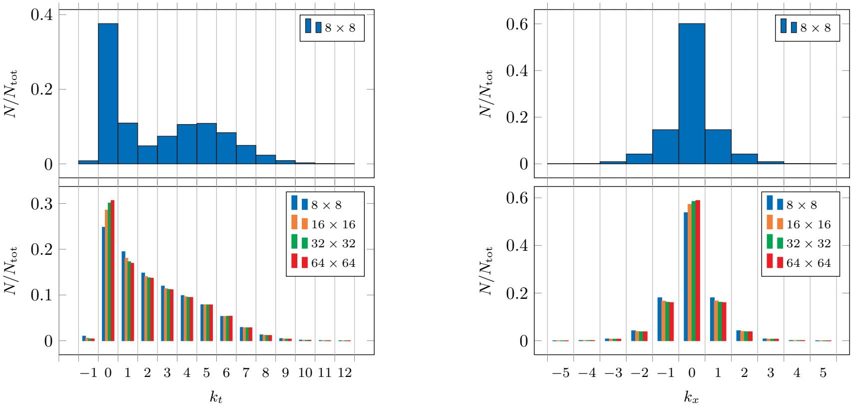
First task data distribution



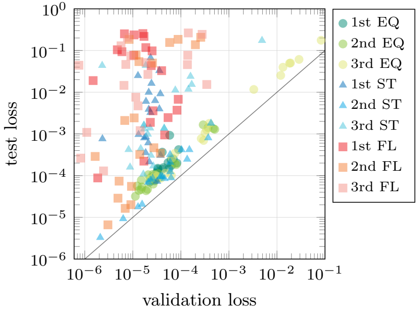


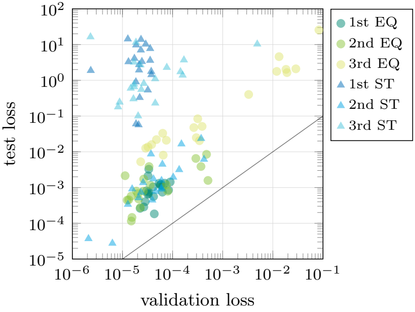


Third task data distribution



Third task: test loss vs validation loss





Test loss vs validation loss table

	validation loss on 8×8		test loss on 8×8		test loss up to 64×64	
	mean	median	mean	median	mean	median
1st EQ	$\mathbf{4.676 \times 10^{-5}}$	4.137×10^{-5}	$\mathbf{2.108 \times 10^{-4}}$	1.483×10^{-4}	$\mathbf{1.008 \times 10^{-3}}$	8.308×10^{-4}
2nd EQ	1.042×10^{-4}	$\mathbf{2.440 \times 10^{-5}}$	3.525×10^{-4}	$\mathbf{8.783 \times 10^{-5}}$	1.807×10^{-3}	$\mathbf{7.936 \times 10^{-4}}$
3rd EQ	8.992×10^{-3}	3.072×10^{-4}	2.105×10^{-2}	9.163×10^{-4}	1.925	4.031×10^{-2}
1st ST	$\mathbf{2.331 \times 10^{-5}}$	2.173×10^{-5}	9.438×10^{-3}	3.576×10^{-3}	4.446	3.026
2nd ST	8.479×10^{-5}	4.372×10^{-5}	$\mathbf{2.545 \times 10^{-4}}$	$\mathbf{9.340 \times 10^{-5}}$	$\mathbf{3.738 \times 10^{-3}}$	$\mathbf{1.171 \times 10^{-3}}$
3rd ST	2.869×10^{-4}	$\mathbf{2.171 \times 10^{-5}}$	1.676×10^{-2}	1.381×10^{-3}	2.943	9.580×10^{-1}
1st FL	$\mathbf{2.602 \times 10^{-5}}$	1.787×10^{-5}	7.837×10^{-2}	3.817×10^{-2}	-	-
2nd FL	4.004×10^{-5}	1.117×10^{-5}	$\mathbf{5.300 \times 10^{-2}}$	$\mathbf{1.285 \times 10^{-3}}$	-	-
3rd FL	5.805×10^{-5}	$\mathbf{1.031 \times 10^{-5}}$	6.382×10^{-2}	3.556×10^{-2}	-	-